Universal Hamiltonians for Exponentially Long Simulation

Thomas C. Bohdanowicz and Fernando G.S.L. Brandão

Institute of Quantum Information and Matter, California Institute of Technology, Pasadena, CA

October 25, 2017

Abstract

We construct a Hamiltonian whose dynamics simulate the dynamics of every other Hamiltonian up to exponentially long times in the system size. The Hamiltonian is time independent, local, one dimensional, and translation invariant. As a consequence, we show (under plausible computational complexity assumptions) that the circuit complexity of the unitary dynamics under this Hamiltonian grows steadily with time up to an exponential value in system size. This result makes progress on a recent conjecture by Susskind, in the context of the AdS/CFT correspondence, that the time evolution of the thermofield double state of two conformal field theories with a holographic dual has circuit complexity increasing linearly in time, up to exponential time.

1 Introduction

The notion of *universality* is central in science. At its core is the idea that a simple set of objects can describe the fundamental properties of a much richer class of objects. One particular successful example in physics is the use of universality in classifying phase transitions. It turns out that critical phenomena can be understood by only looking at a small set of systems and critical exponents associated to them, which are universal in the sense that they can reproduce the critical behavior of every other system [1]. A second example, which is the focus of this paper, is the study of universal quantum dynamics. The goal here is to understand and classify, in several different scenarios, which quantum dynamics can simulate any other [2]. This line of investigation is at the core of the idea of a (universal) quantum computer, which should be able to efficiently simulate the dynamics of any quantum system [3, 4]. It also naturally extends to the quantum domain the study of universality in classical dynamics, usually considered in the context of cellular automata.

There are several different notions of universality of quantum dynamics. One of the first to be considered was the notion of universal quantum gates, in which one is interested in sets of quantum gates that can approximate any unitary operator. This is a rich problem and there is a beautiful theory around it [5, 6], still with many unresolved questions. A noteworthy result is that almost any two qubit gate is universal in the sense that it (together with the ability to swap the qubits) can approximate any other unitary. Therefore, in a well defined sense, universality is the general rule for quantum gates.

Another notion concerns universal Hamiltonians. Here we are interested in identifying Hamiltonians whose dynamics can simulate any other (more precisely we can ask whether we can perform universal quantum computation with the Hamiltonian). One variant of the question is to

1

consider sets of few-qubit interactions and classify which are universal, given the ability of switching the interactions in a time-dependent fashion. This question is closely related to the one for quantum gates and, similarly, a generic two-qubit Hamiltonian is typically universal for quantum computing [7].

Another interesting variant is to consider a fixed time-independent Hamiltonian and ask if it is universal. It is clear that the class of unitaries which can be approximated by the time evolution of a fixed Hamiltonian is rather limited (as the eigenbasis of e^{itH} for a fixed H is always the same). However the situation is more interesting if one allows for some form of encoding of the dynamics by preparing the initial state of the system in a suitable way. Indeed, Vollbrecht and Cirac [8] gave a construction of a time-independent and translation-invariant model which is universal for quantum computing. The desired quantum circuit to be implemented is encoded in the initial state as a particular computational basis state. Later on, Nagaj and Wocjan made refinements to this construction in [9]. In contrast to the other notions of universality, it is unclear if a typical local Hamiltonian will be universal in the sense above. Indeed apart from a few examples, it is still largely unexplored which Hamiltonians are universal for time-independent evolutions.

Circuit Complexity: An a priori unrelated question in quantum complexity theory concerns the circuit complexity of quantum dynamics. The circuit complexity of a unitary operator consists of the minimum number of two-qubit gates needed to (approximately) construct the unitary. Large complexity is a general feature of unitary operators. Indeed a simple counting argument shows that most unitaries on n qubits will have circuit complexity $2^{\Omega(n)}$ [10]. Moreover in Ref. [11] it was shown that most quantum circuits with m gates have circuit complexity at least $\Omega(m^{1/11})$. It is much harder to prove circuit complexity lower bounds for the evolution of a fixed Hamiltonian (or even for the application of a fixed unitary many times). Indeed, proving a superpolynomial lower bound on circuit complexity for exponential long times would result in a major breakthrough in classical complexity theory (as shown in [10], it would imply that PSPACE is not contained in BQP/poly, which appears to be far beyond the reach of current proof techniques).

It turns out that the concept of universality naturally connects with circuit complexity. If one aims to find a unitary whose dynamics has large circuit complexity, it would be natural to consider one whose time evolution is universal, capable of simulating any other evolution (with a reasonable overhead). Indeed, from the result of [11], it directly follows that we can lower bound the circuit complexity of the unitary evolution of the Vollbrecht-Cirac universal Hamiltonian [8] on n qubits, for large enough time, by n^c, for a constant $c > 0$. However it is not clear how to obtain larger lower bounds (i.e. superpolynomial in n) for long times, e.g. exponential in n, even under computational complexity assumptions. This is a natural problem to consider on its own (e.g. it was considered in Ref. [12] under the name of "fast-forwarding" and connected to the time-energy uncertainty relation), but it has recently gained a renewed interest due to a possible unexpected application in the context of quantum gravity and holography, which we now briefly review.

Susskind's Conjecture: The AdS/CFT correspondence [13] posits the equivalence between a theory of quantum gravity in Anti-de-Sitter (AdS) space (i.e. gravity with a negative cosmological constant) with $d+1$ dimensions (called the bulk theory) and a quantum conformal field theory (CFT) with d dimensions (called the boundary theory). This idea suggests that gravity is an emergent phenomenon (as it does not appear explicitly in the CFT description) and has become one of the most influential ideas in physics in the last twenty years. The correspondence is usually phrased as a dictionary between properties of the two theories.

Yet there is one property of the quantum gravity picture for which it is hard to find a suitable partner in the CFT picture. This is the volume of a non-traversible wormhole (or Einstein-Rosen bridge) in AdS space connecting two boundary CFTs. One can argue that the volume of the wormhole will grow linearly with time up to exponential time [14]. But most of the traditionally considered physical properties of the CFT (which usually involve only few-body operators) will equilibrate much sooner (on the timescale of the scrambling time). One property of the CFT system which does seem to have the same behaviour as the volume of the wormhole is the circuit complexity of the joint quantum state[1] of the two CFTs associated with the boundary of the wormhole (given by the time evolution of the so-called 'thermofield double state' (TDS), which at infinite temperature reduces to a maximally entangled state of the two CFTs). Indeed, in analogy with the behavior of the wormhole volume, we expect the circuit complexity of the state to grow linearly with time up to exponential in the size of the system, at which point it has to saturate. Then at recursion times (which is doubly exponential in system size) the circuit complexity will have sharp oscillation before stabilizing again at an exponential value. Such a connection prompted Susskind to conjecture that the circuit complexity of the CFT state is the dual property of the volume of the wormhole.

Although fascinating, Susskind's proposal is still somewhat speculative; it is an interesting and challenging task to make it more concrete. Progress was made by Aaronson and Susskind [15]. They considered a toy model of the problem for which concrete results about the circuit complexity could be established. Let V describe one step of the time evolution of the CFT. The CFT thermofield double state (at infinite temperature and regularizing the theory to have finite dimension N) after t time steps is given by

$$|\text{TDS}(t)\rangle := \frac{1}{\sqrt{N}} \sum_i V^t |i\rangle_{\text{CFT, 1}} \otimes (V^T)^t |i\rangle_{\text{CFT, 2}} = \frac{1}{\sqrt{N}} \sum_i |i\rangle_{\text{CFT, 1}} \otimes U^t |i\rangle_{\text{CFT, 2}} \qquad (1)$$

with $U = (V^T)^2$, where V^T is the transpose of V. For Susskind's correspondence to hold, we need the circuit complexity of $|\text{TDS}(t)\rangle$ to grow linearly in time up to an exponential value in the system size. Aaronson and Susskind abstracted away the fact that U is associated to the dynamics of a CFT and asked if there is any (efficiently implementable) unitary U for which one can show that the corresponding $|\text{TDS}(t)\rangle$ has complexity growing up to exponential with the number of applications t. They proved that choosing U as a step function of a reversible and computationally-universal cellular automaton achieves the goal (under certain computational complexity assumptions). A natural open question is whether one can prove something similar for the evolution of the CFT. Short of that, can we get closer to this goal? For example, can we find a local Hamiltonian whose evolution can replace the step function of the cellular automaton in the Aaronson-Susskind reasoning?

1.1   Main Results

In this paper we consider universality for fixed Hamiltonians up to exponential times. All Hamiltonians will be normalized to have $\|H\| = 1$. Our desired notion of universality is captured by the following definition [2]:

---

[1]Analogously to the circuit complexity of a unitary, the circuit complexity of a state is the minimum number of two-qubit gates (from a given universal set of gates) needed to (approximately) create the state.

[2]This definition was suggested to us by Dorit Aharonov [16].

Definition 1. *A family of Hamiltonians* $\{H_m\}_{m\in\mathbb{N}}$, *indexed by the number of qudits m they act on, is called a* Universal Hamiltonian Family *if for any k-local Hamiltonian* $H \in \mathcal{B}(\mathbb{C}^{2^n})$ *on n qubits (with* $k \leq O(\log(n))$) *and time t, there are* $\mathrm{poly}(n)$-*sized quantum circuits D and E,* $m = \mathrm{poly}(n, \log t)$, *and* $t' = \mathrm{poly}(t, n)$ *such that*

$$\|e^{iHt} - (I^{\otimes n} \otimes \langle 0^{m-n}|)De^{iH_m t'}E(I^{\otimes n} \otimes |0^{m-n}\rangle)\| < 1/\mathrm{poly(n)}. \tag{2}$$

Previous results [8, 9] achieved a weaker notion of universality. In the notation of Definition 1, the size m of the Hamiltonians in Refs. [8, 9] scales as $m = \mathrm{poly}(n, t)$, which prevents efficient simulation for exponentially long times because it would require preparing an exponentially large state encoding the dynamics. So, in words, our construction allows exponentially long simulation time using only polynomial space and without requiring any active control or intervention during the simulation. Note also that this notion of universality is imcomparable to the notion considered in [2]. Rather than reproducing a large class of properties of the system being simulated (e.g. spectrum of Hamiltonian, etc.) our simulation scheme only simulates the *dynamics* and is weaker in that sense. However, our simulation faithfully simulates dynamics for times up to exponentially large in the system size, whereas the construction of Ref. [2] can not, and is in that sense stronger.

Although Definition 1 is natural when discussing universal Hamiltonians, we can also consider the following stronger definition, which will be useful to us:

Definition 2. *A family of Hamiltonians* $\{H_m\}_{m\in\mathbb{N}}$, *indexed by the number of qudits m they act on, is called a* Circuit Universal Hamiltonian Family *if for every* $\mathrm{poly}(n)$-*sized circuit U on n qubits and time t, there are* $\mathrm{poly}(n)$-*sized quantum circuits D and E,* $m = \mathrm{poly}(n, \log t)$, *and* $t' = \mathrm{poly}(t, n)$ *such that*

$$\|U^t - (I^{\otimes n} \otimes \langle 0^{m-n}|)De^{iH_m t'}E(I^{\otimes n} \otimes |0^{m-n}\rangle)\| < 1/\mathrm{poly(n)}. \tag{3}$$

From the Hamiltonian simulation results of [17], it follows that a *Circuit Universal Hamiltonian Family* is also a *Universal Hamiltonian Family*. Our main result is the following:

Theorem 3. *There is a* Circuit Universal Hamiltonian Family *in one spatial dimension with translation-invariance and local spin dimension of 14580. The encoding circuit consists of preparing computational basis product states. The decoding circuit consists of making a measurement in the computational basis and resetting computational basis product states to the all zero state.*

It is an open question to decrease the local dimension of the model; we expect substantial improvements to be possible. In words, we explicitly construct a local term $h_{l,l+1}$ acting on two qudits, each of dimension 14580, such that the family of Hamiltonians

$$H_m = \frac{1}{m} \sum_{l=1}^{m-1} h_{l,l+1} \tag{4}$$

is universal in the sense of Definition 2.

Theorem 3 can be used to argue that the circuit complexity of $e^{iH_m t}$ must grow to a superpolynomial value. The starting point is the result of Atia and Aharonov that unless $\mathrm{PSPACE} = \mathrm{BQP}$, there must exist a 2-sparse row computable Hamiltonian whose long-time dynamics has superpolynomial circuit complexity (see Theorem 6 of [12]). This result together with Theorem 3 gives:

Corollary 4. *Unless* $\mathrm{PSPACE} = \mathrm{BQP}$, *the circuit complexity of* e^{itH_m} *is superpolynomial for* $t = 2^{O(m)}$.

4

An interesting feature of Corollary 4 is that we can identify a *single* Hamiltonian – more precisely a fixed interaction term forming a translation-invariant model for each length – which can be shown to have growing circuit complexity (under certain computational complexity assumptions).

We can also make progress on Susskind proposal discussed before. Define

$$|\text{TDS}_{H_m}(t)\rangle := \frac{1}{\sqrt{N}} \sum_i |i\rangle \otimes e^{itH_m} |i\rangle. \tag{5}$$

Then we have:

Corollary 5. *Unless* PSPACE *is inside* PP/poly, *the circuit complexity of* $|TDS_{H_m}(t)\rangle$ *is superpolynomial for* $t = 2^{O(m)}$. *Assuming that there are no subexponential* PP *circuits for* PSPACE, *the circuit complexity of* e^{itH_m} *for* $t = 2^{O(m)}$ *is* $2^{\Omega(m)}$.

The proof follows directly from Theorem 7.1.2 of [10], by using H_m to simulate the step function of the universal cellular automaton used there. The corollary makes partial progress on Susskind's proposal, by finding a quantum evolution which shares more features with the dynamics of a CFT (locality and translation-invariance) than the cellular automaton of [15]. We leave as open questions demonstrations that our construction can replicate important features and symmetries expected of a CFT Hamiltonian, and whether or not it can be used to create a state whose circuit complexity actually grows *linearly* in t from $t = 0$ up to exponential times (under plausible complexity theoretic assumptions).

1.2   Sketch of Construction

The construction of our circuit universal Hamiltonian family (CUHF) that proves Theorem 3 makes use of the concept of a Hamiltonian Quantum Cellular Automaton (HQCA), as described by Nagaj and Wocjan in [9]. The formal definition of an HQCA is as follows:

Definition 6. *A Hamiltonian Quantum Cellular Automaton (HQCA) is a local, time-independent, translation-invariant Hamiltonian H on a lattice of qudits which carries out quantum computation via the following sequence of steps:*

1. *The input of the computation as well as information describing the computation to be performed (which is described by some unitary operator U) is encoded in the state of the qudits.*

2. *The qudits undergo continuous time evolution under H for some time t.*

3. *A simple basis state measurement on a subset of the qudits collapses, with high probability, the state of the whole system to one where an appropriate subset of the qudits contains the desired output of the quantum computation.*

Such a Hamiltonian H is termed a cellular automaton because the local terms which sum up to make H can usually be thought of as transitions between different basis states for a local part of the lattice, which themselves correspond to reversible transition rules of some classical cellular automaton. This will be evident in the description of our construction.

Our CUHF $\{H_m\}_{m \in \mathbb{N}}$ will actually be a family of universal HQCA on a 1D chain of m qudits, which satisfy the following properties in addition to those in the definition above:

5

- The local qudit dimension is constant.

- A quantum circuit U with C gates acting on N qubits can be applied t times through time evolution via Hamiltonian H_m on a chain of $m = \text{poly}(N, C, \log t)$ qudits.

The starting point of our construction is that of Nagaj and Wocjan [9], which produces a $1D$ CUHF with constant local qudit dimension $d = 20$, for which the size of the chain scales as $m = \text{poly}(N, C)$. A version of this construction is described in detail in section 2, where we also identify and highlight some important features of their construction that are essential to preserve in our eventual final construction [3].

We build an HQCA which only encodes U into the input state and a number t encoded in binary, but which is able to apply U repeatedly up to t times after waiting for a time $t' = \text{poly}(N, C, t)$ [18]. Therefore the size of the system needed for the simulation is only $m = \text{poly}(N, \log t)$. In the following sections, we proceed to augment Nagaj-Wocjan construction [9] step by step to add in more functionality until we achieve the desired $m = \text{poly}(N, \log t)$ scaling for the simulation size.

The innovations which allow us to obtain our final construction dictate the organization of the remainder of the paper:

1. The Nagaj-Wocjan construction can only apply its encoded unitary U once. So, first, we modify the Nagaj-Wocjan construction so that when the HQCA reaches its end state where U has been applied, it can undergo transitions to 'reset' itself and allow for another application of U without undoing the original application of U. This basically has the effect of allowing the Nagaj-Wocjan construction to run over and over again rather than just once. This is described in Section 3.

2. Next, we augment the construction to include a binary clock in the qudit chain which counts the number of times that U has been applied. This allows states of the chain with different numbers of U applications to be orthogonal so that they can be distinguished by a simple measurement. Binary clock constructions of a similar nature also appear in [19, 20, 21], in the context of proving hardness/uncomputability results for the ground-energy/gap of translation-invariant models. This is described in Section 4.

3. Finally, we augment the construction with the addition of a target register that allows one to specify a certain desired number of applications of U which is stored in the start state. Once the clock has counted this number of applications of U, the HQCA continues evolving forward in time similarly to the way in which it did before, but it no longer makes any applications of U to the qubits. This allows one to have complete control over how many times U is applied to the qubits, and is crucial to being able to guarantee that after waiting a reasonable amount of time, performing a simple measurement on a subset of the chain is likely to yield a state with the desired number of applications of U. This is described in Section 5.

The main challenge in achieving each of these results was to carefully engineer transition rules such that starting from the appropriate start state, for every state that can be obtained from the

---

[3]E.g., the definition and consequences of having a uniquely orthogonally generated set of states for an HQCA, see section 2.8

start state through application of transition rules, exactly one forward transition rule and one reverse transition rule is applicable to that state. This guarantees that the sequence of states obtained from the start state through application of the HQCA transition rules can map directly onto position eigenstates of a single particle on a chain, so that the analysis of the time evolution under the HQCA reduces to that of a single particle continuous-time quantum walk on a line (see Sections 2.8 and 5.5).

2 Construction I: Applying U Once

We begin with the modest goal of applying U to our qubits once ($x = 1$) using the time evolution of a simple time-independent and translation-invariant Hamiltonian, where the sequence of gates that implements U is stored in the start state of the qudit chain that we embed our qubits into for the simulation. The construction described in this section only differs from the $d = 20$ HQCA construction of Nagaj and Wocjan [9] in notation (and only slightly at that). We do, however, formalize and elaborate on some of the essential ideas underlying the success and utility of their construction which will be important in subsequent sections when we expand their construction, the most important of which being the definition of what it means for a sequences of quantum states to be 'uniquely orthogonally generated' by a sequence of transition rules of an HQCA (see Section 2.7). The construction described in this section will be referred to as 'construction I' throughout the rest of the paper.

2.1 Gateset

The universal gate set we will use to describe a quantum circuit for U is $\mathcal{G} = \{W, S, I\}$ where each gate acts on a local pair of left and right qubits and

- W is a controlled rotation by $\pi/2$ about the y-axis of the Bloch sphere (where the left qubit always controls the right)

- S is the two-qubit swap gate

- I is shorthand for the two-qubit identity gate $I \otimes I$.

We note that because all of these gates leave the state $|00\rangle$ invariant, to use these gates to simulate the application of an arbitrary U on a state of the form $|0^n\rangle$, some number of qubits in the $|1\rangle$ state must be provided in order to simulate gates that act non-trivially on the $|0\rangle$ state. This overhead is polynomial in the number of gates that U would have in a circuit using a more traditional gateset, so we ignore it and assume that the state that our gates act on has the correct number of qubits in $|1\rangle$ states for the simulation to proceed.

For the convenience of the construction, the way that U breaks down into a sequence of gates from $\mathcal{G}$ will be as follows: we write the unitary as a gate sequence from $\mathcal{G}$ in the form

$$U = (U_{K,1} \ldots U_{K,N-1})II(U_{K-1,1} \ldots U_{K-1,N-1})II \cdots II(U_{1,1} \ldots U_{1,N-1})I. \tag{6}$$

Here we say that U is expressed as K rounds of $N - 1$ gates from $\mathcal{G}$. Note that the use of the term 'round' of gates differs here from the usual usage. Normally, a round of gates from some gate set is some tensor product of $m \leq N$ single or two-qubit gates with the property that each of the N

7

qubits has exactly one gate acting on it (identity counted as a gate). This allows the number of rounds making up U to count the time complexity of applying the circuit. This is not true of our rounds. The gate $U_{k,m} \in \mathcal{G}$ is a two qubit gate acting on qubits m and $m+1$. So, two successive gates in a round, $U_{k,m}$ and $U_{k,m+1}$ will possibly each act non-trivially on qubit $m+1$. Thus, the number of rounds, K, in our circuit description of U will in general under-count the depth or time complexity of applying U as written, but it will undercount at most by a factor of N. The reason we write our circuit this way is for the convenience of constructing our desired HQCA. For ease of discussion, we will use the term *depth* of U to refer to the numbe of rounds, K, in the above sense rather than the traditional sense.

Notice also that in the above description, each round of gates is padded with an additional I on either side, except for the final round which does not have an I on the left side. The reason for this is that it makes the construction of our desired HQCA more convenient, and will be explained below. With the above convention for describing U in terms of $\mathcal{G}$, a unitary U of depth K acting on N qubits has a circuit sequence of length $K(N+1) - 1$.

2.2 Hilbert Space

Here we describe the Hilbert space $\mathcal{C}$ of the 1D chain that our HQCA Hamiltonian H will act on. We imagine our chain as consisting of two registers $\mathcal{C} = P \otimes D$, each with L local sites $\mathcal{C}_j = P_j \otimes D_j$, indexed from left to right. We will describe orthonormal basis states for each local site Hilbert space in terms of symbols:

- The program register P, whose local sites are P_j, stores the sequence of gates that describe U and controls its application to the work qubits. The local hilbert space is 10 dimensional, with basis symbols $\{W, S, I, \overrightarrow{W}, \overrightarrow{S}, \overrightarrow{I}, \blacktriangleleft, \triangleleft, \blacksquare, \rightarrow\}$. So, $P = \mathbb{C}^{10}$.

- The data register D, whose local sites D_j are qubits described by computational basis symbols $\{0, 1\}$, is simply a chain of qubits, N of which are the 'work qubits' to which we want to apply the unitary U. So, $D = \mathbb{C}^2$.

The symbols in the basis states for P_j have the following interpretation

$$
\begin{array}{rl}
W, S, I & : \quad \text{representing unitary gates in the program sequence that will apply U,} \\
\overrightarrow{W}, \overrightarrow{S}, \overrightarrow{I} & : \quad \text{marked characters in the program sequence, used to propagate} \\
& \quad \text{the active spot to the front (left) of the program sequence,} \\
\blacktriangleleft & : \quad \text{apply gate symbol,} \\
\triangleleft & : \quad \text{shift program forward,} \\
\rightarrow & : \quad \text{a control symbol indicating that we're in the process of shifting the program} \\
& \quad \text{sequence to the right,} \\
\blacksquare & : \quad \text{empty spot (before/after the program).}
\end{array}
$$

The way these symbols will interact will become clearer when illustrating the transition rules defined below. The length of the chain will be $L = (2K - 1)(N + 1) + 2$ where K is the number of rounds/depth of the circuit description of U, and N is the number of qubits that U is being applied to. The N qubits which U is being applied to will be referred to as the *work qubits*, and their location in the chain $\mathcal{C}$ will be the sites D_j of the data register with $j = (K - 1)(N + 1) + 1 + n$ for $n = 1...N$.

2.3 Initial State

The rest of the construction will be described with a simple illustrative example in mind, where we are applying a $K = 2$-round unitary $U = (S_{12}W_{23})II(W_{12}S_{23})I$ to $N = 3$ work qubits. Details regarding certain choices about the structure of the initial state will make sense after describing the transition rules and running through applying them to this example initial state. This chain will have $L = 14$, and the initial state is written as a product state over the sites and registers: $|\psi_0\rangle = \otimes_{j=1}^{14}(|p_j\rangle \otimes |d_j\rangle)$ where $|p_j\rangle$ is the state of the jth site of the program register and $|d_j\rangle$ is the state of the jth site of the data register. The following table describes the initial state:

j	1	$\cdots$											$\cdots$	L
p_j	$\rightarrow$	S	W	I	I	W	S	I	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
d_j	1	0	0	0	1	w_1	w_2	w_3	1	0	0	0	1	0

(7)

Note the following about the structure of the initial state:

- The individual initial states of the work qubits have here been labelled w_1, w_2 and w_3 to remind the reader where they are located.

- The work qubits are bordered by data qubits in the 1 state on either side

- The work qubits are further padded to the left by the sequence 10^N $K - 1$ times, and to the right by the sequence 0^N1 $K - 1$ times

- The final data qubit is in the state 0

The reason for this structure will become clearer in the description and application of the transition rules to this example, and will be commented upon further below.

2.4 Transition Rules

We now describe the forward transition rules of our HQCA. Application of these forward transition rules to our initial state $|\psi_0\rangle$ will create a sequence of $T_I + 1$ mutually orthogonal states $|\psi_t\rangle_{t=0}^{T_I}$. Note that T_I is a function of N and K. The transition rules are deliberately constructed such that for each $|\psi_t\rangle$, there is always one single unique forward transition rule that can be applied to it, which results in the state $|\psi_{t+1}\rangle$. For every forward transition rule there is a unique reverse transition rule (the Hermitian conjugate of the forward transition rule), and for each $|\psi_t\rangle$, there is always one single unique reverse transition rule that can be applied to it, which results in $|\psi_{t-1}\rangle$. No reverse transition rule will be applicable to $|\psi_0\rangle$, and no forward transition rule will be applicable to $|\psi_{T_I}\rangle$.

The transition rules are summarised as follows:

$$1: \quad \boxed{\rightarrow \mid A} \quad \rightarrow \quad \boxed{\cdot \mid \overrightarrow{A}} \tag{8}$$

$$2: \quad \boxed{\overrightarrow{A} \mid B} \quad \rightarrow \quad \boxed{A \mid \overrightarrow{B}} \tag{9}$$

$$3: \quad \boxed{\overrightarrow{A} \mid \cdot} \quad \rightarrow \quad \boxed{A \mid \rightarrow} \tag{10}$$

$$4a \; : \quad \boxed{\begin{array}{c} \rightarrow \\ 1 \end{array} \; \bullet} \quad \rightarrow \quad \boxed{\begin{array}{c} \blacktriangleleft \\ 1 \end{array} \; \bullet} \qquad\qquad 4b \; : \quad \boxed{\begin{array}{c} \rightarrow \\ 0 \end{array} \; \bullet} \quad \rightarrow \quad \boxed{\begin{array}{c} \triangleleft \\ 0 \end{array} \; \bullet} \tag{11}$$

$$5a \; : \quad \boxed{\begin{array}{c} A \quad \blacktriangleleft \\ \psi_{j,j+1} \end{array}} \quad \rightarrow \quad \boxed{\begin{array}{c} \blacktriangleleft \quad A \\ A(\psi_{j,j+1}) \end{array}} \qquad 5b \; : \quad \boxed{A \;\; \triangleleft} \quad \rightarrow \quad \boxed{\triangleleft \;\; A} \tag{12}$$

$$6a \; : \quad \boxed{\begin{array}{c} \bullet \quad \blacktriangleleft \\ 1 \end{array}} \quad \rightarrow \quad \boxed{\begin{array}{c} \bullet \quad \rightarrow \\ 1 \end{array}} \qquad 6b \; : \quad \boxed{\begin{array}{c} \bullet \quad \triangleleft \\ 0 \end{array}} \quad \rightarrow \quad \boxed{\begin{array}{c} \bullet \quad \rightarrow \\ 0 \end{array}} \tag{13}$$

Here, A, $B \in \mathcal{G}$, and $\psi_{j,j+1}$ refers to the state of qubits j and $j+1$ and $A(\psi_{j,j+1})$ refers to the gate A applied to the state of qubits j and $j+1$. We will name this set of forward transition rules $\mathcal{F}_I$

2.5 Active Site

A subset of the basis states for P_j whose symbols contain directional arrows of some form, $\mathcal{A}_I = \{\rightarrow, \overrightarrow{W}, \overrightarrow{I}, \overrightarrow{S}, \blacktriangleleft, \triangleleft\}$, are called *active symbols*. By construction, the initial state and all states resulting from applying a sequence of transition rules to the initial state will always contain exactly one active symbol, the site at which it is located being referred to as the *active site* of the chain. All transition rules for the HQCA involve the propagation of the active site to a neighbouring site and/or transforming one active symbol into another. This feature will be helpful in analysing the construction below.

2.6 Illustration

Now we illustrate the use of these transition rules on our example initial state to demonstrate features of the construction and how the HQCA evolves. We start with initial state $|\psi_0\rangle$ described by the table in equation 7:

$$|\psi_0\rangle \quad = \quad \begin{bmatrix} \rightarrow & S & W & I & I & W & S & I & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ 1 & 0 & 0 & 0 & 1 & w_1 & w_2 & w_3 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \tag{14}$$

Applying the sequence of forward transition rules 1, 2 (6 times), and 3, we reach the state,

$$|\psi_8\rangle \quad = \quad \begin{bmatrix} \bullet & S & W & I & I & W & S & I & \rightarrow & \bullet & \bullet & \bullet & \bullet & \bullet \\ 1 & 0 & 0 & 0 & 1 & w_1 & w_2 & w_3 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{15}$$

At this point, transition rule $4a$ applies, turning the $\rightarrow$ symbol into the gate application symbol $\blacktriangleleft$, giving

$$|\psi_9\rangle \quad = \quad \begin{bmatrix} \bullet & S & W & I & I & W & S & I & \blacktriangleleft & \bullet & \bullet & \bullet & \bullet & \bullet \\ 1 & 0 & 0 & 0 & 1 & w_1 & w_2 & w_3 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{16}$$

Now, rule $5a$ is applied 3 times, which applies the first round of gates to the work qubits:

$$|\psi_{12}\rangle \quad = \quad \begin{bmatrix} \bullet & S & W & I & I & \blacktriangleleft & W & S & I & \bullet & \bullet & \bullet & \bullet & \bullet \\ 1 & 0 & 0 & 0 & 1 & & S_{23}\left(W_{12}\left(w_1 w_2 w_3\right)\right) & & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{17}$$

10

Applying rule *5a* four more times will have propagated the active site as far back to the left as it can go, giving

$$|\psi_{16}\rangle \;=\; \begin{bmatrix} \bullet & \blacktriangleleft & S & W & I & I & W & S & I & \bullet & \bullet & \bullet & \bullet & \bullet \\ 1 & 0 & 0 & 0 & 1 & & S_{23}\,(W_{12}\,(w_1 w_2 w_3)) & & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad (18)$$

and applying rule *6a* yields

$$|\psi_{17}\rangle \;=\; \begin{bmatrix} \bullet & \rightarrow & S & W & I & I & W & S & I & \bullet & \bullet & \bullet & \bullet & \bullet \\ 1 & 0 & 0 & 0 & 1 & & S_{23}\,(W_{12}\,(w_1 w_2 w_3)) & & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (19)$$

So, we see that applying this *unique* sequence of 17 forward transitions has moved the active site $\rightarrow$ from the far left of the gate sequence over the to the far right of the gate sequence (equation 15), converted the active symbol $\rightarrow$ into the active symbol $\blacktriangleleft$ which is the gate applying symbol (equation 16), which then moved back to the left applying gates to the qubits below, shifting the gate sequence a single site to the right as it went (equations 17, 18 and 19). We will refer to this single back and forth movement of the active site as a right-moving *oscillation* of the active site. The halfway point of an oscillation, just before the $\rightarrow$ symbol turns around to move left as $\blacktriangleleft$ (as illustrated in equation 15) is called the *turning point* of the oscillation.

From the above illustration of the first oscillation, one can now see that applying this same sequence of 17 forward transitions over and over again will simply generate further right moving oscillations of the active site that proceed nearly identically, shifting the entire gate sequence one space to the right with each completed oscillation, until the entire gate sequence has been shifted all the way to the right. For example, after the second right-moving oscillation, the state of the chain will be

$$|\psi_{34}\rangle \;=\; \begin{bmatrix} \bullet & \bullet & \rightarrow & S & W & I & I & W & S & I & \bullet & \bullet & \bullet & \bullet \\ 1 & 0 & 0 & 0 & 1 & & S_{23}\,(W_{12}\,(w_1 w_2 w_3)) & & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad (20)$$

and after the third oscillation,

$$|\psi_{51}\rangle \;=\; \begin{bmatrix} \bullet & \bullet & \bullet & \rightarrow & S & W & I & I & W & S & I & \bullet & \bullet & \bullet \\ 1 & 0 & 0 & 0 & 1 & & S_{23}\,(W_{12}\,(w_1 w_2 w_3)) & & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad (21)$$

after the fourth,

$$|\psi_{68}\rangle \;=\; \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \rightarrow & S & W & I & I & W & S & I & \bullet & \bullet \\ 1 & 0 & 0 & 0 & 1 & & S_{23}\,(W_{12}\,(w_1 w_2 w_3)) & & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad (22)$$

and finally after the fifth,

$$|\psi_{85}\rangle \;=\; \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \rightarrow & S & W & I & I & W & S & I & \bullet \\ 1 & 0 & 0 & 0 & 1 & & U\,(w_1 w_2 w_3) & & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (23)$$

Note that the first round of gates was applied to the work qubits during the first oscillation, no gates were applied during oscillations 2, 3, or 4, and the second round of gates was applied during the fifth oscillation, completing the application of the desired U. Whether or not a particular oscillation will apply gates to the data qubits in the second half of the oscillation depends on whether or not there is a 1 or 0 in the data qubit below the active symbol when it reaches the

far right of the oscillation. We can see in equations 15 and 16, the active symbol $\rightarrow$ becomes an apply gate symbol ◄ because there is a 1 in the data register of the active site at the turning point. If there were a zero instead, which is true at the turning points of the next three oscillations, $\rightarrow$ would turn into ◁ and no gates would be applied in the second half of the oscillation. This is good because the second round of gates is not yet aligned with the work qubits (see equations 19, 20 and 23), and we do not want to be applying any gates until this alignment is achieved.

So, we see that the pattern of ones and zeros in the data qubits to the right of the work qubits is simply chosen so that transition rules 4a and 4b can properly enforce that gates are only applied for the first in every $N+1$ oscillations, which is precisely when rounds of gates will properly be aligned to the work qubits that they will be applied to. The pattern of the data qubits to the left of the work qubits mirrors that on the right so that transition rules 6a and 6b can properly convert ◄ or ◁ back into $\rightarrow$ at the end of the oscillation. Finally, note also that whenever we begin an oscillation that applies gates, rounds of gates that are not being applied during that round are aligned so that they are always applied to $|00\rangle$ qubit states, on which the S and W gates both act trivially. So, we see that the states on the non-work qubits in the data register remain invariant under the application of our HQCA transition rules.

Finally, after one more half oscillation (8 steps) we reach a state $|\psi_{93}\rangle$ to which no forward transition rules apply:

$$|\psi_{93}\rangle \;=\; \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & S & W & I & I & W & S & I & \rightarrow \\ 1 & 0 & 0 & 0 & 1 & U\,(w_1 w_2 w_3) & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{24}$$

2.7 General Case

Now that we understand the construction via the above worked example, we can make some statements about the general case where we are applying a depth-K unitary to N work qubits. In general, if we want to apply depth-K unitary in the form of equation 6 to N qubits, we initialize the state of the length $L = (2K-1)(N+1)+2$ chain of $d = 20$ qudits as follows:

- Program register P:
  $P_1 = \rightarrow$, $P_2...P_{K(N+1)} = (U_{K,1}\ldots U_{K,N-1})II(U_{K-1,1}\ldots U_{K-1,N-1})II\cdots II(U_{1,1}\ldots U_{1,N-1})I$,
  $P_{j>K(N+1)} = \cdot$

- Data register D: $D_1...D_{(K-1)(N+1)+1} = (1(0)^N)^{K-1}1$, $D_{(K-1)(N+1)+2}...D_{(K-1)(N+1)+1+N} = 0^N$
  (these are the work qubits), $D_{(K-1)(N+1)+1+N+1}...D_{L=(2K-1)(N+1)+2} = 1(0^N1)^{K-1}0$

Consider an instance of construction I with a K-depth unitary being applied to N work qubits. A single full right-moving oscillation will be achieved through the application of $2K(N+1)+1$ forward transitions: Rule 1 once and then rule 2 $K(N+1)-2$ times, and then rule 3 once completes the first half of the oscillation where the active symbol moves all the way to the right and gets to the turning point. Then applying rule $4a/b$ once prepares us for the second half of the oscillation where the active site moves all the way back to the left which is accomplished by applying rule $5a/b$ $K(N+1)-1$ times and then applying $6a/b$ once to prepare the active site to begin the next oscillation. The first half of a right-moving oscillation is $K(N+1)$ steps and the second half is $K(N+1)+1$ steps. The final state is reached after $N(K-1)+K$ full oscillations and one half oscillation (of $K(N+1)$ steps). So, the total number of forward transitions needed to reach the end state from the start state is $T_I = 2N^2K^2 - 2N^2K + 4NK^2 - N + 2K^2 + 2K$.

12

Note that at each step of the sequence of forward transition moves applied on a proper start state $|\psi_0\rangle$, there is only ever exactly one forward transition that can be applied (unless you have reached the end state $|\psi_{T_I}\rangle$). This is, essentially, due to the fact that the start state has a single active symbol, and the transition rules were specifically designed to move and transform the active symbol in a prescribed way.

Nagaj and Wocjan [9] show that by time evolving the appropriate initial state $|\psi_0\rangle$ encoding the desired unitary under Hamiltonian H_I for time polynomial in K and N, a simple measurement on a subset of the chain will collapse the state of the work qubits to $U|0^N\rangle$ with high probability. We defer any run-time analysis to the end of the paper where we will present the run-time analysis for our full construction which is based on this one.

2.8 Hamiltonian and Hilbert Space Geometry

Let us choose a Hamiltonian H_I for this system as a sum of translationally invariant terms:

$$H_I = -\frac{1}{6bL} \sum_{i=1}^{L-1} \sum_{k=1}^{6b} \left(P_k + P_k^\dagger \right)_{(i,i+1)} \tag{25}$$

where the terms $P_{k\,i,i+1}$ correspond to the rules 1-6b and act on two neighboring qudits as

$$P_{1\,i,i+1} = \sum_{A \in \{W,S,I\}} \left| \boldsymbol{\cdot}\, \overrightarrow{A} \right\rangle \left\langle \to A \right|_{p_i,p_{i+1}} \otimes \mathbb{I}_{d_i,d_{i+1}}, \tag{26}$$

$$P_{2\,i,i+1} = \sum_{A,B \in \{W,S,I\}} \left| A\, \overrightarrow{B} \right\rangle \left\langle \overrightarrow{A}\, B \right|_{p_i,p_{i+1}} \otimes \mathbb{I}_{d_i,d_{i+1}}, \tag{27}$$

$$P_{3\,i,i+1} = \sum_{A \in \{W,S,I\}} \left| A \to \right\rangle \left\langle \overrightarrow{A}\, \boldsymbol{\cdot} \right|_{p_i,p_{i+1}} \otimes \mathbb{I}_{d_i,d_{i+1}}, \tag{28}$$

$$P_{4a\,i,i+1} = \left| \blacktriangleleft\, \boldsymbol{\cdot} \right\rangle \left\langle \to\, \boldsymbol{\cdot} \right|_{p_i,p_{i+1}} \otimes |1\rangle \langle 1|_{d_i} \otimes \mathbb{I}_{d_{i+1}}, \tag{29}$$

$$P_{4b\,i,i+1} = \left| \triangleleft\, \boldsymbol{\cdot} \right\rangle \left\langle \to\, \boldsymbol{\cdot} \right|_{p_i,p_{i+1}} \otimes |0\rangle \langle 0|_{d_i} \otimes \mathbb{I}_{d_{i+1}}, \tag{30}$$

$$P_{5a\,i,i+1} = \sum_{A \in \{W,S,I\}} \left| \blacktriangleleft A \right\rangle \left\langle A\, \blacktriangleleft \right|_{p_i,p_{i+1}} \otimes A_{d_i,d_{i+1}}, \tag{31}$$

$$P_{5b\,i,i+1} = \sum_{A \in \{W,S,I\}} \left| \triangleleft A \right\rangle \left\langle A\, \triangleleft \right|_{p_i,p_{i+1}} \otimes \mathbb{I}_{d_i,d_{i+1}}, \tag{32}$$

$$P_{6a\,i,i+1} = \left| \boldsymbol{\cdot} \to \right\rangle \left\langle \boldsymbol{\cdot}\, \blacktriangleleft \right|_{p_i,p_{i+1}} \otimes \mathbb{I}_{d_i} \otimes |1\rangle \langle 1|_{d_{i+1}}, \tag{33}$$

$$P_{6b\,i,i+1} = \left| \boldsymbol{\cdot} \to \right\rangle \left\langle \boldsymbol{\cdot}\, \triangleleft \right|_{p_i,p_{i+1}} \otimes \mathbb{I}_{d_i} \otimes |0\rangle \langle 0|_{d_{i+1}}. \tag{34}$$

Each local term in the sum making up H, when acting on a basis state of $\mathcal{C}$, either changes the state by implementing the corresponding HQCA forward/reverse transition rule or annihilates it if the rule can not be applied to that state. The following definition will be essential for further understanding and analyzing the time evolution generated by this Hamiltonian and those based on it later in the paper:

Definition 7. *Consider the Hilbert space of states $\mathcal{C}$ of any HQCA described in this paper. A sequence of $T+1$ states from $\mathcal{C}$, $\{|\psi_t\rangle\}_{t=0}^{T}$, is called* Uniquely Orthogonally Generated *(UOG) by sequence of forward transition rules of the HQCA, $\mathcal{S}$ (which has length T), if the two following conditions are satisfied:*

13

1. *The states $\{|\psi_t\rangle\}_{t=0}^{T}$ are each a product state over all sites and registers aside from the work qubits, and are mutually orthogonal (thus the states differ in at least one site of the chain)*

2. *For each state $|\psi_{t<T}\rangle$, there is exactly one forward transition rule that can be applied to the state, and the application of this rule yields the state $|\psi_{t+1}\rangle$*

It is clear from the illustration of a single right-moving oscillation as in the example above that the sequence of intermediate states of the chain throughout a right-moving oscillation is UOG by the sequence of transitions described above.

We will name the unique sequence of forward transitions that takes the initial state of the chain, $|\psi_0\rangle$, to the final state $|\psi_{T_I}\rangle$ $\mathcal{T}_I \in \mathcal{F}_I^{T_I}$. This sequence $\{|\psi_t\rangle\}_{t=0}^{T_I}$ is UOG by $\mathcal{T}_I$. This is easily seen from the fact that for each individual right-moving oscillation, the set of transitions that implements it is unique, and for each successive oscillation the sets of intermediate states are orthogonal to those of all other oscillations because the program sequence starts at a different position in each oscillation. Thus, every state in the sequence $\{|\psi_t\rangle\}_{t=0}^{T_I}$ is orthonormal to all others in the sequence. Further, for each state in $\{|\psi_t\rangle\}_{t=0}^{T_I}$, only one forward transition can be applied at each step because this was true of the individual right-moving oscillations. Thus $\{|\psi_t\rangle\}_{t=0}^{T_I}$ is UOG by $\mathcal{T}_I$.

Now we can see the benefit of the UOG property. The fact that the sequence of states $\{|\psi_t\rangle\}_{t=0}^{T_I}$ is UOG by $\mathcal{T}_I$ allows one to be able to think of the set of states $|\psi_t\rangle$ as the set of positions of a particle on a line, so H_I becomes

$$H_{I,line} = - \sum_{t=0}^{T-1} \left(|t\rangle \langle t+1| + |t+1\rangle \langle t| \right). \tag{35}$$

This is the Hamiltonian of a (continuous-time) quantum walk on a line of length $T_I + 1$. Therefore, the HQCA Hamiltonian H_I induces a continuous quantum walk on the "line" of states $\{|\psi_t\rangle\}_{t=0}^{T_I}$ of the qudit chain of length L. Note that starting from an appropriately defined start state $|\psi_0\rangle$, the outcome of any time evolution under H_I is restricted to the subspace of states spanned by $\{|\psi_t\rangle\}_{t=0}^{T_I}$.

2.9 Summary and Conclusions

- Given a depth K unitary on N qubits, we can create a simple initial state $|\psi_0\rangle$ on a qudit chain of length $L = (2K-1)(N+1)+2$ whose local site dimension is $d = 20$ (itself factorizing locally into a qubit and a $d = 10$ qudit) which encodes the desired unitary

- The forward transition rules and their reverse versions define the Hamiltonian H_I which encodes an HQCA

- Starting from $|\psi_0\rangle$ and applying only forward transition rules, one obtains a *unique* sequence of mutually orthogonal states $\{|\psi_t\rangle\}_{t=0}^{T_I-1}$. For each $|\psi_t\rangle$, there is only ever exactly one forward transition rule that can be applied, from which one obtains $|\psi_{t+1}\rangle$

- The continuous time evolution of $|\psi_0\rangle$ under H_I can be mapped to a simple 1D quantum walk of a single particle on a chain of length $T_I + 1$

14

- The state $|\psi_{T_I}\rangle$ has the desirable feature that it is a product state among all sites and registers except for the work qubits, and the work qubits have had the desired unitary applied to them. No forward transition rule applies to $|\psi_{T_I}\rangle$.

- One can do a simple computational basis measurement on the program register to see if the program sequence has moved far enough to the right for U to have been applied to the qubits

3   Construction II - Applying U More Than Once

What is needed in order to apply U more than once? In the construction that has been made so far, we see that after applying U once to the state of the work qubits, the state of the HQCA reaches a state for which no forward transition rules can be applied. In the language of the quantum walk, the particle has reached the end of the line, so all it can do is turn around and walk the other way, which would undo the application of U to the work qubits. So under arbitrarily long time evolution U will never be applied to the work qubits more than once. To apply U twice, we would desire to be in a state like the start state in Eq. (14) except with U already applied to the work qubits. So, if there were some way of taking the end state in Eq. (24) and then somehow move the program sequence back to its starting position *without undoing the application of U to the work qubits*, then we could apply forward transition rules to apply U again. Here, we modify the construction to allow this. We proceed by adding a new set of local basis states to the program register Hilbert states and transition rules that will allow the program sequence to reset itself without undoing the application of U. We will refer to the construction described in this section as 'construction II'.

3.1   Hilbert Space

The following changes are made to the local program register Hilbert space: we add the new basis states represented by symbols $\{\overset{\leftarrow}{W}, \overset{\leftarrow}{S}, \overset{\leftarrow}{I}, \triangleright, \leftarrow, \circlearrowleft\}$. The length of the chain is increased by two sites.

3.2   Initial State

We must now define a new initial state to account for the fact that we have added two new sites. We will illustrate with the same example U being applied to three work qubits. The example initial state for our new construction will be

$$
|\psi_0\rangle \;=\; \begin{bmatrix} \circlearrowleft & \to & S & W & I & I & W & S & I & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & w_1 & w_2 & w_3 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{36}
$$

The rules for constructing this state are identical to those for the base construction except the two ends of the chain have extra sites added with the program register in state $\circlearrowleft$ and data register in state 0.

3.3   Transition Rules

We add the following new forward transition rules:

$$
7: \quad \boxed{A \,\big|\, \leftarrow} \quad \to \quad \boxed{\overset{\leftarrow}{A} \,\big|\, \bullet} \tag{37}
$$

$$8: \boxed{B \;|\; \overleftarrow{A}} \;\rightarrow\; \boxed{\overleftarrow{B} \;|\; A} \tag{38}$$

$$9: \boxed{\,.\; |\; \overleftarrow{A}} \;\rightarrow\; \boxed{\leftarrow \;|\; A} \tag{39}$$

$$10: \boxed{\,.\; |\; \leftarrow} \;\rightarrow\; \boxed{\,.\; |\; \triangleright} \tag{40}$$

$$11: \boxed{\triangleright \;|\; A} \;\rightarrow\; \boxed{A \;|\; \triangleright} \tag{41}$$

$$12: \boxed{\triangleright \;|\; .} \;\rightarrow\; \boxed{\leftarrow \;|\; .} \tag{42}$$

$$13a: \boxed{\rightarrow \;|\; \circlearrowleft} \;\rightarrow\; \boxed{\leftarrow \;|\; \circlearrowleft} \qquad 13b: \boxed{\circlearrowleft \;|\; \leftarrow} \;\rightarrow\; \boxed{\circlearrowleft \;|\; \rightarrow} \tag{43}$$

The forward transition rules 6-12 are designed to mimic the reverse versions of rules 1-6 using the newly introduced basis symbols. Rule 13 shows conversions between the $\rightarrow$ and $\leftarrow$ symbols via the $\circlearrowleft$ symbol, called the turn symbol.

3.4 Illustration

We now show how the additions made to our construction allow us to repeatedly apply U multiple times. Starting with our new start state

$$|\psi_0\rangle \;=\; \begin{bmatrix} \circlearrowleft & \rightarrow & S & W & I & I & W & S & I & . & . & . & . & . & . & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & w_1 & w_2 & w_3 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \tag{44}$$

the same sequence of forward transitions $\mathcal{T}_I$ that took Eq. (14) to Eq. (24) yields

$$|\psi_{T_I}\rangle \;=\; \begin{bmatrix} \circlearrowleft & . & . & . & . & . & . & S & W & I & I & W & S & I & \rightarrow & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U\,(w_1w_2w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{45}$$

Now, whereas in the previous construction, no forward transition rule could apply to $|\psi_{T_I}\rangle$, here there is one forward transition rule that applies to $|\psi_{T_I}\rangle$: rule 13a, which brings us to state $|\psi_{T_I+1}\rangle$:

$$|\psi_{T_I+1}\rangle \;=\; \begin{bmatrix} \circlearrowleft & . & . & . & . & . & . & S & W & I & I & W & S & I & \leftarrow & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U\,(w_1w_2w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{46}$$

Now from state $|\psi_{T_I+1}\rangle$ there is a unique set of T_I forward transitions through mutually orthogonal states that brings us to state $|\psi_{2T_I+1}\rangle$

$$|\psi_{2T_I+1}\rangle \;=\; \begin{bmatrix} \circlearrowleft & \leftarrow & S & W & I & I & W & S & I & . & . & . & . & . & . & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U\,(w_1w_2w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{47}$$

The set of transitions $\mathcal{T}'_I$ that does this is chosen from rules 7-12. Note that each rule 7-12 is similar to the reverse of one of the rules 1-6. To obtain the set of forward transitions that takes

16

$|\psi_{T_I+1}\rangle \to |\psi_{2T_I+1}\rangle$, take the set of forward transitions that took $|\psi_0\rangle \to |\psi_{T_I}\rangle$, $\mathcal{T}_I$, reverse it, and replace each transition from 1-6 with its partner from 7-12. Basically, this is undoing everything that happened in the transitions that took $|\psi_0\rangle \to |\psi_{T_I}\rangle$ but not undoing the application of gates to the work qubits, because there are no apply gate symbols in the transitions 7-12. It is clear, then that the sequence of states $\{|\psi_{T_I+1+t}\rangle\}_{t=0}^{T_I}$ is UOG by $\mathcal{T}_I'$ for the same reason that $\{|\psi_t\rangle\}_{t=0}^{T_I}$ is UOG by $\mathcal{T}_I$. From state $|\psi_{2T_I+1}\rangle$, there is only one forward transition rule that applies: 13b, which brings us to

$$|\psi_{2T_I+2}\rangle \;\; = \;\; \begin{bmatrix} \circlearrowleft & \to & S & W & I & I & W & S & I & . & . & . & . & . & . & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U\,(w_1 w_2 w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \; (48)$$

Now, we see that the state is identical in structure to the initial state $|\psi_0\rangle$, but U has already been applied to the work qubits. So, applying the same set of transitions that took $|\psi_0\rangle \to |\psi_{2T_I+2}\rangle$, we should get

$$|\psi_{4T_I+4}\rangle \;\; = \;\; \begin{bmatrix} \circlearrowleft & \to & S & W & I & I & W & S & I & . & . & . & . & . & . & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U^2\,(w_1 w_2 w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \; (49)$$

This set of forward transitions can be able to be applied over and over again, which will apply U to the work qubits again and again. That is, applying this set of transitions x times should yield the state

$$\left|\psi_{x(2T_I+2)}\right\rangle \;\; = \;\; \begin{bmatrix} \circlearrowleft & \to & S & W & I & I & W & S & I & . & . & . & . & . & . & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U^x\,(w_1 w_2 w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} (50)$$

3.5 Active Site

In the sequence $\mathcal{T}_I'$ which resets the gate sequence, the sequence of right moving oscillations that occurred from the sequence $\mathcal{T}_I$ play out in reverse, but with the active symbol pointing in the opposite direction this time. Seeing one of the symbols $\mathcal{A}_R = \{\to, \overrightarrow{W}, \overrightarrow{I}, \overrightarrow{S}, \blacktriangleleft, \lhd\}$ at the active site in a state indicates that the system is in the process of applying right moving oscillations that shift the gate sequence to the right, applying the gate sequence to the work qubits when appropriate as it goes, whereas seeing one of the symbols $\mathcal{A}_L = \{\leftarrow, \overleftarrow{W}, \overleftarrow{I}, \overleftarrow{S}, \rhd\}$ indicates that the system is in the process of applying left moving oscillations that shift the gate sequence to the left and not applying any gates to the work qubits.

3.6 Conclusion: There is A Catch

We see that we have built a sequence of states $\{|\psi_t\rangle\}_{t=0}^{T_{II}=2T_I+2}$, for which each state has a unique forward transition described by the sequence of transitions $\mathcal{T}_{II} = \{\mathcal{T}_I, 13a, \mathcal{T}_I', 13b\}$, and for which $|\psi_{T_{II}}\rangle$ is identical to $|\psi_0\rangle$ except that U has been applied to the work qubits. We see, then, that by applying the set of transitions $\mathcal{T}_{II}$ repeatedly, the work qubits have U applied to them repeatedly.

The above conclusion would tempt one to say that by building a new Hamiltonian H_{II} analogous to H_I but with the new additional transition rules, we again have a system whose time evolution is analogous to a 1D single particle quantum walk where the farther the particle walks from the starting point of the chain $|\psi_0\rangle$, the more times U will be applied to the work qubits. This, however, is incorrect. For the states $\{|\psi_t\rangle\}_{t=0}^{xT_{II}}$ to map to adjacent position eigenstates of a particle

on a chain to, the states must be UOG by $\mathcal{T}_{II}^x$. However, they are not. The only difference in state $|\psi_t\rangle$ and state $|\psi_{t+kT_{II}}\rangle$ is the number of times that U has been applied to the work qubits. For any quantum state $|\phi\rangle$, $|\phi\rangle$ and $U^k|\phi\rangle$ will not generally be orthogonal.

As a result, if we time evolve the state $|\psi_0\rangle$ with H_{II}, states that have U applied to the work qubits a different number of times can interfere with each other, and can not be distinguished by a simple measurement. There is no clear way to do a measurement to collapse the time evolved state into one where U has been applied a certain number of times. We will remedy this problem by further augmenting our construction to include a binary clock that 'counts' the number of times that U has been applied and orthogonalizes the states which have had U applied a different number of times.

4   Construction III: Binary Clock Construction

We would like to modify Construction II so that by applying forward transition rules, all of the uniquely obtained states of the chain at each step, $|\psi_t\rangle$, are mutually orthogonal, and as a particular consequence, states of the chain where the work qubits have had U applied a different number of times will be orthogonal, and can be distinguished by doing a computational basis state measurement on a subset of the chain that excludes the work qubits. We will accomplish this by further modifying the construction in the previous section. First, we give an overview of how the clock itself should work, then we describe how to implement its control and integration into our existing construction, and then we will prove its correctness. Similar binary clock constructions appear in [19, 20, 21].

4.1   Hilbert Space

The clock, itself, will consist of two registers/layers: the clock register (denoted C), whose local sites C_j are four dimensional with basis $\{\ \centerdot\ , X, 0, 1\}$, and the clock pointer register (denoted CP), whose local sites CP_j are five dimensional with basis states $\{\ \centerdot\ , X, L, R, C\}$. So we add these two new registers to our system (C being the third layer, CP being the fourth). We will also augment the program register's local Hilbert space basis to include the symbol $\Downarrow$.

So, our complete local Hilbert space description is as follows: There are $L = (2K-1)(N+1)+4$ local sites on the chain, each site j consisting of four different local Hilbert spaces, one from each register:

- The program register's local sites P_j have basis $\{W, S, I, \overrightarrow{W}, \overrightarrow{S}, \overrightarrow{I}, \blacktriangleleft, \triangleleft, \centerdot, \rightarrow, \circlearrowright, \overleftarrow{I}, \overleftarrow{W}, \overleftarrow{S}, \triangleright, \leftarrow, \Downarrow\}$

- The data register's local sites D_j have basis $\{0, 1\}$

- The clock register's local sites C_j have basis $\{\ \centerdot\ , 0, 1\}$

- The clock pointer register's local sites CP_j have basis $\{\ \centerdot\ , X, L, R, C\}$

The total local site dimension is $16 \times 2 \times 3 \times 5 = 480$.

4.2 Initial State

The initial state for this augmented construction is as follows:

$$|\psi_0\rangle = \begin{bmatrix} \circlearrowleft & . & . & . & . & . & . & S & W & I & I & W & S & I & \circlearrowleft & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & w_1 & w_2 & w_3 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ . & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ . & R & . & . & . & . & . & . & . & . & . & . & . & . & . & . \end{bmatrix}, \quad (51)$$

Here, the third layer is the clock register and the fourth and final layer is the clock pointer register. Notice that the program register has a different initial configuration than the previous two constructions: this time, the gate sequence starts shifted all the way to the right, and the clock and clock pointers are specifically initialized so that no reverse transition rule applies to the initial state. This will be illustrated below.

4.3 Clock Schematic and Update Algorithm

Here we describe a simple algorithm for updating a binary counter which will be implemented in our HQCA transition rules. The idea is the following: The clock register, consisting of L sites, can store a binary number of up to $L - 1$ digits long. Numbers with $n < L$ significant digits will be stored in the n rightmost sites of the clock register via the 0 and 1 states, and the remaining $L - n$ sites will all be in the state 0 except for the first register which will always be .. Significant bits will increase from right to left, so, for example, the binary representation of the number 6, in our clock, will look like .0...00000110.

Given any binary number represented in this form, we can use the following schematic algorithm for incrementing it by 1:

- A) Initialize a 'pointer' which starts sitting underneath the least significant bit.

- B) If the least significant bit is 0, change it to a 1, end

- C) If the least significant bit is 1, and the next bit is 0, update both bits $10 \to 01$, end

- D) Otherwise, the first two bits are 11. In this case, move the pointer to the left, bit by bit, until it encounters a 1 bit whose next most significant bit is a 0. When it is found, update these two bits $01 \to 10$. Move the pointer back to the right, bit by bit, flipping all of the less significant 1 bits to 0 as it passes, until it is back under the least significant bit, end.

- E) If the number being incremented is .11...1, then the pointer halts when it reaches the most significant 1, as this number can not be incremented further.

In our construction, the CP register is where the pointer that scans through and increments the clock registers will live. Generally, when the clock is not being incremented, the CP register will be in the state . . ... . X. This indicates that the 'pointer' is underneath the least significant bit of the clock, and is inactive, X. The . symbol underneath all of the other bits simply indicates the absence of the clock pointer. When the clock pointer is scanning to the left, looking for a 1 bit whose next most significant bit is a 0, it will be in state L and will hop right past the . symbols until it encounters this configuration in the clock register. Then it updates the clock bits according to the algorithm above, the pointer in state L transitions to a pointer in state R, and will now start

19

scanning back to the right, flipping the less significant 1 bits it encounters along the way to 0 bits. Once it reaches the least significant bit, the clock pointer transitions to the C state to indicate that the clock update is complete, and then back to the X state, in which it will stay until it is once again called upon to increment the clock.

4.4 Transition Rules

Now we give the transition rules which implement the control of the clock as described above. These rules include a modification to rule 13:

$$13a : \boxed{\rightarrow \;|\; \circlearrowright} \;\rightarrow\; \boxed{\circlearrowright \;|\; \Downarrow} \qquad\qquad 13b : \boxed{\circlearrowright \;|\; \leftarrow} \;\rightarrow\; \boxed{\circlearrowright \;|\; \rightarrow} \tag{52}$$

$$14 : \begin{array}{|c|c|} \hline \circlearrowright & \Downarrow \\ \hline - & - \\ \hline - & - \\ \hline \cdot & X \\ \hline \end{array} \;\rightarrow\; \begin{array}{|c|c|} \hline \circlearrowright & \circlearrowright \\ \hline - & - \\ \hline - & - \\ \hline \cdot & L \\ \hline \end{array} \tag{53}$$

$$15 : \begin{array}{|c|c|} \hline \circlearrowright & \circlearrowright \\ \hline - & - \\ \hline - & 0 \\ \hline \cdot & L \\ \hline \end{array} \;\rightarrow\; \begin{array}{|c|c|} \hline \circlearrowright & \circlearrowright \\ \hline - & - \\ \hline - & 1 \\ \hline \cdot & C \\ \hline \end{array} \tag{54}$$

$$16 : \begin{array}{|c|c|} \hline \circlearrowright & \circlearrowright \\ \hline - & - \\ \hline 0 & 1 \\ \hline \cdot & L \\ \hline \end{array} \;\rightarrow\; \begin{array}{|c|c|} \hline \circlearrowright & \circlearrowright \\ \hline - & - \\ \hline 1 & 0 \\ \hline \cdot & C \\ \hline \end{array} \tag{55}$$

$$17 : \begin{array}{|c|c|} \hline - & - \\ \hline - & - \\ \hline 1 & 1 \\ \hline \cdot & L \\ \hline \end{array} \;\rightarrow\; \begin{array}{|c|c|} \hline - & - \\ \hline - & - \\ \hline 1 & 1 \\ \hline L & \cdot \\ \hline \end{array} \tag{56}$$

$$18 : \begin{array}{|c|c|} \hline not\ \circlearrowright & - \\ \hline - & - \\ \hline 0 & 1 \\ \hline \cdot & L \\ \hline \end{array} \;\rightarrow\; \begin{array}{|c|c|} \hline not\ \circlearrowright & - \\ \hline - & - \\ \hline 1 & 0 \\ \hline \cdot & R \\ \hline \end{array} \tag{57}$$

$$19 : \begin{array}{|c|c|} \hline not\ \circlearrowright & - \\ \hline - & - \\ \hline 0 & 1 \\ \hline R & \cdot \\ \hline \end{array} \;\rightarrow\; \begin{array}{|c|c|} \hline not\ \circlearrowright & - \\ \hline - & - \\ \hline 0 & 0 \\ \hline \cdot & R \\ \hline \end{array} \tag{58}$$

20

$$20: \quad \begin{array}{|c|c|} \hline \circlearrowleft & \circlearrowleft \\ \hline - & - \\ \hline 0 & 1 \\ \hline R & \cdot \\ \hline \end{array} \rightarrow \begin{array}{|c|c|} \hline \circlearrowleft & \circlearrowleft \\ \hline - & - \\ \hline 0 & 0 \\ \hline \cdot & C \\ \hline \end{array} \tag{59}$$

$$21: \quad \begin{array}{|c|c|} \hline \circlearrowleft & \circlearrowleft \\ \hline - & - \\ \hline - & - \\ \hline \cdot & C \\ \hline \end{array} \rightarrow \begin{array}{|c|c|} \hline \leftarrow & \circlearrowleft \\ \hline - & - \\ \hline - & - \\ \hline \cdot & X \\ \hline \end{array} \tag{60}$$

4.5 Active Site

In analyzing the correctness of the above transition rules for achieving our goals, it will again be useful to make use of the concept of the active site and active symbol. In our augmented construction, there will again only ever be exactly one active symbol in the state, which will indicate the active site. But this time, the active symbol can either be in the program register *or* the clock pointer register, depending on whether or not the clock is being updated. The set of active symbols is now the union of the active symbols for the P and CP registers: $\mathcal{A} = \mathcal{A}_P \cup \mathcal{A}_{CP}$, with $\mathcal{A}_P = \{\leftarrow, \rightarrow, \overleftarrow{W}, \overleftarrow{S}, \overleftarrow{I}, \overrightarrow{W}, \overrightarrow{S}, \overrightarrow{I}, \triangleleft, \blacktriangleleft, \triangleright, \Downarrow\}$ and $\mathcal{A}_{CP} = \{L, R, C\}$. Transition rules 14 and 21 describe the passing of the active symbol between the program and clock pointer registers. It is still true that every single transition rule describes the moving of an active symbol from one site to a neighbouring site and/or the conversion of one kind of active symbol into another, which is essential for our sequence of states to be UOG and for the dynamics of the HQCA to map to a quantum walk on a line.

4.6 Correctness

We claim the following: starting from the initial state

$$|\psi_0\rangle = \begin{bmatrix} \circlearrowleft & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & S & W & I & I & W & S & I & \circlearrowleft & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & w_1 & w_2 & w_3 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \cdot & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \cdot & R & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}, \tag{61}$$

and applying forward transition rules 1-21, one generates a unique and finite sequence of mutually orthogonal states $\{|\psi_t\rangle\}_{t \geq 0}$, whose states on all registers across all layers aside from the work qubits are orthogonal product states. Moreover,

A) The final state is

$$|\psi_F\rangle = \begin{bmatrix} \circlearrowleft & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & S & W & I & I & W & S & I & \circlearrowleft & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U^k(w_1w_2w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \cdot & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \cdot & L & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}, \tag{62}$$

where U has been applied to the work qubits $k = 2^{L-1}$ times, where $L = (2K-1)(N+1) + 4$ is the number of sites in the chain.

B) Suppose you define a Hamiltonian, H_{III}, out of terms that implement transition rules 1-21 and their Hermitian conjugates, and use this Hamiltonian to continuously time evolve the state $|\psi_0\rangle$. If, at any point in the time evolution, one measures the clock layer to hold the binary number k, and the first site of the CP layer to hold the state C, then the post measurement state of the work qubits is guaranteed to be in the state $U^k |w_1 w_2 w_3\rangle$.

I will separate the proof into several small pieces.

Lemma 1. *Starting from the state*

$$\left|S_0^{U,x}\right\rangle \;=\; \begin{bmatrix} \circlearrowleft & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & S & W & I & I & W & S & I & \leftarrow & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U^x\,(w_1 w_2 w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \cdot & & & & & & \text{binary } x & & & & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X \end{bmatrix} . \quad (63)$$

there is a sequence of states $S^{U,x} = \left\{\left|S_t^{U,x}\right\rangle\right\}_{t=0}^{n_U}$ UOG by sequence of n_U forward transitions $\mathcal{U}$ and whose final state is

$$\left|S_{n_U}^{U,x}\right\rangle \;=\; \begin{bmatrix} \circlearrowleft & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & S & W & I & I & W & S & I & \rightarrow & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U^{x+1}\,(w_1 w_2 w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \cdot & & & & & & \text{binary } x & & & & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X \end{bmatrix} \quad (64)$$

for any $x \in \{0, 1, \cdots, 2^{L-1} - 1\}$.

Proof. By the correctness of Construction II, the sequence of forward transitions $\mathcal{U} = \{\mathcal{T}_I', 13b, \mathcal{T}_I\}$ fulfills the requirements. $\square$

The sequence of forward transitions $\mathcal{U}$ described above will referred to as an *application transition* since it generates a single application of U to the work qubits, and $n_U = T_{II} - 1$ is the number of transitions in $\mathcal{U}$. The sequence of mutually orthogonal states $S^{U,x} = \left\{\left|S_t^{U,x}\right\rangle\right\}_{t=0}^{n_U}$ is called the xth application sequence.

Lemma 2. *Starting with a state of the form*

$$\left|S_0^{C,x}\right\rangle \;=\; \begin{bmatrix} \circlearrowleft & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & S & W & I & I & W & S & I & \circlearrowleft & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U^{x+1}\,(w_1 w_2 w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \cdot & & & & & & \text{binary } x & & & & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & L \end{bmatrix} \quad (65)$$

the sequence of states $S^{C,x} = \left\{\left|S_t^{C,x}\right\rangle\right\}_{t=0}^{n_{C,x}}$ is UOG by sequence of forward transitions $\mathcal{CU}_x$, with final state

$$\left|S_{n_{C,x}}^{C,x}\right\rangle \;=\; \begin{bmatrix} \circlearrowleft & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & S & W & I & I & W & S & I & \circlearrowleft & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U^{x+1}\,(w_1 w_2 w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \cdot & & & & & & \text{binary } x+1 & & & & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & C \end{bmatrix} \quad (66)$$

for all $x \in \{0, 1, \cdots, 2^{L-1}\}$.

Proof. Once a state of the form $\left|S_0^{C,x}\right\rangle$ is reached, there is no longer any active symbol in the program register, and the active symbol is now in the CP register.

First, observe that no matter what number the clock layer holds, there will only ever be one of four possibly valid forward transitions from a state of the form $\left|S_0^{C,k}\right\rangle$, which will be either rule 15, 16, or 17 depending on the values of the clock's least two significant bits. Referring to the clock update algorithm described earlier, rule 15 corresponds to situation B), rule 16 corresponds to situation C), and rule 17 corresponds to situation D).

If the least significant bit of the clock in state $\left|S_0^{C,x}\right\rangle$ is zero (least two significant bits are 10 or 00), then only rule 15 can be applied, resulting in state

$$\left|S_{n_{C,x}}^{C,x}\right\rangle \;=\; \begin{bmatrix} \circlearrowleft & . & . & . & . & . & . & S & W & I & I & W & S & I & \circlearrowleft & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U^{x+1} & (w_1w_2w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ . & & & & & & & binary\ x+1 & & & & & & & \\ . & . & . & . & . & . & . & . & . & . & . & . & . & . & C \end{bmatrix} \quad (67)$$

so we're done. (By examining transition rule 15, it's obvious that in any of these situations, applying the single transition rule correctly increments the clock's stored number, as per case *B* of the clock update algorithm described earlier.)

If the two least significant bits of the clock in state $\left|S_0^{C,x}\right\rangle$ are 01, then only rule 16 can be applied, resulting in state

$$\left|S_{n_{C,x}}^{C,x}\right\rangle \;=\; \begin{bmatrix} \circlearrowleft & . & . & . & . & . & . & S & W & I & I & W & S & I & \circlearrowleft & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U^{x+1} & (w_1w_2w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ . & & & & & & & binary\ x+1 & & & & & & & \\ . & . & . & . & . & . & . & . & . & . & . & . & . & . & C \end{bmatrix} \quad (68)$$

so we're done. (Again, by examining transition rule 16, it's obvious that in any of these situations, applying the single transition rule correctly increments the clock's stored number, as per case *C* of the clock update algorithm described earlier.)

If rules 15 and 16 don't apply to $\left|S_0^{C,x}\right\rangle$, then let $2 \leq p \leq \lceil \log_2(x) \rceil$ be the number of least significant bits of x that form an uninterrupted string of 1s. Then it is clear that there are $p-1$ unique forward transitions through $p-1$ applications of rule 17. Because $x < 2^{L-1}$, we know that $p < L-1$. Then, after the $p-1$ applications of rule 17, registers p and $p+1$ of the chain must fit the description of the initial state of rule 18, in which case rules 15-17 can not apply (after the first application of 17, the program registers won't have the $\circlearrowleft$'s needed in the program register to allow 15 or 16 to apply), therefore only rule 18 can be applied because it is the only remaining transition rule whose initial state has L in the clock pointer.

After rule 18 is applied, the clock pointer's active site is in R mode, so only rules 19 or 20 could potentially be applied. If $p = 2$, then only rule 20 can be applied because the existence of the $\circlearrowleft$ symbol in the program register above the R symbol will forbid application of rule 19. If $p > 2$, then after application of rule 18, we see that bits p and $p-1$ of the number stored in the clock (counting from the right) are exactly the requisite initial state for applying rule 19, so 19 is the only way forward. This will obviously be true for $p-2$ applications of rule 19, at which point the R symbol will have the $\circlearrowleft$ symbol in the program register above it, so only rule 20 will be able to be applied. In either case, at this point we will have implemented case E) of the clock increase

algorithm, and will be in state $\left|S_{n_{C,x}}^{C,x}\right\rangle$ as above. Note that the state of the program and data layers never changed at any point in this sequence because we only used rules 15 through 20, and they leave those layers invariant.

By having described all possibilities, we see that the sequence $S^{C,x}$ is UOG by the appropriate sequence of forward transitions $\mathcal{CU}_x$. $\qquad \square$

The sequence of forward transitions, $\mathcal{CU}_x$, that takes $\left|S_0^{C,x}\right\rangle \to \left|S_{n_{C,x}}^{C,x}\right\rangle$ is called a *clock transition*, and the sequence of mutually orthogonal states $S^{C,x} = S^{C,x} = \left\{ \left|S_t^{C,x}\right\rangle \right\}_{t=0}^{n_{C,x}}$ is called the *x*th *clock sequence*.

Lemma 3. *The states in sequences $S^{C,k}$ and $S^{C,j}$ are mutually orthogonal for $j \neq k$.*

Proof. Suppose that there were states $|\phi_j\rangle \in S^{C,j}$ and $|\phi_k\rangle \in S^{C,k}$ which are not orthogonal. This necessarily means that the configurations of the clock and clock pointer registers in both states are identical. This, then, implies that it is possible to find a sequence of forward transitions to apply to $|\phi_j\rangle$ that brings it to $\left|S_{n_{C,k}}^{C,k}\right\rangle$. But, then, this contradicts Lemma 2 which states that the clock update sequence $S^{C,k}$ with is UOG, as this would imply that there are at least two distinct forward transition rules that could be applied to $|\phi_j\rangle$. $\qquad \square$

Lemma 4. *The states in sequences $S^{P,k}$ and $S^{P,j}$ are mutually orthogonal for $j \neq k$.*

Proof. This is obvious because the clocks will hold different numbers in each sequence (j and k respectively). $\qquad \square$

Lemma 5. *Starting from the state*

$$\left|S_0^{C,2^{L-1}-1}\right\rangle = \begin{bmatrix} \circlearrowleft & . & . & . & . & . & . & S & W & I & I & W & S & I & \circlearrowleft & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U^{2^{L-1}}(w_1w_2w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ . & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ . & . & . & . & . & . & . & . & . & . & . & . & . & . & L \end{bmatrix} \quad (69)$$

there is a unique sequence of $L-2$ forward transitions which maps us to the final state

$$\left|S_0^{C,2^{L-1}-1}\right\rangle = \begin{bmatrix} \circlearrowleft & . & . & . & . & . & . & S & W & I & I & W & S & I & \circlearrowleft & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U^{2^{L-1}}(w_1w_2w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ . & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ . & L & . & . & . & . & . & . & . & . & . & . & . & . & . \end{bmatrix} \quad (70)$$

from which there are no possible forward transitions.

Proof. Clearly, applying rule 17 $L-2$ times does the job. $\qquad \square$

Lemma 6. *Starting from the initial state*

$$|\psi_0\rangle = \begin{bmatrix} \circlearrowleft & . & . & . & . & . & . & S & W & I & I & W & S & I & \circlearrowleft & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & w_1 & w_2 & w_3 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ . & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ . & R & . & . & . & . & . & . & . & . & . & . & . & . & . \end{bmatrix}, \quad (71)$$

and applying forward transition rules 1-21, one generates the sequence $\{|\psi_t\rangle\}_{t=0}^{F}$ which is UOG by some sequence of F transitions, $\mathcal{S}$, and whose final state is reaching the final state

$$|\psi_F\rangle = \begin{bmatrix} \circlearrowleft & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & S & W & I & I & W & S & I & \circlearrowleft & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & U^{2^{L-1}}(w_1w_2w_3) & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \cdot & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \cdot & L & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}. \tag{72}$$

Proof. By lemmas 1 through 5, we see that the sequence of forward transitions of length F, $\mathcal{S} = \{\mathcal{L}_{-1}, \{\mathcal{L}_x\}_{x=0}^{2^{L-1}-1}, \mathcal{L}_F\}$, with subsequences defined

$$\mathcal{L}_{-1} = \{19^{L-3}, 20, 21\} \tag{73}$$
$$\mathcal{L}_x = \{\mathcal{U}, 13a, 14, \mathcal{UC}_x, 21\} \tag{74}$$
$$\mathcal{L}_F = \{\mathcal{U}, 13a, 14, 17^{L-2}\} \tag{75}$$

does the job. $\qquad\square$

5   Construction IV: Deterministic Selection of x in time $\mathrm{poly}(x, N)$

Now we augment construction III so that a particular number x of applications of U can be selected to be applied to the work qubits, and we can obtain the state where this has happened more or less deterministically by time evolving for a time polynomial in x and N. To do this, we build in a new register that holds the number of desired applications x in binary form, and once the number in the clock register reaches x, forward transitions are modified so that although the quantum walk continues forward, gates are no longer applied to data qubits, so there is some $T < F$ such that for all $|\psi_{t>T}\rangle$, the work qubits will always have U^x applied to them.

5.1   Hilbert Space

We add two new registers in this construction: the target register T, and the a second clock register C_2. The local sites of both registers are three dimensional $\{0, 1, \cdot\}$. We also make the following additions of new states to the local sites of the other registers:

- The program register adds symbols $\{\overrightarrow{I}^{\times}, \overrightarrow{S}^{\times}, \overrightarrow{W}^{\times}, \triangleleft^{\times}, \rightarrow^{\times}, \overleftarrow{I}^{\times}, \overleftarrow{S}^{\times}, \overleftarrow{W}^{\times}, \triangleright^{\times}, \leftarrow^{\times}, \Downarrow^{\times}\}$

- The clock pointer register adds symbols $\{\overleftarrow{C}, CX, R^{\times}, C^{\times}, L^{\times}\}$

With these additions, the local dimensions of the program, data, clock, clock pointer, target and second clock pointer registers are now 27, 2, 3, 10, 3, and 3 respectively. The total local dimension of a site of the qudit chain $\mathcal{C}$ is thus $d = 27 \times 2 \times 3 \times 10 \times 3 \times 3 = 14580$.

5.2   Initial State

In the initial state we must now have initialization of the fifth and sixth layers, the target and second clock registers respectively. The target register is initialized to hold the binary representation of the target number of applications, x, with significant digits increasing from right to left, and

all places greater than the most significant 1 of x being padded by $\blacksquare$ symbols. The second clock register is initialized the same way the clock register was initialized in the previous construction.

$$|\psi_0\rangle \;=\; \begin{bmatrix} \circlearrowleft & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & S & W & I & I & W & S & I & \leftarrow & \circlearrowleft \\ 0 & 1 & 0 & 0 & 0 & 1 & w_1 & w_2 & w_3 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \blacksquare & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & X \\ \blacksquare & & & & & & & binary\ x & & & & & & & & \\ \blacksquare & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (76)$$

5.3 Transition Rules

Here are the added transition rules, including a modification of transition rule 21:

$$21: \quad \rightarrow \qquad\qquad\qquad\qquad (77)$$

$$22: \quad \rightarrow \qquad\qquad\qquad\qquad (78)$$

$$23a: \quad \rightarrow \qquad 23b: \quad \rightarrow \qquad (79)$$

$$23c: \quad \rightarrow \qquad\qquad\qquad\qquad$$

$$24a: \quad \rightarrow \qquad 24b: \quad \rightarrow \qquad (80)$$

26

24c :

not ↻	—
—	—
.	a
.	$\overleftarrow{C}$
not .	a

$\rightarrow$

not ↻	—
—	—
.	a
$\overleftarrow{C}$	.
not .	a

25 :

↻	—
—	—
—	a
.	C
—	$\bar{a}$

$\rightarrow$

↻	—
—	—
—	a
.	CX
—	$\bar{a}$

(81)

26 :

not ↻	—
—	—
—	a
.	$\overleftarrow{C}$
—	$\bar{a}$

$\rightarrow$

not ↻	—
—	—
—	a
.	CX
—	$\bar{a}$

(82)

27 :

—	—
—	—
—	a
CX	.
—	a

$\rightarrow$

—	—
—	—
—	a
.	CX
—	a

(83)

28 :

—	—
—	—
.	a
.	$\overleftarrow{C}$
.	a

$\rightarrow$

—	—
—	—
.	a
$\overleftarrow{C}$	.
.	a

(84)

29 :

—	not ↻
—	—
.	.
.	$\overleftarrow{C}$
.	.

$\rightarrow$

—	not ↻
—	—
.	.
$\overleftarrow{C}$	.
.	.

(85)

30 :

—	↻
—	—
—	—
.	$\overleftarrow{C}$
—	—
0	1

$\rightarrow$

—	↻
—	—
—	—
.	$R^{\times}$
—	—
0	1

(86)

$$31: \quad \boxed{\rightarrow^{\times} \mid A} \quad \rightarrow \quad \boxed{\textbf{.} \mid \vec{A}^{\times}} \tag{87}$$

$$32: \quad \boxed{\vec{A}^{\times} \mid B} \quad \rightarrow \quad \boxed{A \mid \vec{B}^{\times}} \tag{88}$$

$$33: \quad \boxed{\vec{A}^{\times} \mid \textbf{.}} \quad \rightarrow \quad \boxed{A \mid \rightarrow^{\times}} \tag{89}$$

$$34: \quad \boxed{\rightarrow^{\times} \mid \textbf{.}} \quad \rightarrow \quad \boxed{\vartriangleleft^{\times} \mid \textbf{.}} \tag{90}$$

$$35: \quad \boxed{A \mid \vartriangleleft^{\times}} \quad \rightarrow \quad \boxed{\vartriangleleft^{\times} \mid A} \tag{91}$$

$$36: \quad \boxed{\textbf{.} \mid \vartriangleleft^{\times}} \quad \rightarrow \quad \boxed{\textbf{.} \mid \rightarrow^{\times}} \tag{92}$$

$$37: \quad \boxed{A \mid \leftarrow^{\times}} \quad \rightarrow \quad \boxed{\overleftarrow{A}^{\times} \mid \textbf{.}} \tag{93}$$

$$38: \quad \boxed{B \mid \overleftarrow{A}^{\times}} \quad \rightarrow \quad \boxed{\overleftarrow{B}^{\times} \mid A} \tag{94}$$

$$39: \quad \boxed{\textbf{.} \mid \overleftarrow{A}^{\times}} \quad \rightarrow \quad \boxed{\leftarrow^{\times} \mid A} \tag{95}$$

$$40: \quad \boxed{\textbf{.} \mid \leftarrow^{\times}} \quad \rightarrow \quad \boxed{\textbf{.} \mid \vartriangleright^{\times}} \tag{96}$$

$$41: \quad \boxed{\vartriangleright^{\times} \mid A} \quad \rightarrow \quad \boxed{A \mid \vartriangleright^{\times}} \tag{97}$$

$$42: \quad \boxed{\vartriangleright^{\times} \mid \textbf{.}} \quad \rightarrow \quad \boxed{\leftarrow^{\times} \mid \textbf{.}} \tag{98}$$

$$43a: \quad \boxed{\rightarrow^{\times} \mid \circlearrowright} \quad \rightarrow \quad \boxed{\circlearrowright \mid \downdownarrows^{\times}} \qquad 43b: \quad \boxed{\circlearrowright \mid \leftarrow^{\times}} \quad \rightarrow \quad \boxed{\circlearrowright \mid \rightarrow^{\times}} \tag{99}$$

$$44: \quad
\begin{array}{|c|c|}
\hline
\circlearrowright & \downdownarrows^{\times} \\
\hline
- & - \\
\hline
- & - \\
\hline
\textbf{.} & X \\
\hline
- & - \\
\hline
- & - \\
\hline
\end{array}
\quad \rightarrow \quad
\begin{array}{|c|c|}
\hline
\circlearrowright & \circlearrowright \\
\hline
- & - \\
\hline
- & - \\
\hline
\textbf{.} & L^{\times} \\
\hline
- & - \\
\hline
- & - \\
\hline
\end{array}
\tag{100}$$

45 :

↻	↻
−	−
−	−
.	$L^\times$
−	−
−	0

$\rightarrow$

↻	↻
−	−
−	−
.	$C^\times$
−	−
−	1

(101)

46 :

↺	↻
−	−
−	−
.	$L^\times$
−	−
0	1

$\rightarrow$

↺	↻
−	−
−	−
.	$C^\times$
−	−
1	0

(102)

47 :

−	−
−	−
−	−
.	$L^\times$
−	−
1	1

$\rightarrow$

−	−
−	−
−	−
$L^\times$	.
−	−
1	1

(103)

48 :

not ↺	−
−	−
−	−
.	$L^\times$
−	−
0	1

$\rightarrow$

not ↺	−
−	−
−	−
.	$R^\times$
−	−
1	0

(104)

49 :

not ↺	−
−	−
−	−
$R^\times$	.
−	−
0	1

$\rightarrow$

not ↺	−
−	−
−	−
.	$R^\times$
−	−
0	0

(105)

50 :

↺	↻
−	−
−	−
$R^\times$	.
−	−
0	1

$\rightarrow$

↺	↻
−	−
−	−
.	$C^\times$
−	−
0	0

(106)

Whenever just a appears in a transition rule it must be one of $\{0,1\}$. Whenever a and $\bar{a}$ appear, one of a or $\bar{a}$ must be in $\{0,1\}$ and the remaining symbol is either the logical negation, or the $.$ symbol.

5.4 Discussion

Here's a brief outline of how this modified construction with these new transition rules works:

- Transition rules 22-30 implement a new part of the clock update phase by giving it a new function at the end: when the clock pointer has finished updating the clock and is in C mode, it then begins a second sweep to the left in $\overleftarrow{C}$ mode, comparing the number stored in the clock to the number stored in the target register bit by bit

- If the clock pointer discovers bits that do not match, it turns into CX mode, moves back to the right, and the active symbol moves back up to the program layer to begin the next unitary application as normal in construction III

- If the clock pointer in $\overleftarrow{C}$ mode determines that the clock and target registers match, it continues moving left until it reaches the left end of the chain, and this causes the clock pointer to transition to $R^\times$ mode

- Once the the clock pointer has transitioned to $R^\times$ mode, all future active symbols will have the $\times$ symbol attached and obey transition rules 31-50, which identically mimic transition rules 1-20 of construction III, but are modified slightly so that a unitary is never applied to the work qubits ever again (the presence of the $\times$ symbol on all active symbols indicates this), and now the clock pointer (operating in an identical fashion to how it did in construction III, without a phase where it compares to a target register) treats the the second clock register $C2$ as the clock register.

At this point, based on our in depth analysis of construction III, it is easy to check that the sequence of states generated by forward transition rules $1 - 50$ is UOG by the appropriate sequence of unique forward transitions. The outcome, then, is that the Hamiltonian H_{IV} based on transition rules 1-50 implements a quantum walk on a line whose length is $l = \exp(N, K)$ (the length of the line for constructions H_{III} and H_{IV} is exponential because the clock can count up to a time $\exp(N, K)$). Once the quantum walk has proceeded far enough for U to have been applied to the work qubits x times (this distance is $\text{poly}(x, N, K)$), the quantum walk can continue forward for $\exp(N, K)$ more steps without ever changing the state of the data and clock registers ever again. We complete the analysis of the construction below by proving that we only need to run the simulation for a time $\text{poly}(x, N, K)$ in order to be able to do a simple measurement on the chain that will collapse the state of the work qubits to one where U^x has been applied with high probability.

5.5 Run-Time Analysis

Here we prove that after running construction IV with an appropriate input state for a time $\tau = \text{poly}(x, N, K)$, a computational basis measurement on the clock register will return the result that x is stored in the clock register (and therefore U^x has been applied to the work qubits) with high probability.

This is accomplished by slightly modifying a lemma from [9]:

Lemma 7. *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 $\tau \leq \tau^*$ chosen uniformly at*

random, starting from position basis state $|0\rangle$ (one end of the line). The probability to measure a state $|m\rangle$ with $m > (1 - F)l$, $F \in [0, 1]$, is then bounded from below as $p^* \geq F - O\left(\frac{l}{\tau^*}\right) - O\left(\frac{1}{l}\right)$.

Proof. Let $p_\tau(m)$ be the probability of measuring state $|m\rangle$ at time τ in the above scenario. Then the time average of $p_\tau(m)$ for $0 \leq \tau \leq \tau^*$ is

$$\overline{p}_{\tau^*}(m) = \frac{1}{\tau^*} \int_0^{\tau^*} p_\tau(m) d\tau \qquad (107)$$

which, in the limit $\tau^* \to \infty$, converges to the limiting distribution [9]

$$\pi(m) = \frac{2 + \delta_{m,0} + \delta_{m,l-1}}{2(l+1)}, \qquad (108)$$

in the sense that

$$\sum_{m=0}^{l-1} |\overline{p}_{\tau^*}(m) - \pi(m)| \leq O\left(\frac{l}{\tau^*}\right). \qquad (109)$$

Then, the probability of measuring state $|m\rangle$ for $m > (1 - F)l$ for time $\tau < \tau^*$ chosen uniformly at random is

$$p^* = \sum_{m > (1-F)l} \overline{p}_{\tau^*}(m). \qquad (110)$$

We have that

$$\sum_{m=0}^{l-1} |\overline{p}_{\tau^*}(m) - \pi(m)| \geq \sum_{m > (1-F)l} |\overline{p}_{\tau^*}(m) - \pi(m)| \qquad (111)$$

$$\geq \left| \sum_{m > (1-F)l} (\overline{p}_{\tau^*}(m) - \pi(m)) \right| \qquad (112)$$

$$= \left| p^* - \sum_{m > (1-F)l} \frac{2 + \delta_{m,0} + \delta_{m,l-1}}{2(l+1)} \right| \qquad (113)$$

$$= \left| p^* - F + \frac{F}{l+1} + \frac{1}{2(l+1)} \right| \qquad (114)$$

$$\geq -p^* + F - \frac{F}{l+1} - \frac{1}{2(l+1)} \qquad (115)$$

But, $O\left(\frac{l}{\tau^*}\right) \geq \sum_{m=0}^{l-1} |\overline{p}_{\tau^*}(m) - \pi(m)|$, thus, we have that

$$p^* \geq F - O\left(\frac{l}{\tau^*}\right) - O\left(\frac{1}{l}\right), \qquad (116)$$

as desired. $\qquad\qquad\square$

We now consider the consequences of this lemma for the run-time of our simulation. The length of the chain that the states $|\psi_t\rangle$ of the simulator map onto position eigenstates $|t\rangle$ of is $l = \exp(N, K)$. For $x = \exp(N)$, parameters can easily be chosen such that the fraction F of l such that $\left|\psi_{t>(1-F)l}\right\rangle$ has had U^x applied to the work qubits is $1 - \exp(-N)$. Then by setting $\tau^* = \text{poly}(l)$ appropriately, and picking a uniformly random time between 0 and τ^*, we will succeed in applying U^x to the work qubits with probability $p^* \geq 1 - 1/\exp(\text{poly}(N, K))$. For all states $\left|\psi_{t>(1-F)l}\right\rangle$, which have had U^x applied to the work qubits, the clock register will hold the number x which was originally set in the target register. So we only need to do a computational basis measurement on the clock register and find it in the state holding the binary number x to know that we've succeeded.

Note that if we would like to simulate the application of unitary U^x for $x = \text{poly}(N)$, the above strategy does not actually seem very helpful: because $l = \exp(\text{poly}(N, K))$ is exponential, we need to wait for, on average, exponential time to guarantee high success probability even though we're simulating something that should only take polynomial time. This can easily be fixed. By modifying the start state of the simulator, $|\psi_0\rangle$, so that the $\cdot$ symbol in the target register is far enough to the right, one can exponentially reduce l. This is because once the target register starts functioning as a clock (after U^x has been applied), the length of the remaining distance the particle can travel is exponential in the number of sites to the right of the $\cdot$ symbol in the target register. The end state is not reached until the target register has counted up to the largest number that it can. So, one can place the $\cdot$ at the appropriate site (say, 3 sites to the left of the most significant digit of the number x stored in the target register) so that l will be polynomial in N, but will also be large enough relative to x so that one only needs τ^* to be polynomial to achieve a high rate of success. Note, however, that in this scenario, the probability of success will only be inverse polynomially close to unity rather than exponentially.

Finally, we mention that from the start state defined in equation (76), it is actually possible to apply reverse transitions to this start state which allows you to start running a 'clock check and update' phase in reverse using the reverse of transition rules 21, 25 and 20 (if the target register's least significant bit is 1), or rules 21, 27, 23, 24 and 20 (if the target register's least significant bit is 0). However, it is easy to check that in either case, only one unique sequence of no more than $3L$ reverse transitions can be applied to this state before reaching a state from which no further reverse transitions can be made. Thus the effect on run-time analysis is insignificant.

6   Acknowledgements

References

[1] Leo P. Kadanoff. Scaling laws for ising models near tc. *Physics*, 2(6), 1966.

[2] Toby Cubitt, Ashley Montanaro, and Stephen Piddock. Universal quantum hamiltonians. *eprint arXiv:quantph/1701.05182*, 2017.

[3] Richard Phillip Feynman. Simulating physics with computers. *Int. J. Theor. Phys.*, 21, 1982.

[4] David Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London A*, 4, 1985.

[5] Alexei Yu Kitaev. Quantum computations: algorithms and error corrections. *Russian Mathematical Surveys*, 52(6):1191–1249, 1997.

[6] C. M. Dawson and M. A. Nielsen. The Solovay-Kitaev algorithm. *eprint arXiv:quantph/0505030*, 2005.

[7] Andrew M Childs, Debbie Leung, Laura Mančinska, and Maris Ozols. Characterization of universal two-qubit hamiltonians. *arXiv preprint arXiv:1004.1645*, 2010.

[8] K. G. H. Vollbrecht and J. I. Cirac. Quantum simulators, continuous-time automata, and translationally invariant systems. *Phys. Rev. Lett.*, 100(1), 2008.

[9] Daniel Nagaj and Pavel Wocjan. Hamiltonian quantum cellular automata in one dimension. *Phys. Rev. A*, 78(3), 2008.

[10] Scott Aaronson. The complexity of quantum states and transformations: from quantum money to black holes. *eprint arXiv:quantph/1607.05256*, 2016.

[11] Fernando GSL Brandao, Aram W Harrow, and Michal Horodecki. Local random quantum circuits are approximate polynomial-designs. *arXiv preprint arXiv:1208.0692*, 2012.

[12] Yosi Atia and Dorit Aharonov. Fast-forwarding of hamiltonians and exponentially precise measurements. *arXiv preprint arXiv:1610.09619*, 2016.

[13] Juan Maldacena. The large-N limit of superconformal field theories and supergravity. *Adv. Theor. Math. Phys.*, 2, 1998.

[14] Leonard Susskind. Computational complexity and black hole horizons. *eprint arXiv:hepth/1402.5674*, 2014.

[15] Scott Aaronson and Leonard Susskind. *in preparation*, 2017.

[16] Dorit Aharonov. *private communication*, 2017.

[17] Dominic W Berry, Andrew M Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 792–809. IEEE, 2015.

[18] We thank Elizabeth Crosson for suggesting this modification.

[19] Dorit Aharonov, Daniel Gottesman, Sandy Irani, and Julia Kempe. The power of quantum systems on a line. *Comm. Math. Physics*, 287(1):41–65, 2009.

[20] Daniel Gottesman and Sandy Irani. The quantum and classical complexity of translationally invariant tiling and hamiltonian problems. *arXiv preprint arXiv:0905.2419*, 2010.

[21] Toby Cubitt, David Perez-Garcia, and Michael M. Wolf. Undecidability of the spectral gap (full version). *arXiv preprint arXiv:1502.04573*, 2015.