Hamiltonian Quantum Cellular Automata in 1D

Daniel Nagaj

Center for Theoretical Physics, MIT, Cambridge, MA 02139, USA
Research Center for Quantum Information, Slovak Academy of Sciences,
Dubravska cesta 9, 845 11 Bratislava, Slovakia and
Quniverse, Liscie udolie 116, 841 04, Bratislava, Slovakia[*]

Pawel Wocjan

School of Electrical Engineering and Computer Science,
University of Central Florida, Orlando, FL 32816, USA[†]
(Dated: April 22, 2008)

We construct a simple translationally invariant, nearest-neighbor Hamiltonian on a chain of 10-dimensional qudits that makes it possible to realize universal quantum computing without any external control during the computational process. We only require the ability to prepare an initial computational basis state which encodes both the quantum circuit and its input. The computational process is then carried out by the autonomous Hamiltonian time evolution. After a time polynomially long in the size of the quantum circuit has passed, the result of the computation is obtained with high probability by measuring a few qudits in the computational basis.

This result also implies that there cannot exist efficient classical simulation methods for generic translationally invariant nearest-neighbor Hamiltonians on qudit chains, unless quantum computers can be efficiently simulated by classical computers (or, put in complexity theoretic terms, unless BPP=BQP).

I. INTRODUCTION

One of the most important challenges in quantum information science is to identify quantum systems that can be controlled in such a way that they can be used to realize universal quantum computing. The quantum circuit model abstracts from the details of concrete physical systems and states that the required elementary control operations are: (i) initialization in basis states, (ii) implementation of one and two-qubit gates, and (iii) measurement of single qubits in basis states. Meanwhile, many other models have been proposed such as measurement-based quantum computing [6, 11, 15, 18], adiabatic quantum computing [2, 7], or topological quantum computing [10] that reduce or modify the set of elementary control operations. However, the common principle underlying all these models is that the computation process is always driven by applying a sequence of control operations.

Instead, we consider a model that does not require any control during the computational process. This model consists of a quantum system with a Hamiltonian that makes it possible to realize universal quantum computing by the following protocol: (1) prepare an initial state in the computational basis that encodes both the program and input, (2) let the Hamiltonian time evolution act undisturbed for a sufficiently long time, and (3) measure a small subsystem in the computational basis to obtain the result of the computation with high probability. We refer to this model as a *Hamiltonian quantum computer* and more specifically as a *Hamiltonian quantum cellular automaton* (HQCA) provided that the Hamiltonian acts on qudits that are arranged on some lattice, is invariant with respect to translations along the symmetry axis of the lattice, and contains only finite range interactions. Most natural Hamiltonians have these properties, so it is important to construct HQCA that are as close as possible to natural interactions.

Hamiltonian QCA are related to the more usual discrete-time QCA (for further review of the different types of quantum cellular automata we refer the reader to [16]). However, while the evolution of discrete-time QCA proceeds in discrete update steps (corresponding to tensor products of local unitary operations, see e.g. [17, 19]), the states of Hamiltonian QCA change in a continuous way according to the Schrödinger equation (with a time-independent Hamiltonian). For this reason, Hamiltonian QCA are also called *continuous-time* QCA [16]. Also, in the HQCA model, all the couplings (interactions) are present all the time, while for the the discrete-time QCA, the execution of updates on overlapping cells is synchronized by external control. Therefore, the nearest-neighbor interactions of a

arXiv:0802.0886v4 [quant-ph] 14 Aug 2008

HQCA have to include a mechanism that ensures that the logical transformations are carried out in the correct order.

The motivation to consider Hamiltonian computers is threefold. First, it is a fundamental question in the thermodynamics of computation how to realize computational processes within a closed physical system. Such Hamiltonian computers were presented and discussed by Benioff [4], Feynman [8], and Margolus [9]. Second, Hamiltonian quantum cellular automata could lead to new ideas for reducing the set of necessary control operations in current proposals for quantum computing by using the inherent computational power of the interactions. HQCA are at one end of the spectrum of possible implementations; more realistic perspectives for quantum computing could arise by combining this model with more conventional models involving external control operations throughout the computation. Third, this model can show the limitations of current and future methods in condensed matter physics for simulating the time evolution of translationally invariant systems. If evolving with a certain Hamiltonian can realize universal quantum computing, then there cannot exist any classical method for efficiently simulating the corresponding time evolution unless classical computers are as powerful as quantum computers (BPP=BQP).

The first theoretical computational models based on a single time-independent Hamiltonian go back to [4, 8, 9]. However, these Hamiltonian computers were not explicitly designed for realizing universal *quantum* computing. Margolus' model [9] has the attractive feature that it is laid out on a 2-dimensional lattice with translationally invariant, finite-range interactions. (In [5] it was argued that the part of the Hamiltonian responsible for the synchronization in a 1-dimensional variant is close to real interaction in solid states.) However, this scheme does not satisfy the requirement (1) since its initial state has to be prepared in a superposition. Building upon Margolus' idea, a translationally invariant Hamiltonian universal for quantum computing even if the initial state is restricted to be a canonical basis state was given in [12]. This model requires 10-local, finite-range interactions among qubits on a 2-dimensional rectangular lattice wrapped around a cylinder. Subsequently, it was established in [13] that nearest-neighbor interactions among qutrits on a 2-dimensional lattice suffice. However, the Hamiltonian of [13] is translationally invariant only when translated over several lattice sites. A different approach was taken by Vollbrecht and Cirac in [21], showing that one can implement universal quantum computation with a translationally invariant, nearest-neighbor Hamiltonian on a chain of 30-dimensional qudits. Also, recently another 1D translationally invariant Hamiltonian computer construction was given by Kay [22], using particles with dimension $d = 31$.

We present two different simplified HQCA constructions on one-dimensional qudit chains. In both models, we think of the qudit chain as composed of two registers, data and program. The work qubits we compute on are located at a static location in the data register. Driven by the autonomous Hamiltonian time evolution, the program sequence contained in the program register moves past the work qubits and the gates are applied to them. After we let the system evolve for a time not larger than a polynomial in the length of the program, we measure one or two qudits in the computational basis to read out the output of the computation with high probability.

Our first construction is for a chain of 10-dimensional qudits and is related to the ideas of [21]. The mechanism behind the progress of the program sequence in this particular model can be thought of as the diffusion of a system of free fermions on a line. Concurrently with our preprint, Chase and Landahl [23] found another 1D construction with particles with dimension $d = 8$. However, their Hamiltonian is not translationally invariant. If we release the translational invariance requirement in our $d = 10$ model, we obtain a $d = 8$ construction as well. Whether the required dimension $d = 10$ can be decreased while keeping translational invariance in our HQCA model remains an open question.

Our second construction uses qudits with dimension $d = 20$ and is inspired by [14], utilizing a technique of [3] to transport the program. Here, the mechanism for the progress of the computation can be thought of as a quantum walk on a line.

The paper is organized as follows. First, in Section II A we present the HQCA construction with cell size $d = 10$ and analyze the required run-time of this model in Section II B. Second, we give the HQCA construction with cell size $d = 20$ in Section III A and discuss the readout procedure and the required run-time in Section III B. We provide some useful results for the continuous time quantum walk on a line in Appendix A and prove a lemma concerning the diffusion of free fermions on a line in Appendix B.

II. THE $d = 10$ HAMILTONIAN QUANTUM CELLULAR AUTOMATON

We present a simple universal HQCA on a chain of qudits with dimension $d = 10$. First, we encode the progression of a quantum circuit U on N qubits into a set of states $|\varphi_\sigma\rangle$ of a chain of qudits with length $L = poly(N)$. Second, we give a translationally invariant nearest-neighbor Hamiltonian on this chain of qudits, which induces a quantum walk on the set of states $|\varphi_\sigma\rangle$. Finally, using a mapping to a system of free fermions in 1D, we prove that when we initialize the qudit chain in an easily determined computational basis state and let the system evolve for a time $\tau \leq \tau_{10} = O(L \log L)$ chosen uniformly at random, we can read out the result of the quantum circuit U with probability
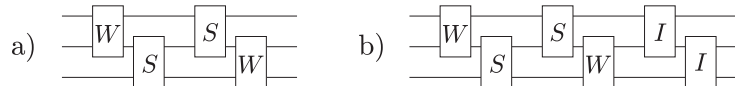
FIG. 1: a) A quantum circuit consisting of two rounds of gates acting on nearest neighbors. b) The previous circuit with a third round of identity gates added.

$p_{10} \geq \frac{5}{6} - O\left(\frac{1}{\log L}\right)$ by measuring one of the qudits in the computational basis. We then show that this is enough to ensure universality of our HQCA for the class BQP.

A. The Construction

1. Encoding a quantum circuit

The gate set {Toffoli, Hadamard} is universal for quantum computation [20]. With only polynomial overhead, one can simulate a circuit consisting of these gates using only the gate W (controlled $\frac{\pi}{2}$ rotation about the y-axis)

$$
W = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}
\tag{1}
$$

if it can be applied to any pair of qubits. Let us consider implementing universal quantum computation on a qubit chain using only nearest neighbor gates. Let us also restrict the use of the W gate so that the control qubit has to be to the left of the target qubit. Using only polynomially many additional swap gates S, one can still do universal quantum computation on a qubit chain. Thus given a quantum circuit U' on N' qubits with $poly(N')$ generic two-qubit gates, we can transform it into a circuit U on a chain of $N = poly(N')$ qubits with nearest neighbor gates W (with control on the left) and S without loss of universality. We then add identity to our gate set and further transform the circuit U to have the following form (see Figure 1). Rewrite the circuit as K rounds of nearest neighbor gates $U_{k,g} \in \{W, S, I\}$, where gate $U_{k,g}$ belongs to the k-th round and acts on the pair of qubits w_g, w_{g+1}:

$$
U = (U_{K,N-1} \ldots U_{K,1}) \cdots (U_{1,N-1} \ldots U_{1,1}).
\tag{2}
$$

We wish to encode the progression of the circuit U into the states of a chain of qudits with dimension 10, with length $L = poly(N)$. The basis states of each qudit $|q\rangle = |p\rangle \otimes |d\rangle$ are constructed as a tensor product of a 5-dimensional program register and a 2-dimensional data register, where $p \in \{\ \cdot\ , \blacktriangleright, W, S, I\}$ and $d \in \{0, 1\}$ label the corresponding basis states. We start by writing the initial product state

$$
|\varphi\rangle = \bigotimes_{j=1}^{L} (|p_j\rangle \otimes |d_j\rangle)_j,
\tag{3}
$$

with p_j and d_j as in Figure 2 (here we give an example for the circuit in Figure 1a):

j	1			$\cdots$	M	$\cdots$						$2M$
p_j	$\cdot$	$\cdot$	$\blacktriangleright$	$\cdot$	$\cdot$	$\blacktriangleright$	I	W	S	I	S	W
d_j	0	0	0	0	0	0	w_1	w_2	w_3	0	0	0

(4)

The qudit chain has length

$$
L = 2M = 2KN,
\tag{5}
$$

where K is the number of the rounds of gates in (2). The left half of the top (program) register contains K pointer symbols $\blacktriangleright$ at positions kN for $k = 1 \ldots K$ and empty symbols $\cdot$ everywhere else. The right half holds the program
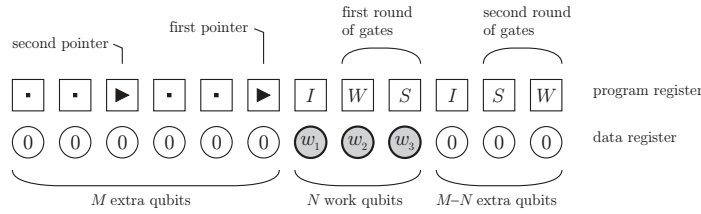
FIG. 2: The initial state $|\varphi\rangle$ of the qudit chain for the circuit in Figure 1a).

in the form

$$I \underbrace{U_{1,1}\ldots U_{1,N-1}}_{\text{first round of gates}} I \underbrace{U_{2,1}\ldots U_{2,N-1}}_{\text{second round of gates}} I \ldots I \underbrace{U_{K,1}\ldots U_{K,N-1}}_{\text{last round of gates}}, \qquad (6)$$

with $U_{k,g} \in \{W, S, I\}$ and each round of gates is preceded by an identity gate. Later we will show a way to execute the program while moving it to the left above the data. The bottom (data) register contains N work qubits labeled w_n in a canonical basis product state at positions $M + n$ for $n = 1 \ldots N$. We designate w_N as the readout qubit. Finally, the rest of the data register (positions $1, \ldots, M$ and $M + N + 1, \ldots, 2M$) is filled with extra qubits in the state $|0\rangle$.

We now describe how to generate the set of states $\{|\varphi_\sigma\rangle\}$ encoding the progression of the quantum computation U. Starting from the initial state $|\varphi\rangle$ (4), we obtain each state $|\varphi_\sigma\rangle$ by applying a particular sequence σ of the following two update rules. Each of these rules can be applied to two neighboring qudits in the chain. The first rule is: if there is an empty spot $\cdot$ to the left of a symbol $A \in \{W, S, I\}$ (from now on we call it a 'gate') somewhere in the program register, move A one step to the left, leaving the data register unchanged.

$$1 : \begin{array}{|c|c|} \hline \cdot & A \\ \hline x & y \\ \hline \end{array} \rightarrow \begin{array}{|c|c|} \hline A & \cdot \\ \hline x & y \\ \hline \end{array} \qquad (7)$$

The second rule says: at a place in the qudit chain where a gate $A \in \{W, S, I\}$ meets a 'pointer' $\blacktriangleright$, switch their positions and apply the gate A to the qubits in the data register below.

$$2 : \begin{array}{|c|c|} \hline \blacktriangleright & A \\ \hline x & y \\ \hline \end{array} \rightarrow \begin{array}{|c|c|} \hline A & \blacktriangleright \\ \hline A(x,y) \\ \hline \end{array} \qquad (8)$$

There is only one place where one of these rules can be applied to the initial state (4) – the second rule can be applied at the place where $\blacktriangleright$ stands to the left of I. After this first step, there are several ways to proceed, generating many different states $|\varphi_\sigma\rangle$.

Let us look more closely at what happens to the initial state $|\varphi\rangle$ as we apply some sequence of rules. There are K pointers $\blacktriangleright$ in $|\varphi\rangle$, one for each round of gates in the circuit (see Figure 2). Observe that $|\varphi\rangle$ is constructed in such a way that as a gate $U_{k,g}$ from the k-th round of gates moves to the left, it meets the k-th (counting from the right) pointer $\blacktriangleright$ exactly above the work qubits w_g, w_{g+1} (see Figure 3c for a depiction of the second pointer meeting the second round of gates). As noted before (2), this is the pair of qubits the gate $U_{k,g}$ should act on. It could then seem that after many applications of the update rules the program moves to the left of the work qubits while the corresponding states $|\varphi_\sigma\rangle$ we obtain have the computation executed on their work qubits as planned. However, before drawing that conclusion, we need to consider what happens when a gate $U_{k,g}$ from the k-th round meets a pointer while not above the work qubits (see Figure 3). First, if this happens above a pair of (extra) data qubits in the state $|00\rangle$, the qubits stay unchanged after we apply (8), as we have $U_{k,g}|00\rangle = |00\rangle$ for the three possible $U_{k,g}$ (the controlled gate W, a swap or the identity). The second possibility is that a gate meets a pointer above the boundary of the work qubits, i.e. either above d_M, w_1 or above w_N, d_{M+N+1}. Observe that the identity gates we inserted in front of each round of gates in (6) are the only ones for which this can happen. The extra qubit and the work qubit involved then stay unchanged after the application of (8). The extra qubits in the data register thus always remain in the state $|0\rangle$ and the only processing in the data register happens on the work qubits. The gates in (6) are applied sequentially from left to right. Therefore, when all the gates move to the left half of the chain, the corresponding state $|\varphi_\sigma\rangle$ then contains the result of the quantum circuit U in the state of the work qubits.
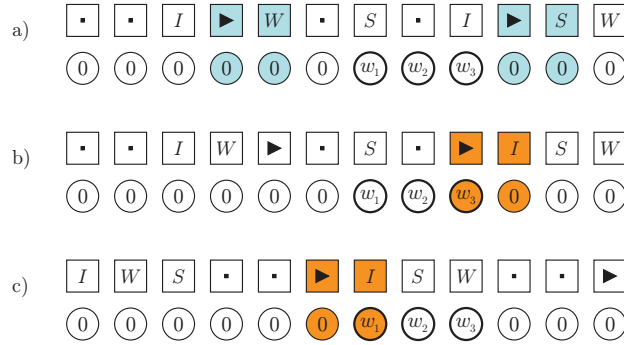
FIG. 3: Analysis of the second rule (8). a) A gate meeting a pointer above two extra qubits in the state $|00\rangle$. b) The gate meeting a pointer above the right boundary of the work qubits is the identity gate. c) The only gate meeting a pointer above the left boundary of the work qubits is the identity gate.

2. The Hamiltonian

We now construct a Hamiltonian whose transition rules will be (7, 8) and their inverses

$$1^\dagger : \begin{array}{|c|c|} \hline A & \cdot \\ \hline x & y \\ \hline \end{array} \rightarrow \begin{array}{|c|c|} \hline \cdot & A \\ \hline x & y \\ \hline \end{array}, \qquad 2^\dagger : \begin{array}{|c|c|} \hline A & \blacktriangleright \\ \hline x & y \\ \hline \end{array} \rightarrow \begin{array}{|c|c|} \hline \blacktriangleright & A \\ \hline A^\dagger(x,y) & \\ \hline \end{array}. \tag{9}$$

Note that the inverse of the rule (8) involves applying $A^\dagger$ to the data qubits, uncomputing the gate. Our Hamiltonian is a sum of translationally invariant terms

$$H_{10} = -\sum_{j=1}^{L-1} \left(R + R^\dagger\right)_{(j,j+1)}, \tag{10}$$

where R corresponds to the rules (7, 8) and acts on two neighboring qudits as

$$R = \sum_{A \in \{W,S,I\}} \left[|A \,\cdot\rangle\langle \cdot\, A|_{p_1,p_2} \otimes \mathbb{I}_{d_1,d_2} + |A \,\blacktriangleright\rangle\langle \blacktriangleright\, A|_{p_1,p_2} \otimes A_{d_1,d_2} \right], \tag{11}$$

where p stands for the program register and d for the data register of the respective qudit. Similarly, $R^\dagger$ corresponds to the inverse rules given in (9).

3. The computational subspace

To show that we can obtain a result of a quantum computation by time evolving the state (4) with this Hamiltonian, we return to the set of states $\{|\varphi_\sigma\rangle\}$ generated using the rules (7, 8). There are only finitely many possible sequences σ. However, when we expand the ruleset by adding the inverse rules (9), the number of possible sequences σ of rule (forward and backward) applications then becomes infinite. For example, we get the possibility of returning back to the initial state $|\varphi\rangle$, completely undoing the computation. Nevertheless, the space of states $\{|\varphi_\sigma\rangle\}$ remains finite-dimensional for the following three reasons.

First, the rules (7, 8, 9) do not change the order of the gates in the program register. Second, once we know the locations of the gates, the rest of the program register is uniquely determined. It contains the sequence

$$\underbrace{\cdot \quad \cdot \quad \cdots \quad \cdot}_{N-1} \blacktriangleright \tag{12}$$

repeatedly written out K times from left to right at the positions unoccupied by the gates. Third, we constructed the initial state and the rules so that the gates in (6) are applied (and uncomputed) to the work qubits sequentially, while all the extra qubits remain in the state $|0\rangle$. Given the positions of the gates in the program register of $|\varphi_\sigma\rangle$, we can simply determine how many of the gates $U_{k,g}$ in (6) have presently been applied to the work qubits. For any $|\varphi_\sigma\rangle$, there can be at most one pointer $\blacktriangleright$ directly above the work qubits $w_1 \ldots w_N$.

1. If there is no pointer located above the work qubits (see Figure 2, 3a and 3c), label k_p the number of pointers already to the right of w_N. The first Nk_p gates then have already been applied.

2. If there is a pointer above the work qubits (see Figure 3b), all the gates to the left of this pointer have already been applied.

Let us look at a few examples of the first case. In Figure 2, $k_p = 0$ and thus no gates have been applied yet. In Figures 3a) and 3c) $k_p = 1$ and because $N = 3$, we know that the first 3 gates (IWS) have been executed. The state of the work qubits is thus equal to $S_{23}W_{12} |w_1 w_2 w_3\rangle$, in both Figure 3a) and 3c).

Therefore, regardless of the particular sequence of update rules (and their inverses) through which we obtained $|\varphi_\sigma\rangle$, the state of the data register is uniquely determined by the positions of the gates in the program register. In fact, all the states $|\varphi_\sigma\rangle$ with the same position of gates in the program register *are* the same state. Let us then label this state $|\varphi_C\rangle$, where C is a weight M bit string of length L. The M ones in the string C are located at positions

$$\{a_1^{(C)}, \ldots, a_M^{(C)}\}, \tag{13}$$

corresponding to the positions of the M gates in the program register of the state $|\varphi_\sigma\rangle$. The number of different states in the set $\{|\varphi_C\rangle\}$ is $\binom{L}{M}$. Note also that the states $|\varphi_C\rangle$ with different C are mutually orthogonal, as their corresponding program registers are in mutually orthogonal product states. The set $\{|\varphi_C\rangle\}$ thus forms a basis of an $\binom{L}{M}$-dimensional subspace of the Hilbert space of the qudit chain.

The transition rules in the Hamiltonian (10) correspond to (7, 8, 9). The time evolution of the initial state $|\varphi\rangle$ (4) will thus necessarily happen within

$$\mathcal{H}_{comp} = span\{|\varphi_C\rangle\}, \tag{14}$$

the subspace spanned by the states $|\varphi_C\rangle$. We call $\mathcal{H}_{comp}$ the computational subspace. In Section II B 1, we show that our Hamiltonian (10) restricted to the computational subspace has a particularly simple form corresponding to a Hamiltonian of a fermionic system. This will allow us to analyze the time evolution of the initial state.

B. Required Time Analysis

Our model of computation with the Hamiltonian (10) consists of initializing the qudit chain in the state $|\varphi\rangle$ (4) and evolving the system for a time τ randomly chosen between 0 and τ_{10}. In this Section, we will show that when we measure the output qubit w_N in the data register at time τ, we will read out the result of the quantum computation U with high probability. Moreover, we will show that the upper bound on the running time of our model, τ_{10}, is polynomial in L.

Each state $|\varphi_C\rangle$ described above can be written as

$$|\varphi_C\rangle = |\pi_C\rangle_{program} \otimes |\theta_C\rangle_{work} \otimes |\alpha\rangle_{extra}, \tag{15}$$

where $|\pi_C\rangle_{program}$ labels a state of the program register of the chain, the corresponding state of the work qubits is $|\theta_C\rangle_{work}$ and the extra data qubits are in the state $|\alpha\rangle_{extra} = |0\rangle_{extra}^{\otimes(2M-N)}$. Note that while each program register state $|\pi_C\rangle_{program}$ is a product state, the corresponding state of the work qubits $|\theta_C\rangle_{work}$ is entangled.

The time evolved state $|\varphi(\tau)\rangle$ is a superposition of the states $|\varphi_C\rangle$:

$$|\varphi(\tau)\rangle = e^{-iH_{10}\tau} |\varphi\rangle = \sum_C c_C(\tau) |\varphi_C\rangle = \sum_C c_C(\tau) |\pi_C\rangle_{program} \otimes |\theta_C\rangle_{work} \otimes |\alpha\rangle_{extra}, \tag{16}$$

where the sum over C is over all weight M length L bit strings and $c_C(\tau)$ is the amplitude of the state $|\varphi_C\rangle$ in $|\varphi(\tau)\rangle$.

When all of the gates in $|\pi_C\rangle_{program}$ have moved to the left of the work qubits, the state of the work qubits holds the output state of the computation U. For our original setup (4), there is only one such state:

$$|\varphi_{111111000000}\rangle = |\ I\ W\ S\ I\ S\ W\ \cdot\ \cdot\ \blacktriangleright\ \cdot\ \cdot\ \blacktriangleright\ \rangle_{program} \otimes U |w_1 \ldots w_N\rangle_{work} \otimes |\alpha\rangle_{extra}. \tag{17}$$

For our model of computation, this would mean that we would need a single amplitude $c_{1\ldots10\ldots0}(\tau)$ to be large at a random time τ, which is unlikely. However, a simple modification will turn the tables. Let us pad the qudit chain by $(f-1)M$ extra sites on the left and M extra sites on the right, with f a constant to be determined later. Let the new initial state be

$$|\varphi^{pad}\rangle = \boxed{\begin{array}{c} \cdot \\ \hline 0 \end{array}}^{\otimes(f-1)M} \otimes |\varphi\rangle \otimes \boxed{\begin{array}{c} I \\ \hline 0 \end{array}}^{\otimes M}, \tag{18}$$

where $|\varphi\rangle$ is the original initial state (4). The original chain had length $2M$, so the length of this padded chain is $L = (f+2)M$, while the number of gates becomes $2M$. As in Section II A 3, let us now label $\{|\varphi_D\rangle\}$ the set of states which we can obtain from (18) by applying rules (7, 8, 9). This time, the label D is a weight $2M$ length $L = (f+2)M$ bit string, and $\{a_1^{(D)}, \ldots, a_{2M}^{(D)}\}$ denote the positions of the $2M$ gates in the program register of $|\varphi_D\rangle$.

The $2M$ gates in the program register of the initial state $|\varphi^{pad}\rangle = |\varphi_{0\ldots01\ldots1}\rangle$ are located at positions $a_m^{(0\ldots01\ldots1)} = fM + m$ with $m = 1 \ldots 2M$. Only the first M gates are relevant for the computation and the other M are the identity gates we inserted as padding. These identity gates leave the data register intact, regardless of whether they move to the left of the work qubits or stay to the right of them. Therefore, every state $|\varphi_D\rangle$ in which the first M (relevant) gates have already moved to the left of the work qubits contains the finished computation in the state of its work qubits. Let us label the corresponding set of weight $2M$ length $L = (f+2)M$ bit strings

$$\mathcal{D}_{done} = \{D : a_M^{(D)} \leq fM\}. \tag{19}$$

For all states $|\varphi_D\rangle$ with $D \in \mathcal{D}_{done}$, the state of the work qubits is the same and equal to $|\theta_U\rangle = U |w_1 \ldots w_N\rangle$. Let us write the time evolved state of the modified chain using the notation (15) as

$$
\begin{aligned}
|\varphi^{pad}(\tau)\rangle &= e^{-iH_{10}\tau} |\varphi^{pad}\rangle \\
&= \left(\sum_{D \notin \mathcal{D}_{done}} c_D(\tau) |\varphi_D\rangle \right) + \left(\sum_{D \in \mathcal{D}_{done}} c_D(\tau) |\pi_D\rangle_{program} \right) \otimes |\theta_U\rangle_{work} \otimes |\alpha\rangle_{extra} \\
&= \sqrt{1 - p_{10}} \, |\varphi_{fail}(\tau)\rangle + \sqrt{p_{10}} \, |\pi_{done}(\tau)\rangle \otimes |\theta_U\rangle_{work} \otimes |\alpha\rangle_{extra}.
\end{aligned}
\tag{20} \tag{21}
$$

The state of the work qubits with the computation done $|\theta_U\rangle_{work}$ factorizes out in the second term. The two terms are orthogonal, as the states of the program register of $|\varphi_D\rangle$ with different D are mutually orthogonal. We will now analyze the amplitude of the second term. Assume we have a measurement discriminating different states $|\pi_D\rangle$ of the program register. In that case p_{10} denotes the probability of obtaining $|\pi_D\rangle$ with $D \in \mathcal{D}_{done}$ when measuring at a random time $\tau \leq \tau_{10}$.

1. Mapping onto a fermionic system

We will now prove that when we choose the time τ uniformly at random in $(0, \tau_{10})$, with $\tau_{10} = poly(M)$, the probability of finding a state with the computation executed (with $a_M \leq fM$) is $p_{10} \geq \frac{5}{6} - O\left(\frac{L}{\tau_{10}}\right)$ with $L = (f+2)M$. For this, we will analyze the time evolution of $|\varphi^{pad}\rangle$ (18) with H_{10} (10). We will do it by mapping the states of the qudit chain onto states of a chain of spin-$\frac{1}{2}$ particles, where the positions of gates in the qudit chain will correspond to fermionic excitations in the spin chain. By analyzing the latter system, we will bound p_{10}.

As in II A 3, the time evolution of the padded initial state (18) happens only in the computational subspace

$$\mathcal{H}_{comp}^{pad} = span\{|\varphi_D\rangle\}. \tag{22}$$

The basis states of the computational subspace have the form

$$|\varphi_D\rangle = |\pi_D\rangle_{program} \otimes \left(U^{(D)} |w_1 \ldots w_N\rangle_{work} \right) \otimes |\alpha\rangle_{extra}, \tag{23}$$

where the label D is a weight $2M$ bit string of length $L = (f+2)M$. The states $|\pi_D\rangle$ of the program register are canonical basis product states with

$$I U_{1,1} \ldots U_{1,N-1} \, I U_{2,1} \ldots U_{2,N-1} \, \cdots \, I U_{K,1} \ldots U_{K,N-1} \underbrace{I I \ldots I}_{M} \tag{24}$$

written out from left to right at the positions $\{a_1^{(D)}, \ldots, a_{2M}^{(D)}\}$ of the ones in D, and the rest of the program register is filled with

$$\underbrace{\centerdot \, \centerdot \centerdot \cdots \centerdot \, \centerdot}_{N-1} \blacktriangleright \tag{25}$$

written out fK times from left to right. Finally, $U^{(D)}$ stands for the part of the sequence of unitaries (24) that has already been applied to the work qubits when the gates are in positions $\{a_1^{(D)}, \ldots, a_{2M}^{(D)}\}$. We described the procedure to determine $U^{(D)}$ from D in Section II A 3.

We will now focus on the restriction of H_{10} to $\mathcal{H}_{comp}^{pad}$, as the time-evolved state (18) never leaves the computational subspace. This restriction has a simple form—it is the negative of an adjacency matrix, where two states are connected if one can be obtained from the other by one of the rules (7, 8, 9). More explicitly, two of the states $|\varphi_D\rangle$ are connected, if one can obtain the other by having one gate hop to the left (or right) and applying the appropriate unitary (or do nothing) to the work qubits. Consider now

$$H_q = -\sum_{j=1}^{L-1} (|10\rangle\langle 01| + |01\rangle\langle 10|)_{j,j+1}, \tag{26}$$

the Hamiltonian of a line of $L = (f+2)M$ spin-$\frac{1}{2}$ particles with a simple hopping interaction, restricted to the subspace spanned by states with $2M$ spin up particles. Observe that with the mapping

$$|\varphi_D\rangle_{\text{qudit chain}} \quad \leftrightarrow \quad |D\rangle_{\text{spin chain}}, \tag{27}$$

where D is a weight $2M$ bit string of length L, the restriction of H_{10} (10) to $\mathcal{H}_{comp}^{pad}$ (22) is matrix-wise the same as the restriction of H_q to the subspace with $2M$ up spins, i.e.

$$H_{10}|_{\mathcal{H}_{comp}^{pad}} \;=\; H_q|_{2M \text{ up spins}}. \tag{28}$$

To show that p_{10} in (21) is large, it will be convenient to analyze the time evolution of the initial state of the spin chain

$$|\underbrace{0\ldots0}_{fM}\underbrace{1\ldots1}_{2M}\rangle \tag{29}$$

with the spin Hamiltonian (26) instead of the time evolution of (18) with (10). First, we use the usual Wigner-Jordan transformation to define the operators

$$b_j^\dagger \;=\; \sigma_1^z\ldots\sigma_{j-1}^z \otimes |1\rangle\langle 0|_j \otimes \mathbb{I}_{j+1,\ldots,L}, \tag{30}$$
$$b_j \;=\; \sigma_1^z\ldots\sigma_{j-1}^z \otimes |0\rangle\langle 1|_j \otimes \mathbb{I}_{j+1,\ldots,L}. \tag{31}$$

We invite the reader to verify the properties $\{b_i, b_j^\dagger\} = \delta_{ij}\mathbb{I}$ and $b_j^2 = b_j^{\dagger 2} = 0$ for $b_j^\dagger$ and b_j. Thus, $b_j^\dagger$ and b_j can be viewed as the creation and annihilation operators for a fermion at site j. Rewriting (26) in terms of (30) and (31), we obtain

$$H_f = -\sum_{j=1}^{L-1} b_j^\dagger b_{j+1} + h.c., \tag{32}$$

a Hamiltonian for a system of free fermions in second quantization. Following our mapping, the initial state $|\varphi^{pad}\rangle$ of the qudit chain thus corresponds to the state of the fermionic system $|\Psi\rangle = b_{fM+1}^\dagger\ldots b_{(f+2)M}^\dagger|0\rangle$ with $2M$ fermions on the right end of the line (here $|0\rangle$ is the state with no fermions). We now use the following Lemma (proved in Appendix B):

Lemma 3. *Consider the state $|\Psi\rangle$ of $2M$ fermions on the right end of a line with $L = (f+2)M$ sites. Let the system evolve for a time chosen uniformly at random between 0 and τ_{10} with the Hamiltonian given by (32) and measure the number of fermions in the region $1 \leq x \leq fM$. The probability to measure a number greater than M is $p_{10} \geq \frac{f-2}{f+2} - O\left(\frac{L}{\tau_{10}}\right)$.*

Let us choose $f = 22$ and $\tau_{10} = O(L\log L) = O(M\log M)$. Following the mapping we did from our qudit chain backwards, this implies that when we initialize the qudit chain of length $L = 24M$ in $|\varphi^{pad}\rangle$ as in (18) and let it evolve with H_{10} (10) for a random time $\tau \leq \tau_{10}$, the probability for the chain to be in a state where the gate particles have moved sufficiently to the left for the computation to be done ($a_M \leq fM$) is

$$p_{10} \geq \frac{5}{6} - O\left(\frac{1}{\log M}\right). \tag{33}$$

Using this bound on the probability of having the computation done we now proceed to prove universality of our model for the class BQP.

2. Universality for BQP

To prove the universality of our model for recognizing languages in the class BQP, let us recall its definition. Consider a language L in BQP, a uniform family of circuits U and a problem instance x. When $x \in L_{yes}$, the set of instances with the answer 'yes', the probability of the circuit U outputting 'yes' is not smaller than $\frac{2}{3}$. On the other hand, when $x \notin L_{yes}$, the probability of the circuit outputting 'yes' is not greater than $\frac{1}{3}$. Let us assume the worst case for our circuit U, i.e. that the circuit outputs 'yes' on a good proof for a 'yes' instance with probability $p_U = \frac{2}{3}$. In the language of spins, the circuit U outputs 'yes' when we measure spin up on the output qubit. Therefore, the expected value of measuring σ_z on output qubit of the circuit U is bounded from below by

$$\langle \sigma_{w_N}^{(z)} \rangle_{yes}^{circuit} \geq 1 \times p_U + (-1) \times (1 - p_U) = 2p_U - 1 = \frac{1}{3} \tag{34}$$

when $x \in L_{yes}$. Analogously, for $x \in L_{no}$, it is bounded from above by

$$\langle \sigma_{w_N}^{(z)} \rangle_{no}^{circuit} \leq -2p_U + 1 = -\frac{1}{3}. \tag{35}$$

To solve BQP problems with our automaton, we need to distinguish the 'yes' from the 'no' cases, i.e. we need to show that the expectation value of measuring $\sigma^{(z)}$ on the output qubit of our automaton at a random time $\tau \leq \tau_{10}$ is greater than zero in the 'yes' case, and smaller than zero in the 'no' case. The probability of finding a state where the computation is finished is p_{10}. Let us consider a 'yes' instance ($x \in L_{yes}$). Using (21) and recalling that the two terms in it are orthogonal, we have

$$\langle \sigma_{w_N}^{(z)} \rangle_{yes} = \langle \varphi^{pad}(\tau) | \sigma_{w_N}^{(z)} | \varphi^{pad}(\tau) \rangle = (1 - p_{10}) \langle \varphi_{fail}(\tau) | \sigma_{w_N}^{(z)} | \varphi_{fail}(\tau) \rangle + p_{10} \underbrace{\langle \theta_U | \sigma_{w_N}^{(z)} | \theta_U \rangle_{work}}_{\text{output of } U}. \tag{36}$$

The second term is the circuit output (34), therefore

$$\langle \theta_U | \sigma_{w_N}^{(z)} | \theta_U \rangle_{work} = \langle \sigma_{w_N}^{(z)} \rangle_{yes}^{circuit} \geq 2p_U - 1. \tag{37}$$

The first term can be bounded from below (adversarially, i.e. for every time the computation is not finished, the output qubit gives the opposite of the correct answer) by

$$\langle \varphi' | \sigma_{w_N}^{(z)} | \varphi' \rangle \geq -1. \tag{38}$$

Putting it together, we have

$$\langle \sigma_{w_N}^{(z)} \rangle_{yes} \geq p_{10}(2p_U - 1) - (1 - p_{10}) = 2p_{10}p_U - 1. \tag{39}$$

Analogously, for the $x \notin L$ case, we obtain

$$\langle \sigma_{w_N}^{(z)} \rangle_{no} \geq -2p_{10}p_U + 1. \tag{40}$$

Therefore, equations (39) and (40) now read

$$\langle \sigma_{w_N}^{(z)} \rangle_{yes} \geq \frac{1}{9} - O\left(\frac{1}{\log M}\right),$$
$$\langle \sigma_{w_N}^{(z)} \rangle_{no} \leq -\frac{1}{9} + O\left(\frac{1}{\log M}\right). \tag{41}$$

Therefore, we can recognize any language in BQP using the HQCA we described above.

As an aside, note that there is a way to determine that we obtained a state in which the computation has been done with certainty (and thus getting rid of the first term in (36)). We could have chosen to measure all the program qudits to the right of the first work qubit and check whether all the S and W are gone. This happens with the above probability p_{10}, and the postselected state of the work qubits now surely contains the output of the circuit U. Note also that we can think of the state of all the work qubits as the circuit output, as compared to only the last work qubit. Nevertheless, thinking only about the last work qubit is enough to ensure universality of our HQCA for the class BQP.

Concurrently with our preprint, Chase and Landahl [23] found a $d = 8$ Hamiltonian computer construction universal for BQP. However, their Hamiltonian is not translationally invariant, and thus not a HQCA. If we release the translational invariance in our $d = 10$ HQCA model, we can be sure where the computational qubits are and use the gate set $\{S, SW\}$ instead of $\{S, W, I\}$ in our program register. This brings the required dimensionality of our model to $d = 8$ as well. However, it remains an open question whether this can be done with keeping the translational invariance of the Hamiltonian, i.e. whether a $d \leq 10$ HQCA in 1D exists.

III.   THE $d = 20$ HQCA

We now present our second construction, a HQCA for a chain of 20-dimensional qudits. As in Section II A, we describe an encoding of the progression of a quantum circuit U into a set of states of a qudit chain. However, the geometry of this set of states $|\psi_t\rangle$ will be now much simpler, as we can label them by a "time" label $t = 1, \ldots, T$ with $T = poly(N)$, thinking of the set of states as a "line". The Hamiltonian H_{20} we construct induces a quantum walk on this "line" of states. We conclude by proving that when we let the initial state $|\psi_0\rangle$ evolve with H_{20} for a time τ chosen uniformly at random between 0 and $\tau_{20} = O(T \log T)$, we can read out the result of the quantum computation U with probability $p_{20} \geq \frac{5}{6} - O\left(\frac{T}{\tau_{20}}\right)$ by measuring two of the qudits in the computational basis.

A.   The Construction

We encode the progression of a quantum circuit U in the form (2) (see also Figure 1) into a set of states $|\psi_t\rangle$ of a qudit chain with length $L = (2K - 1)(N + 1) + 2$. As in Section II A, each qudit consists of a program register and a data register. The data register is again two-dimensional, but the program register can now be in the following 10 states:

$$
\begin{aligned}
W, S, I \quad &: \quad \text{the program sequence,} \\
Ⓦ, Ⓢ, Ⓘ \quad &: \quad \text{marked characters in the program sequence, used to propagate} \\
&\qquad \text{the active spot to the front (left) of the program sequence,} \\
\blacktriangleright \quad &: \quad \text{apply gate symbol,} \\
\triangleright \quad &: \quad \text{shift program forward,} \\
\circlearrowleft \quad &: \quad \text{a turn-around symbol,} \\
\centerdot \quad &: \quad \text{empty spot (before/after the program).}
\end{aligned}
$$

Similarly to (3) and (4), the initial product state $|\psi_0\rangle = \bigotimes_{j=1}^{L} \left(|p_j\rangle \otimes |d_j\rangle\right)_j$ is given by (we write an example for the circuit in Figure 1a)

$$
\begin{array}{c|ccccccccccccc}
j & 1 & \cdots & & & & & & & & & \cdots & L \\
\hline
p_j & \centerdot & \centerdot & \centerdot & \centerdot & \centerdot & \centerdot & I & W & S & I & I & S & W & \circlearrowleft \\
d_j & 0 & 1 & 0 & 0 & 0 & 1 & w_1 & w_2 & w_3 & 1 & 0 & 0 & 0 & 1
\end{array}
\tag{42}
$$

In general, the data register contains N work qubits (labeled w_n in our example) at positions $(K-1)(N+1)+2+n$ for $n = 1 : N$ (counting from the left). Qubit w_N is the designed output qubit for the computation, i.e. once the computation is done, w_N contains the output of U. Next, the data register contains qubits in the state $|1\rangle$ at positions $(k-1)(N+1)+2$ for $k = 1 \ldots 2K$ and qubits in the state $|0\rangle$ everywhere else. The 1's serve as sequence boundary markers. The program register has empty symbols $\centerdot$ on the left, and then it contains the program in the form

$$
I \underbrace{U_{1,1} \ldots U_{1,N-1}}_{1^{\text{st}} \text{ gate sequence}} I I \underbrace{U_{2,1} \ldots U_{2,N-1}}_{2^{\text{nd}} \text{ gate sequence}} I I \cdots I I \underbrace{U_{K,1} \ldots U_{K,N-1}}_{\text{last gate sequence}},
\tag{43}
$$

with the program written from left to right. In our example (42), the first gate sequence (see Figure 1a) is WS and the second gate sequence is SW. Finally, the last qudit in the program register is in the state $\circlearrowleft$, marking an active spot in the computation.

We now give the rules to obtain the sequence of states $|\psi_t\rangle$ from $|\psi_0\rangle$. These rules are constructed so that there is always only one of them that can be applied to a given state $|\psi_t\rangle$, thus giving us a unique state $|\psi_{t+1}\rangle$. (Also, using the rules backwards, one obtains a unique $|\psi_{t-1}\rangle$ from $|\psi_t\rangle$). The first three are

$$
\begin{array}{ll}
1 : & \boxed{A \mid \circlearrowleft} \;\rightarrow\; \boxed{Ⓐ \mid \centerdot} \\[4pt]
2 : & \boxed{A \mid Ⓑ} \;\rightarrow\; \boxed{Ⓐ \mid B} \\[4pt]
3 : & \boxed{\centerdot \mid Ⓐ} \;\rightarrow\; \boxed{\circlearrowleft \mid A}
\end{array}
\tag{44}
$$

where A, B stands for either W, S or I. These rules ensure the passing of the active spot from the back end (right side) of the program to the front (left side), without modifying the data register or the order of the gates in the program

sequence. Next, we have

$$4a \; : \quad \boxed{\begin{array}{c} \cdot \;\; \circlearrowleft \\ \hline 1 \end{array}} \quad \rightarrow \quad \boxed{\begin{array}{c} \cdot \;\; \blacktriangleright \\ \hline 1 \end{array}} \qquad 4b \; : \quad \boxed{\begin{array}{c} \cdot \;\; \circlearrowleft \\ \hline 0 \end{array}} \quad \rightarrow \quad \boxed{\begin{array}{c} \cdot \;\; \triangleright \\ \hline 0 \end{array}} \tag{45}$$

After the active spot has moved to the front of the program, there are two possibilities. The turn symbol $\circlearrowleft$ can change to the apply gate symbol $\blacktriangleright$ (rule 4a), or to the shift program symbol $\triangleright$ (rule 4b), depending on whether the data qubit below contains the sequence boundary marker state 1. Afterward, for the states containing the apply gate symbol $\blacktriangleright$, we have:

$$5a \; : \quad \boxed{\begin{array}{cc} \blacktriangleright & A \\ \hline x & y \end{array}} \quad \rightarrow \quad \boxed{\begin{array}{cc} A & \blacktriangleright \\ \hline A(x,y) \end{array}} \qquad 6a \; : \quad \boxed{\begin{array}{cc} \blacktriangleright & \cdot \\ \hline 1 \end{array}} \quad \rightarrow \quad \boxed{\begin{array}{cc} \circlearrowleft & \cdot \\ \hline 1 \end{array}} \tag{46}$$

(Note that rule 6a at the right end of the chain involves only the two particles directly above each other, as no particle to the right of them exists.) When applying rule 5a, the apply gate symbol $\blacktriangleright$ moves to the right, while a gate from the program sequence is applied to the qubits in the data register below. Applying the rule repeatedly, the $\blacktriangleright$ symbol moves to the right end of the program sequence. As an example, we now write out the state $|\psi_{12}\rangle$ that we obtained from the state $|\psi_0\rangle$ applying rules 1, 2 (6 times), 3, 4a and 5a (3 times) from the state $|\psi_0\rangle$.

$$|\psi_{12}\rangle \;\; = \;\; \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & I & W & S & \blacktriangleright & I & I & S & W & \cdot \\ 0 & 1 & 0 & 0 & 0 & 1 & | \dots \theta \dots \rangle & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{47}$$

where $|\dots\theta\dots\rangle$ stands for the state of the three work qubits after the gates W_{12} and then S_{23} were applied to them. Let us have a closer look at the marker qubits (all qubits in the data register except for the work qubits w_n) and the application of rule 5a. The marker qubits stay unchanged for all $|\psi_t\rangle$. The gate applied to pairs $|0\rangle\,|1\rangle$ and $|1\rangle\,|0\rangle$ of marker qubits or the pairs of qubits $|1\rangle\,|q_1\rangle$ and $|q_N\rangle\,|1\rangle$ (the left and right ends of the work qubit sequence) is always I, because of the identity gates we inserted between sequences of gates in the program (43). Finally, the qubit pairs $|0\rangle\,|0\rangle$ between the 1 markers do not change under the swap operation or the W gate (a controlled gate).

After the apply gate $\blacktriangleright$ symbol gets to the end of the sequence, it changes into the turn symbol $\circlearrowleft$ via rule 6a. Note that the boundary markers in the data register are spaced in such a way, that the $\blacktriangleright$ symbol will arrive at the right end of the sequence when the qubit below is in the state 1. Using rule 6a, $\blacktriangleright$ will then change into the turn symbol $\circlearrowleft$. After applying rules 1, 2 (6 times) and 3, the active spot again moves to the left of the program. Because the $\circlearrowleft$ symbol is now above a 0 marker qubit, rule 4b can be used, and we get a state with the shift program symbol $\triangleright$. Finally, here are the last two rules:

$$5b \; : \quad \boxed{\begin{array}{cc} \triangleright & A \end{array}} \quad \rightarrow \quad \boxed{\begin{array}{cc} A & \triangleright \end{array}} \qquad 6b \; : \quad \boxed{\begin{array}{cc} \triangleright & \cdot \\ \hline 0 \end{array}} \quad \rightarrow \quad \boxed{\begin{array}{cc} \circlearrowleft & \cdot \\ \hline 0 \end{array}} \tag{48}$$

where again A stands for either W, S or I. Rule 5b makes the program shift to the left while the $\triangleright$ symbol moves to the right. Finally, rule 6b deals with what happens when the $\triangleright$ symbol arrives at the end of the program sequence. Because of the way we constructed the data register in $|\psi_0\rangle$, the data qubit below the $\triangleright$ symbol will then be in the state 0, so that the $\triangleright$ symbol changes to the turn symbol $\circlearrowleft$. The reason why we need to look at the qubit in the data register below the $\circlearrowleft$ symbol in rules 6a and 6b is that when we apply the rules backwards (making $|\psi_{t-1}\rangle$ from $|\psi_t\rangle$), again only one of them applies for each $|\psi_t\rangle$.

After applying rule 1, 2a (6 times), 3 and 4b, the $\triangleright$ symbol appears again and starts shifting the program further to the left. After several rounds of this, when the program shifts to the left by $N+1$, rule 4a can be used again (as the $\circlearrowleft$ symbol will be above a 1 marker qubit), and subsequently, the $\blacktriangleright$ symbol facilitates the application of the second sequence of gates to the work qubits.

After many applications of the above rules, we arrive at the state $|\psi_T\rangle$, for which none of our (forward) rules apply.

$$|\psi_T\rangle \;\; = \;\; \begin{bmatrix} \circlearrowleft & I & W & S & I & I & S & W & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 1 & 0 & 0 & 0 & 1 & | \dots \theta' \dots \rangle & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{49}$$

This is the state in which the program has moved to the left of the qudit chain, and all sequences of gates have been applied to the qubits in the data register. The state $|\dots\theta'\dots\rangle$ is thus the output state of the circuit U and the last of the work qubits (w_N) holds the output of the quantum computation.

Starting from (42), we have constructed the set of states $|\psi_t\rangle$ for $t = 0 \dots T$ with $T = O(K^2 N^2) = poly(N)$. As t grows, these states encode the progress of a quantum circuit U. Let us now think of the geometry of this set of states.

They are labeled by a discrete label t, with the state $|\psi_t\rangle$ obtainable only from the states $|\psi_{t-1}\rangle$ and $|\psi_{t+1}\rangle$ using the above rules and their backward applications. Therefore, the states $|\psi_t\rangle$ can be thought of as position basis states on a line of length $T+1$

$$|\psi_t\rangle \quad \leftrightarrow \quad |t\rangle_{line}, \tag{50}$$

where $t = 0 \ldots T$.

Let us choose a Hamiltonian H_{20} for this system as a sum of translationally invariant terms:

$$H_{20} = -\sum_{i=1}^{L-1}\sum_{k=1}^{6b}\left(P_k + P_k^\dagger\right)_{(i,i+1)} \tag{51}$$

where the terms P_k correspond to the rules 1-6b (44),(45),(46) and (48) and act on two neighboring qudits as

$$P_1 = \sum_{A\in\{W,S,I\}} |Ⓐ\ ▪\rangle\langle A\ ↺|_{p_1,p_2}\otimes\mathbb{I}_{d_1,d_2}, \tag{52}$$

$$P_2 = \sum_{A,B\in\{W,S,I\}} |Ⓐ\ B\rangle\langle A\ Ⓑ|_{p_1,p_2}\otimes\mathbb{I}_{d_1,d_2}, \tag{53}$$

$$P_3 = \sum_{A\in\{W,S,I\}} |↺\ A\rangle\langle ▪\ Ⓐ|_{p_1,p_2}\otimes\mathbb{I}_{d_1,d_2}, \tag{54}$$

and

$$P_{4a} = |\ ▪\ ▶\rangle\langle\ ▪\ ↺|_{p_1,p_2}\otimes\mathbb{I}_{d_1}\otimes|1\rangle\langle1|_{d_2}, \tag{55}$$

$$P_{4b} = |\ ▪\ ▷\rangle\langle\ ▪\ ↺|_{p_1,p_2}\otimes\mathbb{I}_{d_1}\otimes|0\rangle\langle0|_{d_2}, \tag{56}$$

$$P_{5a} = \sum_{A\in\{W,S,I\}} |A\ ▶\rangle\langle▶\ A|_{p_1,p_2}\otimes A_{d_1,d_2}, \tag{57}$$

$$P_{5b} = \sum_{A\in\{W,S,I\}} |A\ ▷\rangle\langle▷\ A|_{p_1,p_2}\otimes\mathbb{I}_{d_1,d_2}, \tag{58}$$

$$P_{6a} = |↺\ ▪\rangle\langle▶\ ▪|_{p_1,p_2}\otimes|1\rangle\langle1|_{d_1}\otimes\mathbb{I}_{d_2}, \tag{59}$$

$$P_{6b} = |↺\ ▪\rangle\langle▷\ ▪|_{p_1,p_2}\otimes|0\rangle\langle0|_{d_1}\otimes\mathbb{I}_{d_2}. \tag{60}$$

When thinking of the set of states $|\psi_t\rangle$ as the set of positions of a particle on a line (50), H_{20} becomes

$$H_{line} = -\sum_{t=0}^{T-1}\left(|t\rangle\langle t+1| + |t+1\rangle\langle t|\right). \tag{61}$$

This is the Hamiltonian of a (continuous-time) quantum walk on a line of length $T+1$. Therefore, H_{20} induces a quantum walk on the "line" of states $|\psi_t\rangle$ of the qudit chain of length L.

B. Required Evolution Time Analysis

The final step of our model of computation after initializing the qudit chain in the state $|\psi_0\rangle$ and evolving with H_{20} for time τ is to read out the output of the computation. As in Section II B, we need to ensure that the probability of finding the chain of qudits in a state where the computation was performed completely is high. To raise this probability, we choose to pad the program (K sequences of gates) with another $5K$ sequences of identity gates and redo the construction in the previous section. The length of the qudit chain thus becomes $L = (2(6K)-1)(N+1)+2$. The states $|\psi_{t>T/6}\rangle$ (with T modified) now all contain the result of the quantum circuit U in the readout qubit w_N, as the relevant gates have been applied to the work qubits in those states. Note that as the extra identity gates pass by, the state of the work qubits does not change.

The readout procedure consists of two steps. First, measure the qudit $p_{L-K(N+1)}$ in the program register (the qudit with distance from the right end of the chain equal to the length of the original program). Let us call p_{20} the probability to measure ▪ (which would mean the program has moved to the left of the qudit we just measured). When this happens, we are assured we have a state in which the computation was done. Second, we measure w_N,

the last of the work qubits, and read out the result of the computation U. We will now prove that when we choose to measure $p_{L-K(N+1)}$ at a random time $0 \leq \tau \leq \tau_{20}$ with $\tau_{20} = poly(N)$, the probability p_{20} of obtaining the state $\blacksquare$ is close to $\frac{5}{6}$.

To simplify the notation, let us label the states $|\psi_t\rangle$ as $|t\rangle$. In this basis, the Hamiltonian (51) is the negative of the adjacency matrix of a line graph with $T + 1$ nodes. For the analysis of time evolution with H we refer the reader to Appendix A. We now use the following lemma about a quantum walk on a line (proved in Appendix A):

Lemma 2. *Consider a continuous time quantum walk on a line of length $T+1$, where the Hamiltonian is the negative of the adjacency matrix for the line. Let the system evolve for a time τ chosen uniformly at random between 0 and τ_{20}, starting in a position basis state $|c\rangle$. The probability to measure a state $|t\rangle$ with $t > T/6$ is then $p_{20} \geq \frac{5}{6} - O\left(\frac{T+1}{\tau_{20}}\right)$.*

This implies that when we initialize the qudit chain in the state $|\psi_0\rangle$ (corresponding to the leftmost state on the line $|c\rangle = |1\rangle$) and let it evolve with H for a random time $\tau \leq \tau_{20}$ with $\tau_{20} = O(T \log T)$, the probability to find a state with $t > T/6$ is close to $\frac{5}{6}$. Therefore, when we measure the program qudit $p_{L-K(N+1)}$, we will obtain $\blacksquare$ with probability close to $\frac{5}{6}$. Finally, when we subsequently measure the work qubit w_N, we will obtain the result of the quantum circuit U.

Note that we can also avoid this postselection procedure and simply measure the output qubit. The analysis of the outcome would then follow what we did above in Section II B, resulting in (41) again, with M replaced by T.

APPENDIX A: QUANTUM WALK ON A LINE

Here we analyze the quantum walk on a line and prove two useful lemmas used in Section III B and Appendix B.

Consider a continuous time quantum walk on a line of length L, where the Hamiltonian is the negative of the adjacency matrix for the line

$$H_1 = -\sum_{j=1}^{L-1} \left(|j\rangle \langle j+1| + |j+1\rangle \langle j|\right). \tag{A1}$$

The eigenvalues of this Hamiltonian are

$$\lambda_j = -2\cos\left(\frac{j\pi}{L+1}\right), \tag{A2}$$

for $j = 1 \ldots L$, while the corresponding eigenvectors $\left|\phi^{(j)}\right\rangle = \sum_{k=1}^{L} \phi_k^{(j)} |k\rangle$ have components

$$\phi_k^{(j)} = \sqrt{\frac{2}{L+1}} \sin\left(\frac{jk\pi}{L+1}\right). \tag{A3}$$

Consider the time evolution of a particular basis state $|c\rangle$. The probability of finding the system in a basis state $|m\rangle$ at some time τ can be found by expanding $|c\rangle$ and $|m\rangle$ in the basis of the eigenvectors (A3):

$$p_\tau(m|c) = \left|\langle m| e^{-iH\tau} |c\rangle\right|^2 = \sum_{j,k=1}^{L} e^{-i(\lambda_j - \lambda_k)\tau} \phi_m^{(j)} \phi_c^{(j)*} \phi_m^{(k)*} \phi_c^{(k)}. \tag{A4}$$

Because the time evolution (according to the Schrödinger equation) is unitary, this probability $p_\tau(m|c)$ does not converge. On the other hand, let us define the time average of $p_\tau(m|c)$ for time $0 \leq \tau \leq \tau_{20}$ as

$$\bar{p}_{\tau_{20}}(m|c) = \frac{1}{\tau_{20}} \int_0^{\tau_{20}} p_\tau(m|c)\, d\tau. \tag{A5}$$

As we will show below in Lemma 1, this average probability distribution does converge to a limiting distribution $\pi(m|c)$, defined as the $\tau_{20} \to \infty$ limit of the average probability distribution (A5). All the eigenvalues (A2) are different, so we can express the limiting distribution as

$$\pi(m|c) = \lim_{\tau_{20} \to \infty} \bar{p}_{\tau_{20}}(m|c) = \sum_{j=1}^{L} \left|\phi_m^{(j)}\right|^2 \left|\phi_c^{(j)}\right|^2, \tag{A6}$$

which in our case is

$$\pi(m|c) = \frac{2 + \delta_{m,c} + \delta_{m,L+1-c}}{2(L+1)}. \tag{A7}$$

According to the following lemma, the average probability (A5) converges to the limiting distribution $\pi(m|c)$.

Lemma 1. *Consider a continuous time quantum walk on a line of length L, where the Hamiltonian is the negative of the adjacency matrix for the line. Let the system evolve for time $\tau \leq \tau_{20}$ chosen uniformly at random, starting in a position basis state $|c\rangle$. The average probability distribution $\bar{p}_{\tau_{20}}(\cdot|c)$ converges to the limiting probability distribution $\pi(\cdot|c)$ as*

$$\sum_{m=1}^{L} |\bar{p}_{\tau_{20}}(m|c) - \pi(m|c)| \leq O\left(\frac{L}{\tau_{20}}\right). \tag{A8}$$

Proof. To prove our Lemma 1, we recall Lemma 4.3 of [1] for the total variation distance of the probability distribution $\bar{p}_{\tau_{20}}$ from the limiting distribution, saying

$$\sum_m |\bar{p}_{\tau_{20}}(m|c) - \pi(m|c)| \leq \frac{2}{\tau_{20}} \sum_{j \neq k} \frac{\left|\phi_c^{(j)}\right|^2}{|\lambda_j - \lambda_k|}. \tag{A9}$$

Using (A2) and (A3), we can bound the expression on the right of (A9). When j is close to k, i.e. $|j - k| \leq C_1$, we can obtain

$$\frac{\left|\phi_c^{(j)}\right|^2}{|\lambda_j - \lambda_k|} < 2. \tag{A10}$$

On the other hand, for $|j - k| > C_1$ we can write

$$\frac{\left|\phi_c^{(j)}\right|^2}{|\lambda_j - \lambda_k|} < \frac{C_2}{L+1}, \tag{A11}$$

with C_1 and C_2 constants independent of L. Inserting into (A9), we have

$$\sum_{m=1}^{L} |\bar{p}_{\tau_{20}}(m|c) - \pi(m|c)| \leq \frac{8C_1 L}{\tau_{20}} + \frac{C_2 L}{\tau_{20}} = O\left(\frac{L}{\tau_{20}}\right), \tag{A12}$$

which concludes the proof. □

Using Lemma 1, we will now prove a useful result utilized in the time analysis of the $d = 20$ HQCA in Section (III).

Lemma 2. *Consider a continuous time quantum walk on a line of length L, where the Hamiltonian is the negative of the adjacency matrix for the line. Let the system evolve for a time $\tau \leq \tau_{20}$ chosen uniformly at random, starting in a position basis state $|c\rangle$. The probability to measure a state $|t\rangle$ with $t > L/6$ is then bounded from below as $p_{20} \geq \frac{5}{6} - O\left(\frac{L}{\tau_{20}}\right)$.*

Proof. The probability to measure a state $|t\rangle$ with $t > L/6$ at time $\tau \leq \tau_{20}$ chosen uniformly at random is

$$p_{20} = \sum_{m > \frac{L}{6}} \bar{p}_{\tau_{20}}(m|c). \tag{A13}$$

Starting with (A8), we have

$$O\left(\frac{L}{\tau_{20}}\right) \geq \sum_{m=1}^{L} |\bar{p}_{\tau_{20}}(m|c) - \pi(m|c)| \tag{A14}$$

$$\geq \sum_{m>\frac{L}{6}} |\bar{p}_{\tau_{20}}(m|c) - \pi(m|c)| \tag{A15}$$

$$\geq \left| \sum_{m=\frac{5L}{6}}^{L} \bar{p}_{\tau_{20}}(m|c) - \sum_{m>\frac{L}{6}} \pi(m|c) \right| \tag{A16}$$

$$= \left| p_{20} - \frac{5}{6} + O\left(\frac{1}{L}\right) \right|. \tag{A17}$$

Therefore, the probability of finding the chain in state $|\psi_{t>L/6}\rangle$ at a random time $\tau \leq \tau_{20}$ is thus bounded from below by

$$p_{20} \geq \frac{5}{6} - O\left(\frac{L}{\tau_{20}}\right). \tag{A18}$$

$\square$

APPENDIX B: DIFFUSION OF FERMIONS ON A LINE

We now prove Lemma 3, a result about the mixing of a discrete free fermion gas.

Lemma 3. *Consider the state*

$$|\Psi_0\rangle = b_{fM+1}^\dagger b_{fM+2}^\dagger \ldots b_{fM+2M}^\dagger |0\rangle. \tag{B1}$$

of $2M$ fermions on the right end of a line with $L = (f+2)M$ sites. Let the system evolve for a time chosen uniformly at random between 0 and τ_{10} with the Hamiltonian

$$H_f = -\sum_{j=1}^{L-1} b_j^\dagger b_{j+1} + h.c. \tag{B2}$$

and measure the number of fermions in the region $1 \leq x \leq fM$. The probability to measure a number greater than M is $p_{10} \geq \frac{f-2}{f+2} - O\left(\frac{L}{\tau_{10}}\right)$.

Proof. Let us start with the outline of the proof. We look at the fermionic system in both first and second quantization to obtain an expression for the time evolution of the creation and annihilation operators in the Heisenberg picture, mapping it to a quantum walk on a line. We then consider the observable X, the number of particles sufficiently far from the right end of the line. We will show that when we choose the time to measure X uniformly at random between 0 and τ_{10}, the expected value we will obtain is approaching a number close to $2M$. To show this, we will express the expected value of X in the time-averaged state of the system using the results from a quantum walk on a line. Finally, because the number of particles in the system is $2M$, we will deduce that the probability to measure a number less than M is then small.

Observe that H_f is the Hamiltonian of a free fermion gas on a line in second quantization (a special case of the XY model). The time evolution of the state $|\Psi_0\rangle$ can be obtained by looking at the problem back in the first quantization, where we write $|\Psi_0\rangle$ as

$$|\Psi_0\rangle = \left[|\phi_{fM+1}\rangle \otimes |\phi_{fM+2}\rangle \otimes \cdots \otimes |\phi_{fM+2M}\rangle \right]^-, \tag{B3}$$

with $|\phi_j\rangle = |j\rangle$ in the position basis and $[\,\cdot\,]^-$ the standard antisymmetrization operator. We first solve for the time evolution of the corresponding one-particle wavefunction $|\phi_j(\tau)\rangle$ with the Hamiltonian

$$H_1 = -\sum_{j=1}^{L-1} (|j\rangle \langle j+1| + |j+1\rangle \langle j|), \tag{B4}$$

and then obtain the solutions for the many-particle problem by antisymmetrization as

$$|\Psi(\tau)\rangle = \left[|\phi_{fM+1}(\tau)\rangle \otimes |\phi_{fM+2}(\tau)\rangle \otimes \cdots \otimes |\phi_{fM+2M}(\tau)\rangle \right]^{-}. \tag{B5}$$

The eigenfunctions of H_1 (quantum walk on a line) are plain waves (as in (A2) and (A3)), and the time evolved states $|\phi_j(\tau)\rangle$ thus readily available. Let us define the unitary matrix $u(\tau)$ by

$$|j(\tau)\rangle = \sum_{k=1}^{L} u_{jk}(\tau) |k\rangle. \tag{B6}$$

Returning to the second quantized system, the time evolution of the creation and annihilation operators in the Heisenberg picture is then

$$\begin{aligned} b_j^{\dagger}(\tau) &= \sum_{k=1}^{L} u_{jk}(\tau) b_k^{\dagger}, \\ b_j(\tau) &= \sum_{k=1}^{L} u_{jk}^{*}(\tau) b_k. \end{aligned} \tag{B7}$$

Consider now the observable X, the number of particles in the first fM sites of the line with length $L = (f+2)M$

$$X = \sum_{m=1}^{fM} \hat{n}_m. \tag{B8}$$

Its expectation value at time τ is

$$E_\tau(X) = \sum_{m=1}^{fM} \langle \Psi(\tau)| \hat{n}_m |\Psi(\tau)\rangle. \tag{B9}$$

The number operator for site m is $\hat{n}_m = b_m^{\dagger} b_m$. We can go to the Heisenberg picture and use (B7) to write

$$\begin{aligned} \langle \Psi(\tau)| \hat{n}_m |\Psi(\tau)\rangle &= \langle \Psi_0| b_m^{\dagger}(\tau) b_m(\tau) |\Psi_0\rangle \tag{B10} \\ &= \sum_{c=1}^{L} \sum_{d=1}^{L} u_{mc}(\tau) u_{md}^{*}(\tau) \langle \Psi_0| b_c^{\dagger} b_d |\Psi_0\rangle \tag{B11} \\ &= \sum_{c=1}^{L} |u_{mc}(\tau)|^2 \langle \Psi_0| b_c^{\dagger} b_c |\Psi_0\rangle \tag{B12} \\ &= \sum_{c=fM+1}^{L} \underbrace{|u_{mc}(\tau)|^2}_{p_\tau(m|c)}, \tag{B13} \end{aligned}$$

where each term $|u_{mc}(\tau)|^2 = p_\tau(m|c)$ can be thought of as the probability of finding a particle at site m at time τ when it started from the site c and performed a quantum walk on a line, according to (B4). Inserting this into (B9), the expected number of particles not in the rightmost part of the chain at time τ is

$$E_\tau(X) = \sum_{c=fM+1}^{L} \left(\sum_{m=1}^{fM} p_\tau(m|c) \right). \tag{B14}$$

Let us now choose the time τ uniformly at random between 0 and τ_{10}. The average value of X (the expectation value in the time-average state) is

$$\bar{E}_{\tau_{10}}(X) = \frac{1}{\tau_{10}} \int_0^{\tau_{10}} E_\tau(X) \, d\tau. \tag{B15}$$

For a quantum walk on a line, the time-averaged probability (A5) of finding a particle that started at position c at final position m converges to the limiting distribution (A7) according to Lemma 1 (A8) proved in Appendix A.

Using this fact, we can show that the expectation value $\bar{E}_{\tau_{10}}(X)$ in the time-averaged state converges to the limiting expectation value

$$\bar{E}(X) = \sum_{m \leq fM} \sum_{c > fM} \pi(m|c) \tag{B16}$$

as

$$\left| \bar{E}_{\tau_{10}}(X) - \bar{E}(X) \right| \leq O\left(\frac{LM}{\tau_{10}}\right). \tag{B17}$$

Recalling the limiting probability distribution for a quantum walk on a line of length L (A7), we have

$$\bar{E}(X) = \sum_{m \leq fM} \sum_{c > fM} \pi(m|c) \tag{B18}$$

$$= fM \times 2M \times \frac{2}{2(L+1)} + 2M \times \frac{1}{2(L+1)} \tag{B19}$$

$$= 2M\left(\frac{f}{f+2}\right) + O(1). \tag{B20}$$

Putting this into (B17), the average value of X when the time $\tau \leq \tau_{10}$ is chosen uniformly at random is bounded from below as

$$\bar{E}_{\tau_{10}}(X) \geq 2M\left(\frac{f}{f+2}\right) - O\left(\frac{LM}{\tau_{10}}\right). \tag{B21}$$

We want to find the probability of measuring $X > M$. First, the maximum possible value we could measure at any time is $2M$, the number of particles in the system. Second, the average value $\bar{E}_{\tau_{10}}(X)$ at time τ chosen randomly is close to $2M$. Therefore, the fraction Δ of times at which we measure a number significantly lower than $2M$ must be small. Let us bound Δ in the worst case scenario. This is when each unsuccessful measurement yields $X = M$, and each successful measurement gives us $2M$. We then have

$$\Delta M + (1 - \Delta)2M \geq \bar{E}_{\tau_{10}}(X), \tag{B22}$$

$$\Delta \leq \frac{2\bar{E}_{\tau_{10}}(X) - M}{M}. \tag{B23}$$

Hence we arrive at the desired bound on the probability to measure $X > M$:

$$p_{10} = 1 - \Delta \geq \frac{2M\left(\frac{f}{f+2}\right) - O\left(\frac{LM}{\tau_{10}}\right) - M}{M} = \frac{f-2}{f+2} - O\left(\frac{L}{\tau_{10}}\right). \tag{B24}$$

$\square$

---

[1] D. Aharonov, A. Ambainis, J. Kempe, U. Vazirani, Quantum Walks on Graphs, In *Proc. of ACM Symposium on Theory of Computation* (STOC'01), p. 50-59, 2001.

[2] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, in *Proc. 45th IEEE Symp. on Foundations of Computer Science* (FOCS), p. 42 (2004).

[3] D. Aharonov, D. Gottesman, S. Irani, J. Kempe, The power of quantum systems on a line, *Proc. 48th IEEE Symposium on the Foundations of Computer Science* (FOCS), 373-383 (2007)

[4] P. Benioff, The computer as a physical system: a microscopic quantum mechanical model of computers represented by Turing machines, *J. Stat. Phys.*, vol. 5, no. 5, pp. 562–591, 1980.

[5] M. Biafore, Can quantum computers have simple Hamiltonians?, In. *Proc. Workshop on Physics of Computation*, pp. 63–86, Los Alamitos, CA, IEEE Computer Society Press, 1994.

[6] A. M. Childs, D. W. Leung, and M. A. Nielsen, Unified derivations of measurement-based schemes for quantum computation, Phys. Rev. A 71, 032318 (2005)

[7] E. Farhi et al., A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem, Science 292, 472 (2001).

[8] R. Feynman, Quantum mechanical computers, *Opt. News*, vol. 11, pp. 11–46, 1985.

[9] N. Margolus, Parallel quantum computation, In W. Zurek, editor, *Complexity, Entropy and the Physics of Information*, Addison Wesley, 1990.

[10] A. Yu. Kitaev, Fault-tolerant quantum computation by anyons, Annals Phys. 303, pp.2-30 (2003),

[11] D. Leung, Quantum computation by measurements, Int. J. Quant. Inf. 2, 33 (2004)

[12] D. Janzing and P. Wocjan, Ergodic quantum computing, Quantum Inf. Process. 4, 129 (2005)

[13] D. Janzing, Spin 1/2 particles moving on a two dimensional lattice with nearest-neighbor interactions can realize an autonomous quantum computer, Phys. Rev. A 75, 012307 (2007)

[14] D. Janzing, P. Wocjan, and S. Zhang, A single-shot measurement of the energy of product states in a translation invariant spin chain can replace any quantum computation, quant-ph/0710.1615

[15] M. Nielsen, Quantum computation by measurement and quantum memory, Phys. Lett. A 308, 96 (2003)

[16] C. Perez-Delgado and D. Cheung, Models of Quantum Cellular Automata, quant-ph/0508164

[17] R. Raussendorf, Quantum cellular automaton for universal quantum computation, Phys. Rev. A 72, 022301 (2005)

[18] R. Raussendorf and H. Briegel, A One-Way Quantum Computer, PRL 86, 5188, (2000).

[19] D. J. Shepherd, T. Franz, R. F. Werner, Universally Programmable Quantum Cellular Automaton, PRL 97, 020502 (2006)

[20] Y. Shi, Both Toffoli and controlled-Not need little help to do universal quantum computation, quant-ph/0205115 (2002)

[21] K. G. H. Vollbrecht and J. I. Cirac, Quantum Simulators, continuous-time automata, and translationally invariant system, Phys. Rev. Lett. 100, 010501 (2008)

[22] Alastair Kay, The Computational Power of Translationally Invariant Hamiltonians, arXiv:0801.3228 (2008)

[23] Bradley A. Chase and Andrew J. Landahl, Universal quantum walks and adiabatic algorithms by 1D Hamiltonians, arXiv:0802.1207 (2008)