# Hamiltonian Quantum Cellular Automata in 1D

Daniel Nagaj* Pawel Wocjan†

February 6, 2008

**Abstract**

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).

## 1 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 sytems 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 [18, 15, 11, 6], adiabatic quantum computing [7, 2], 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

*Center for Theoretical Physics, MIT, Cambridge, MA 02139, USA, Email: nagaj@mit.edu

†School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816, USA, Email: wocjan@cs.ucf.edu

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 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 coud 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 (3) 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.

We present two different simiplified 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 number of computational qubits, 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. 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 2.1 we present the HQCA construction with cell size $d = 10$ and analyze the required run-time of this model in Section 2.2. Second, we give the HQCA construction with cell size $d = 20$ in Section 3.1 and discuss the readout procedure and the required run-time in Section 3.2. We summarize some useful results for the quantum walk on a line in Appendix A and prove a lemma concerning the diffusion of free fermions on a line in Appendix B.

# 2 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 $p_{10} \geq \frac{5}{6} - O\left(\frac{1}{\log L}\right)$ 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.

## 2.1 The Construction

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 $\frac{\pi}{4}$ 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 $K$ sequences of nearest neighbor gates $U_{k,g} \in \{W, S, I\}$, where gate $U_{k,g}$ belongs to the $k$-th sequence and acts on the pair of qubits $(g, g+1)$:

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

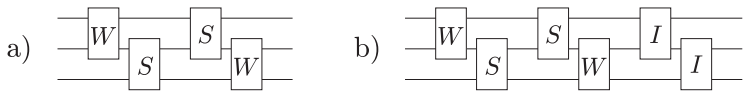


Figure 1: a) A quantum circuit consisting of two sequences of gates acting on nearest neighbors. b) The previous circuit with a third sequence of identity gates added.

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 $p \in \{\ \blacksquare, \blacktriangleright, W, S, I\}$ and a 2-dimensional data register $d \in \{0, 1\}$. We start by writing the initial product state

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

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

| $j$ | 1 | | | | | $\cdots$ | $M$ | $\cdots$ | | | | | $2M$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $p_j$ | $\blacksquare$ | $\blacksquare$ | $\blacksquare$ | $\blacktriangleright$ | $\blacksquare$ | $\blacksquare$ | $\blacktriangleright$ | $I$ | $W$ | $S$ | $I$ | $S$ | $W$ |
| $d_j$ | 0 | 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 gate sequences 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 $\blacksquare$ everywhere else. The right half holds the program in the form

$$I \underbrace{U_{1,1} \ldots U_{1,N-1}}_{\text{gate sequence 1}} I \underbrace{U_{2,1} \ldots U_{2,N-1}}_{\text{gate sequence 2}} I \ldots I \underbrace{U_{K,1} \ldots U_{K,N-1}}_{\text{last sequence}} , \tag{6}$$

with $U_{k,g} \in \{W, S, I\}$ and each sequence preceded by an identity gate. The bottom (data) register contains $N$ work qubits (labeled $w_n$ in the table) at positions $M + n$ for $n = 1 \ldots N$ and qubits in the state $|0\rangle$ everywhere else. We designate $w_N$ as the readout qubit.

We now give two simple rules, many applications of which generate the set of states $\{|\varphi_\sigma\rangle\}$ from the initial state $|\varphi\rangle$. We label the states by $\sigma$, the description of the sequence of rules we choose to apply to the initial state. We invite the reader to work out what happens to $|\varphi\rangle$ (4) as the rules are applied, noting that there are usually several possible rules that can be applied to a given state. The first rule says that the symbols $A \in \{W, S, I\}$ (from now on we call them gates and think of them as particles) in the program register can move one step to the left, if there is an empty spot there:

$$1: \quad \boxed{\blacksquare \mid A} \quad \rightarrow \quad \boxed{A \mid \blacksquare} \tag{7}$$

The second rule concerns what happens when a gate meets a pointer symbol:

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

In this case, the gate moves to the left, the pointer $\blacktriangleright$ gets pushed to the right, and the gate $A \in \{W, S, I\}$ is applied to the qubits in the data register below.

The initial state has $K$ pointers, one for each sequence of gates in the circuit. We constructed the initial state (4) in such a way that as a gate $U_{k,g}$ from the $k$-th gate sequence moves to the left, it meets the $k$-th (counting from the right) pointer $\blacktriangleright$ exactly above the work qubits to which the gate was intended to be applied $(g, g+1)$. However, one also needs to consider what happens when a gate meets a pointer for a different sequence. If this happens over a pair of extra qubits in the state $|0\rangle\,|0\rangle$, the qubits stay unchanged because the gate is either the controlled gate $W$, a swap gate or the identity. The second possibility is that a gate meets a pointer above the boundary of the work qubits (i.e. $|0\rangle_M\,|w_1\rangle_{M+1}$ or $|w_N\rangle_{M+N}\,|0\rangle_{M+N+1}$). We ensured that it is going to be the identity gate by inserting one in front of each gate sequence in (6). This implies that the leftmost (or the rightmost) work qubit again stays unchanged. Therefore, we are ensured that the state $|\varphi_\sigma\rangle$ in which the gate particles have all moved to the left half of the chain contains the result of the quantum computation $U$ in the state of the work qubits, while the additional qubits in the data register remain in the state $|0\rangle$.

Let us now allow all the rules to be applied backwards as well, opening the possibility of returning back to the initial state $|\varphi\rangle$ (undoing the computation). Although the number of possible sequences of rule (and backward rule) applications then becomes infinite, the space of states $\{|\varphi_\sigma\rangle\}$ is nevertheless finite-dimensional. In every state $|\varphi_\sigma\rangle$, the $M$ gates $\{W, S, I\}$ in the program register that started at positions $\{M+1, \ldots, 2M\}$ occupy some combination $C = \{a_1^{(C)}, \ldots, a_M^{(C)}\}$ of the $L$ sites of the chain. Given the rules (7)-(8), the order of the gates in the program register cannot change, i.e. $a_k^{(C)} < a_m^{(C)}$ for $k < m$. Because the positions of the pointers $\blacktriangleright$ and the state of the data register are also uniquely determined by $\{a_1^{(C)}, \ldots, a_M^{(C)}\}$, we can label the states we constructed by $|\varphi_C\rangle$ with $C = 1 \ldots \binom{L}{M}$.

We now give the universal translationally invariant Hamiltonian as a sum of translationally invariant terms

$$H_{10} = -\sum_{j=1}^{L-1} \left(R + R^\dagger\right)_{(j,j+1)}, \tag{9}$$

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

$$R = \sum_{A \in \{W,S,I\}} \Big[ |\, A \;\; \centerdot \,\rangle \langle \;\centerdot \; A\,|_{p_1,p_2} \otimes \mathbb{I}_{d_1,d_2} + |\, A \;\; \blacktriangleright\rangle \langle \blacktriangleright \; A\,|_{p_1,p_2} \otimes A_{d_1,d_2} \Big], \tag{10}$$

where $p$ stands for the program register and $d$ for the data register of the respective qudit.

Recall that our model of computation consists of initializing the qudit chain in the state $|\varphi\rangle$ and evolving the system for a time $\tau \leq \tau_{10}$ which we will determine in Section 2.2. Finally, we want to show that when we measure the output qubit $w_N$ in the data register, we will read out the result of the quantum computation $U$ with high probability.

## 2.2 Required Time Analysis

The time evolved state $|\varphi(\tau)\rangle$ (obtained from $|\varphi\rangle$ by evolving with $H_{10}$ for time $\tau$) is a superposition of the states $|\varphi_C\rangle$. We can write it as

$$|\varphi(\tau)\rangle = \sum_{C=1}^{\binom{L}{M}} \varphi_C(\tau)\,|\pi_C\rangle_{program} \otimes |\theta_C\rangle_{work} \otimes |\alpha\rangle_{extra}, \tag{11}$$

where $|\pi_C\rangle$ is the state of the program register of the chain in the state $|\varphi_C\rangle$, the corresponding state of the work qubits is $|\theta_C\rangle$ and $|\alpha\rangle$ is the state (all zero) of the extra data qubits. The state $|\theta_C\rangle$ of the work qubits holds the output state of the computation $U$, if all of the gate

"particles" have moved to the left of the work qubits. Let us now pad the qubit chain with $(f-1)M$ empty sites on the left and $M$ sites containing $I$ in the program register on the right:

$$|\varphi^{pad}\rangle \quad = \quad \begin{bmatrix} \cdot \\ 0 \end{bmatrix}^{\otimes(f-1)M} \otimes |\varphi\rangle \otimes \begin{bmatrix} I \\ 0 \end{bmatrix}^{\otimes M}. \tag{12}$$

The original chain had length $2M$, so the length of this padded chain is $L = (f+2)M$. The program register of the initial state $|\varphi^{pad}\rangle = |\varphi_{C=1}\rangle$ now has $2M$ gate "particles" $\{W, S, I\}$ at positions $a_m^{(1)} = fM + m$ with $m = 1 \dots 2M$. On this modified chain, every state $|\varphi_C\rangle$ in which the first $M$ gate "particles" are located in the first $fM$ sites of the chain ($a_m^{(C)} \leq fM$ for $m \leq M$) contains the finished computation in the state of its work qubits $|\theta_C\rangle$. Note that for all these states, the state of the work qubits is the same and equal to $|\theta_U\rangle = U|w_1 \dots w_N\rangle$, as it does not change under the extra identity gates we added. We can now rewrite the time evolved state (11) as

$$|\varphi(\tau)\rangle \quad = \quad \left( \sum_{a_M^{(C)} > fM} \varphi_C(\tau) |\varphi_C\rangle \right) + |\pi\rangle_{prog} \otimes |\theta_U\rangle_{work} \otimes |\alpha\rangle_{extra}, \tag{13}$$

where the sum in the first term is over the set of positions of the $2M$ gate particles in which the $M$-th particle is still near the right end of the chain and the computation is thus not finished yet. Meanwhile,

$$|\pi\rangle_{prog} = \sum_{a_M^{(C)} \leq fM} \varphi_C(\tau) |\pi_C\rangle_{prog}. \tag{14}$$

is a superposition of the program register states which correspond to an executed computation.

Let us recall the definition of the class BQP. Consider a language $L$ in BQP, a uniform family of circuits $U$ and a problem instance $x$. When $x \in L$, the probability of the circuit $U$ outputting *yes* is $p_{y \to y} \geq \frac{2}{3}$. On the other hand, when $x \notin L$, the probability of the circuit outputting *yes* is $p_{n \to y} \leq \frac{1}{3}$. Let us assume the worst case for our circuit $U$, i.e. $p_U = p_{y \to y} = 1 - p_{n \to y} = \frac{2}{3}$. 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 $\sigma_z$ on output qubit of the circuit $U$ is bounded from below by

$$\langle \sigma_{w_N}^{(z)} \rangle_{yes}^{circuit} \geq 1 \times p_U + (-1) \times (1 - p_U) = 2p_U - 1 = \frac{1}{3} \tag{15}$$

when $x \in L$, and analogously, for $x \neq L$, bounded from above by

$$\langle \sigma_{w_N}^{(z)} \rangle_{no}^{circuit} \leq -2p_U + 1 = -\frac{1}{3}. \tag{16}$$

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. Let us consider a *yes* instance ($x \in L$). Recalling (13) and observing that $|\pi\rangle_{prog}$ is orthogonal to the states of the program register in which the computation is not finished, we have

$$\langle \sigma_{w_N}^{(z)} \rangle_{yes} = \langle \varphi(\tau)| \sigma_{w_N}^{(z)} |\varphi(\tau)\rangle \quad = \quad p_{10} \underbrace{\langle \theta_U| \sigma_{w_N}^{(z)} |\theta_U\rangle_{work}}_{\text{output of } U} + (1 - p_{10}) \langle \varphi'| \sigma_{w_N}^{(z)} |\varphi'\rangle, \tag{17}$$

where $|\varphi'\rangle$ is the normalized first term in (13). The first term is the circuit output (15), therefore

$$\langle\theta_U|\,\sigma^{(z)}_{w_N}\,|\theta_U\rangle_{work} = \langle\sigma^{(z)}_{w_N}\rangle^{circuit}_{yes} \geq 2p_U - 1. \tag{18}$$

The second 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

$$\langle\varphi'|\,\sigma^{(z)}_{w_N}\,|\varphi'\rangle \geq -1. \tag{19}$$

Putting it together, we have

$$\langle\sigma^{(z)}_{w_N}\rangle_{yes} \quad \geq \quad p_{10}(2p_U - 1) - (1 - p_{10}) = 2p_{10}p_U - 1. \tag{20}$$

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

$$\langle\sigma^{(z)}_{w_N}\rangle_{no} \quad \geq \quad -2p_{10}p_U + 1. \tag{21}$$

We will now prove that when we choose the time $\tau$ 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 $L = (f+2)M$.

Let us analyze the time evolution of $|\varphi^{pad}\rangle$ (12) under $H_{10}$ (9). We can restrict the analysis to the program register of the chain, as the content of the data register in the time-evolved state $|\varphi(\tau)\rangle$ is completely determined by the content of the program register. The data register does not hinder the time evolution of the program register in any way. In fact, there exist bases, in which $H_{10}$ is identical (as a matrix) to $H_{10}$ restricted to the program register. Moreover, let us consider a further mapping of the system restricted to the program register to a line of qubits with length $L = (f+2)M$ as follows. Map the states $\{\blacktriangleright, \blacksquare\}$ to the state $|0\rangle$, and the states $\{W, S, I\}$ to the state $|1\rangle$. The mapping of $H_{10}$ to this system is a sum of hopping terms

$$H_q = -\sum_{j=1}^{L-1}(|10\rangle\langle 01| + |01\rangle\langle 10|)_{j,j+1}. \tag{22}$$

Let us use the Wigner-Jordan transformation to define the operators

$$b^\dagger_j \quad = \quad \sigma^z_1 \ldots \sigma^z_{j-1} \otimes |1\rangle\langle 0|_j \otimes \mathbb{I}_{j+1,\ldots,6M}, \tag{23}$$

$$b_j \quad = \quad \sigma^z_1 \ldots \sigma^z_{j-1} \otimes |0\rangle\langle 1|_j \otimes \mathbb{I}_{j+1,\ldots,6M}. \tag{24}$$

As $b^\dagger_j$ and $b_j$ have the required properties $\{b_i, b^\dagger_j\} = \delta_{ij}\mathbb{I}$ and $b^2_j = b^{\dagger 2}_j = 0$, they can be viewed as the creation and annihilation operators for a fermion at site $j$. Rewriting (22) in terms of (23)-(24), we obtain

$$H_f = -\sum_{j=1}^{L-1} b^\dagger_j b_{j+1} + h.c., \tag{25}$$

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 $|\Psi\rangle = b^\dagger_{4M+1} \ldots b^\dagger_{6M}\,|0\rangle$ 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 $\tau_{10}$ with the Hamiltonian given by (25) 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 (12) and let it evolve with $H_{10}$ (9) 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

$$p_{10} > \frac{5}{6} - O\left(\frac{1}{\log M}\right). \tag{26}$$

Therefore, equations (20) and (21) now read

$$\begin{aligned} \langle \sigma^{(z)}_{w_N} \rangle_{yes} &\geq \frac{1}{9} - O\left(\frac{1}{\log M}\right), \\ \langle \sigma^{(z)}_{w_N} \rangle_{no} &\leq -\frac{1}{9} + O\left(\frac{1}{\log M}\right). \end{aligned} \tag{27}$$

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 second term in (17)). 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.

# 3 The $d = 20$ HQCA

We now present our second construction, a HQCA for a chain of 20-dimensional qudits. As in Section 2.1, we describe an ecoding 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 $t = 1 \dots T = poly(N)$, 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 $\tau$ 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.

## 3.1 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 2.1, 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:

| | | |
|---|---|---|
| $W$, $S$, $I$ | : | the program sequence, |
| $\textcircled{W}, \textcircled{S}, \textcircled{I}$ | : | marked characters in the program sequence, used to propagate the active spot to the front (left) of the program sequence, |
| ▶ | : | apply gate symbol, |
| ▷ | : | shift program forward, |
| ↺ | : | a turn-around symbol, |
| . | : | empty spot (before/after the program). |

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)

| $j$ | 1 | $\cdots$ | | | | | | | | | | | $\cdots$ | $L$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $p_j$ | . | . | . | . | . | . | $I$ | $W$ | $S$ | $I$ | $I$ | $S$ | $W$ | ↻ |
| $d_j$ | 0 | 1 | 0 | 0 | 0 | 1 | $w_1$ | $w_2$ | $w_3$ | 1 | 0 | 0 | 0 | 1 |

(28)

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 . on the left, and then it contains the program in the form

$$I \underbrace{U_{1,1} \ldots U_{1,N-1}}_{1^{\text{st}} \text{ gate sequence}} I\, I \underbrace{U_{2,1} \ldots U_{2,N-1}}_{2^{\text{nd}} \text{ gate sequence}} I\, I \;\cdots\; I\, I \underbrace{U_{S,1} \ldots U_{S,N-1}}_{\text{last gate sequence}}, \tag{29}$$

with the program written from left to right. In our example (28), 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 ↻, 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

$$\begin{array}{lcl} 1: & \boxed{A}\boxed{\circlearrowleft} & \to \quad \boxed{\textcircled{A}}\boxed{.} \\ 2: & \boxed{A}\boxed{\textcircled{B}} & \to \quad \boxed{\textcircled{A}}\boxed{B} \\ 3: & \boxed{.}\boxed{\textcircled{A}} & \to \quad \boxed{\circlearrowleft}\boxed{A} \end{array} \tag{30}$$

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

$$4a: \quad \begin{array}{|c|c|}\hline . & \circlearrowleft \\ \hline & 1 \\ \hline \end{array} \quad \to \quad \begin{array}{|c|c|}\hline . & \blacktriangleright \\ \hline & 1 \\ \hline \end{array} \qquad 4b: \quad \begin{array}{|c|c|}\hline . & \circlearrowleft \\ \hline & 0 \\ \hline \end{array} \quad \to \quad \begin{array}{|c|c|}\hline . & \triangleright \\ \hline & 0 \\ \hline \end{array} \tag{31}$$

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

$$5a: \quad \begin{array}{|c|c|}\hline \blacktriangleright & A \\ \hline x & y \\ \hline \end{array} \quad \to \quad \begin{array}{|c|c|}\hline A & \blacktriangleright \\ \hline A(x,y) \\ \hline \end{array} \qquad 6a: \quad \begin{array}{|c|c|}\hline \blacktriangleright & . \\ \hline 1 \\ \hline \end{array} \quad \to \quad \begin{array}{|c|c|}\hline \circlearrowleft & . \\ \hline 1 \\ \hline \end{array} \tag{32}$$

When applying rule 5a, the apply gate symbol ▶ 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 ▶ 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 $|\psi_0\rangle$.

$$|\psi_{12}\rangle \;=\; \left[\begin{array}{cccccccccccccc} . & . & . & . & . & I & W & S & \blacktriangleright & I & I & S & W & . \\ 0 & 1 & 0 & 0 & 0 & 1 & |\ldots\theta\ldots\rangle & & & 1 & 0 & 0 & 0 & 1 \end{array}\right], \tag{33}$$

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

$$5b: \quad \boxed{\triangleright}\boxed{A} \quad \rightarrow \quad \boxed{A}\boxed{\triangleright} \qquad\qquad 6b: \quad \begin{array}{|c|c|}\hline \triangleright & \cdot \\ \hline 0 & \\ \hline \end{array} \quad \rightarrow \quad \begin{array}{|c|c|}\hline \circlearrowleft & \cdot \\ \hline 0 & \\ \hline \end{array} \tag{34}$$

where again $A$ stands for either $W$, $S$ or $I$. Rule 5b makes the program shift to the left while the ▷ symbol moves to the right. Finally, rule 6b deals with what happens when the ▷ 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 ▷ symbol will then be in the state 0, so that the ▷ symbol changes to the turn symbol ↻. The reason why we need to look at the qubit in the data register below the ↻ 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 ▷ 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 ↻ symbol will be above a 1 marker qubit), and subsequently, the ▶ 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.

$$|\psi_T\rangle \quad = \quad \left[\begin{array}{cccccccccccccc} \circlearrowleft & I & W & S & I & I & S & W & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 1 & 0 & 0 & 0 & 1 & |\dots\theta'\dots\rangle & & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right]. \tag{35}$$

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 (28), we have constructed the set of states $|\psi_t\rangle$ for $t = 0\dots T$ with $T = O(K^2N^2) = poly(N)$. 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 $T+1$

$$|\psi_t\rangle \quad \leftrightarrow \quad |t\rangle_{line}\,, \tag{36}$$

where $t = 0\dots T$.

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

$$H_{20} = -\sum_{i=1}^{L-1}\sum_{k=1}^{6b}\left(P_k + P_k^\dagger\right)_{(i,i+1)} \tag{37}$$

where the terms $P_k$ correspond to the rules 1-6b (30),(31),(32) and (34) and act on two neighboring qudits as

$$P_1 = \sum_{A\in\{W,S,I\}} |\textcircled{A}\ \blacksquare\rangle\langle A\ \circlearrowleft|_{p_1,p_2}\otimes\mathbb{I}_{d_1,d_2}, \tag{38}$$

$$P_2 = \sum_{A,B\in\{W,S,I\}} |\textcircled{A}\ B\rangle\langle A\ \textcircled{B}|_{p_1,p_2}\otimes\mathbb{I}_{d_1,d_2}, \tag{39}$$

$$P_3 = \sum_{A\in\{W,S,I\}} |\circlearrowleft\ A\rangle\langle\ \blacksquare\ \textcircled{A}|_{p_1,p_2}\otimes\mathbb{I}_{d_1,d_2}, \tag{40}$$

and

$$P_{4a} = |\ \blacksquare\ \blacktriangleright\rangle\langle\ \blacksquare\ \circlearrowleft|_{p_1,p_2}\otimes\mathbb{I}_{d_1}\otimes|1\rangle\langle 1|_{d_2}, \tag{41}$$

$$P_{4b} = |\ \blacksquare\ \triangleright\rangle\langle\ \blacksquare\ \circlearrowleft|_{p_1,p_2}\otimes\mathbb{I}_{d_1}\otimes|0\rangle\langle 0|_{d_2}, \tag{42}$$

$$P_{5a} = \sum_{A\in\{W,S,I\}} |A\ \blacktriangleright\rangle\langle\blacktriangleright\ A|_{p_1,p_2}\otimes A_{d_1,d_2}, \tag{43}$$

$$P_{5b} = \sum_{A\in\{W,S,I\}} |A\ \triangleright\rangle\langle\triangleright\ A|_{p_1,p_2}\otimes\mathbb{I}_{d_1,d_2}, \tag{44}$$

$$P_{6a} = |\circlearrowleft\ \blacksquare\rangle\langle\blacktriangleright\ \blacksquare|_{p_1,p_2}\otimes|1\rangle\langle 1|_{d_1}\otimes\mathbb{I}_{d_2}, \tag{45}$$

$$P_{6b} = |\circlearrowleft\ \blacksquare\rangle\langle\triangleright\ \blacksquare|_{p_1,p_2}\otimes|0\rangle\langle 0|_{d_1}\otimes\mathbb{I}_{d_2}. \tag{46}$$

When thinking of the set of states $|\psi_t\rangle$ as the set of positions of a particle on a line (36), $H_{20}$ becomes

$$H_{line} = -\sum_{t=0}^{T-1}\big(|t\rangle\langle t+1| + |t+1\rangle\langle t|\big). \tag{47}$$

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 $L$.

## 3.2 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 $\tau$ is to read out the output of the computation. As in Section 2.2, 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 $\blacksquare$ (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 $0 \le \tau \le \tau_{20}$ 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 (37) 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 $\tau$ chosen uniformly at random between 0 and $\tau_{20}$, 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 . 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 2.2, resulting in (27) again, with $M$ replaced by $T$.

## Acknowledgments

The authors would like to thank Dominik Janzing and Jeffrey Goldstone for helpful discussions. P.W. would like to thank Eddie Farhi's group for their hospitality and the W. M. Keck Foundation for partial support. D.N. gratefully acknowledges the support from the W. M. Keck Foundation Center for Extreme Quantum Information Theory and P.W. gratefully acknowledges the suppport by the NSF grants CCF-0726771 and CCF-0746600.

## A Quantum Walk on a Line

Here we analyze the quantum walk on a line and prove two useful lemmas used in Section 3.2 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

$$H_1 = -\sum_{j=1}^{L-1} \left(|j\rangle\langle j+1| + |j+1\rangle\langle j|\right). \tag{48}$$

The eigenvalues of this Hamiltonian are

$$\lambda_j = -2\cos\left(\frac{j\pi}{L+1}\right), \tag{49}$$

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

$$\phi_k^{(j)} = \sqrt{\frac{2}{L+1}} \sin\left(\frac{jk\pi}{L+1}\right). \tag{50}$$

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

(50):

$$p_\tau(m|c) = \left|\langle m|\, e^{-iH\tau}\, |c\rangle\right|^2 = \sum_{j,k=1}^{L} e^{-i(\lambda_j-\lambda_k)\tau}\phi_m^{(j)}\phi_c^{(j)}\phi_m^{(k)*}\phi_c^{(k)*}. \tag{51}$$

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 $0 \le \tau \le \tau_{20}$ as

$$\bar{p}_{\tau_{20}}(m|c) = \frac{1}{\tau_{20}}\int_0^{\tau_{20}} p_\tau(m|c). \tag{52}$$

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 (52). All the eigenvalues (49) are different, so we can express the limiting distribution as

$$\pi(m|c) = \lim_{\tau_{20}\to\infty}\bar{p}_{\tau_{20}}(m|c) = \sum_{j=1}^{L}\left|\phi_m^{(j)}\right|^2\left|\phi_c^{(j)}\right|^2, \tag{53}$$

which in our case is

$$\pi(m|c) = \frac{2+\delta_{m,c}+\delta_{m,L+1-c}}{2(L+1)}. \tag{54}$$

According to the following lemma, the average probability (52) 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 $\tau \le \tau_{20}$ chosen uniformly at random, starting in a position basis state $|c\rangle$. The average probability distribution $p_{\tau_{20}}(\cdot|c)$ converges to the limiting probability distribution $\pi(\cdot|c)$ as*

$$\sum_{m=1}^{L}|\bar{p}_{\tau_{20}}(m|c) - \pi(m|c)| \le O\left(\frac{L}{\tau_{20}}\right). \tag{55}$$

*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

$$\sum_{m}|\bar{p}_{\tau_{20}}(m|c) - \pi(m|c)| \le \frac{2}{\tau_{20}}\sum_{j\neq k}\frac{\left|\phi_c^{(j)}\right|^2}{|\lambda_j-\lambda_k|}. \tag{56}$$

Using (49) and (50), we can bound the expression on the right of (56). When $j$ is close to $k$, i.e. $|j-k| \le C_1$, we can obtain

$$\frac{\left|\phi_c^{(j)}\right|^2}{|\lambda_j-\lambda_k|} < 2. \tag{57}$$

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

$$\frac{\left|\phi_c^{(j)}\right|^2}{|\lambda_j-\lambda_k|} < \frac{C_2}{L+1}, \tag{58}$$

with $C_1$ and $C_2$ constants independent of $L$. Inserting into (56), we have

$$\sum_{m=1}^{L}|\bar{p}_{\tau_{20}}(m|c) - \pi(m|c)| \le \frac{8C_1L}{\tau_{20}} + \frac{C_2L}{\tau_{20}} = O\left(\frac{L}{\tau_{20}}\right), \tag{59}$$

which concludes the proof. □

Using Lemma 1, we will now prove an useful result utilized in the time analysis of the $d = 20$ HQCA in Section (3).

**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

$$p_{20} = \sum_{m=\frac{5L}{6}}^{L} \bar{p}_{\tau_{20}}(m|c). \tag{60}$$

Starting with (55), we have

$$\begin{aligned} O\left(\frac{L}{\tau_{20}}\right) &\geq \sum_{m=1}^{L} |\bar{p}_{\tau_{20}}(m|c) - \pi(m|c)| && (61)\\ &\geq \sum_{m=\frac{5L}{6}}^{L} |\bar{p}_{\tau_{20}}(m|c) - \pi(m|c)| && (62)\\ &\geq \left| \sum_{m=\frac{5L}{6}}^{L} \bar{p}_{\tau_{20}}(m|c) - \sum_{m=\frac{5L}{6}}^{L} \pi(m|c) \right| && (63)\\ &= \left| p_{20} - \frac{5}{6} + O\left(\frac{1}{L}\right) \right|. && (64) \end{aligned}$$

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

$$p_{20} \geq \frac{5}{6} - O\left(\frac{L}{\tau_{20}}\right). \tag{65}$$

□

# 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*

$$|\Psi_0\rangle = b^\dagger_{fM+1} b^\dagger_{fM+2} \dots b^\dagger_{fM+2M} |0\rangle. \tag{66}$$

*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 $\tau_{10}$ with the Hamiltonian*

$$H_f = -\sum_{j=1}^{L-1} b^\dagger_j b_{j+1} + h.c. \tag{67}$$

*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 $\tau_{10}$, 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

$$|\Psi_0\rangle = \Big[\, |\phi_{fM+1}\rangle \otimes |\phi_{fM+2}\rangle \otimes \cdots \otimes |\phi_{fM+2M}\rangle \,\Big]^{-}, \tag{68}$$

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

$$H_1 = -\sum_{j=1}^{L-1} \left( |j\rangle \langle j+1| + |j+1\rangle \langle j| \right), \tag{69}$$

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

$$|\Psi(\tau)\rangle = \Big[\, |\phi_{fM+1}(\tau)\rangle \otimes |\phi_{fM+2}(\tau)\rangle \otimes \cdots \otimes |\phi_{fM+2M}(\tau)\rangle \,\Big]^{-}. \tag{70}$$

The eigenfunctions of $H_1$ (quantum walk on a line) are plain waves (as in (49) and (50)), and the time evolved states $|\phi_j(\tau)\rangle$ thus readily available. Let us define the unitary matrix $u(\tau)$ by

$$|j(\tau)\rangle = \sum_{k=1}^{L} u_{jk}(\tau) \, |k\rangle \,. \tag{71}$$

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

$$\begin{aligned} b_j^\dagger(\tau) &= \sum_{k=1}^{L} u_{jk}(\tau) b_k^\dagger, \\ b_j(\tau) &= \sum_{k=1}^{L} u_{jk}^*(\tau) b_k. \end{aligned} \tag{72}$$

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

$$X = \sum_{m=1}^{fM} \hat{n}_m. \tag{73}$$

Its expectation value at time $\tau$ is

$$E_\tau(X) = \sum_{m=1}^{fM} \langle \Psi(\tau)| \, \hat{n}_m \, |\Psi(\tau)\rangle \,. \tag{74}$$

The number operator for site $m$ is $\hat{n}_m = b_m^\dagger b_m$. We can go to the Heisenberg picture and use (72) to write

$$\begin{aligned}
\langle \Psi(\tau)| \hat{n}_m |\Psi(\tau)\rangle &= \langle \Psi_0| b_m^\dagger(\tau) b_m(\tau) |\Psi_0\rangle && (75)\\
&= \sum_{c=1}^{L}\sum_{d=1}^{L} u_{mc}(\tau) u_{md}^*(\tau) \langle \Psi_0| b_c^\dagger b_d |\Psi_0\rangle && (76)\\
&= \sum_{c=1}^{L} |u_{mc}(\tau)|^2 \langle \Psi_0| b_c^\dagger b_c |\Psi_0\rangle && (77)\\
&= \sum_{c=fM+1}^{L} \underbrace{|u_{mc}(\tau)|^2}_{p_\tau(m|c)}, && (78)
\end{aligned}$$

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 $\tau$ when it started from the site $c$ and performed a quantum walk on a line, according to (69). Inserting this into (74), the expected number of particles not in the rightmost part of the chain at time $\tau$ is

$$E_\tau(X) = \sum_{c=fM+1}^{L} \left( \sum_{m=1}^{fM} p_\tau(m|c) \right). \tag{79}$$

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

$$\bar{E}_{\tau_{10}}(X) = \frac{1}{\tau_{10}} \int_0^{\tau_{10}} E_\tau(X)\, d\tau. \tag{80}$$

For a quantum walk on a line, the time-averaged probability (52) of finding a particle that started at position $c$ at final position $m$ converges to the limiting distribution (54) according to Lemma 1 (55) proven 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

$$\bar{E}(X) = \sum_{m \le fM} \sum_{c > fM} \pi(m|c) \tag{81}$$

as

$$\left| \bar{E}_{\tau_{10}}(X) - \bar{E}(X) \right| \le O\left( \frac{LM}{\tau_{10}} \right). \tag{82}$$

Recalling the limiting probability distribution for a quantum walk on a line of length $L$ (54), we have

$$\begin{aligned}
\bar{E}(X) &= \sum_{m \le fM} \sum_{c > fM} \pi(m|c) && (83)\\
&= fM \times 2M \times \frac{2}{2(L+1)} + 2M \times \frac{1}{2(L+1)} && (84)\\
&= 2M \left( \frac{f}{f+2} \right) + O(1). && (85)
\end{aligned}$$

Putting this into (82), the average value of $X$ when the time $\tau \le \tau_{10}$ is chosen uniformly at random is bounded from below as

$$\bar{E}_{\tau_{10}}(X) \ge 2M \left( \frac{f}{f+2} \right) - O\left( \frac{LM}{\tau_{10}} \right). \tag{86}$$

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 $\tau$ chosen randomly is close to $2M$. Therefore, the fraction $\Delta$ of times at which we measure a number significantly lower than $2M$ must be small. Let us bound $\Delta$ in the worst case scenario. This is when each unsuccessful measurement yields $X = M$, and each successful measurement gives us $2M$. We then have

$$\Delta M + (1-\Delta)2M \geq \bar{E}_{\tau_{10}}(X), \tag{87}$$

$$\Delta \leq \frac{2\bar{E}_{\tau_{10}}(X) - M}{M}. \tag{88}$$

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

$$p_{10} = 1 - \Delta \geq \frac{2M\left(\frac{f}{f+2}\right) - O\left(\frac{LM}{\tau_{10}}\right) - M}{M} = \frac{f-2}{f+2} - O\left(\frac{L}{\tau_{10}}\right). \tag{89}$$

□

# References


[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 Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, p. 42 (2004).

[3] D. Aharonov, J. Kempe, D. Gottesman, Power of Quantum Systems on a Line, quant-ph/0705.4077, and S. Irani, The Complexity of Quantum Systems on a One-dimensional Chain, quant-ph/0705.4067

[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., Science 292, 472 (2001).

[8] R. Feyman, 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), Preskill and Ogburn, Topological quantum computation (1997), Preskill, Fault-tolerant quantum computation (1997), Mochon, Anyons from non-solvable groups are sufficient for universal quantum computation (2003), Mochon, Anyon computers with smaller groups (2004), Freedman, Larsen, and Wang, A modular functor which is universal for quantum computation (2000), Freedman, Kitaev, and Wang, Simulation of topological field theories by quantum computers (2000)

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

[12] D. Janzing and P. Wocjan, 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, 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/020115 (2002)

[21] K. G. H. Vollbrecht and J. I. Cirac, Quantum Simulators, continuous-time automata, and translationally invariant system, quant-ph/0704.3432