International Journal of Theoretical Physics, Vol. 38, No. 9, 1999

Quantum Automata: An Overview

Stanley Gudder[1]

Received January 13, 1999

Quantum state machines are introduced. Amplitudes of computational paths, computational bases, superposition states, and evolution operators are discussed. The main part of the paper develops a theory of quantum automata and their slightly more general versions, q-automata. Quantum languages and η-quantum languages, $0 \leq \eta < 1$, are studied. A method is given for reducing the size of the state space. Functions that can be realized as probability maps for q-automata are characterized. Quantum gates are discussed. A quantum pumping lemma is employed to show that there are regular languages that are not η-quantum, $0 \leq \eta < 1$. The paper closes with a list of open problems.

1. INTRODUCTION

Although the theory of quantum computers has been studied for the past 16 years [3, 4, 9, 10, 13, 14], it did not receive much attention until 1994 [7, 8, 11, 21]. Since that time, quantum computation has been subjected to intense investigation. Most of the previous research in this field has involved quantum Turing machines [6, 17, 19, 23], quantum logic gates [2, 10, 12, 13], quantum complexity theory [1, 5, 6, 21, 22], and quantum algorithms [9, 11, 20, 21]. It now appears to be desirable to go back to the foundations and to study simpler quantum machines such as quantum automata [16, 17]. In this paper we shall investigate some of the basic properties of quantum automata. In particular, we shall be concerned with properties of classical probabilistic automata that carry over to quantum automata [19].

Section 2 reviews elements of Hilbert space theory that will be needed in the sequel. Section 3 provides an introduction to quantum computing machines. Its intent is to give the reader a basic understanding of how these machines operate. We begin with the simplest possible case, namely a quantum

[1] Department of Mathematics and Computer Science, University of Denver, Denver, Colorado 80208; e-mail: sgudder@cs.du.edu.

0020-7748/99/0900-226 1$16.00/0

state machine (QSM). A QSM M has no input or output; it just evolves from one state to another in equally spaced time steps. Transition amplitudes are introduced and justified. The amplitude of a computational path is discussed and the probability that M is in a state s is defined. We then discuss the concepts of computational bases, superposition states, and evolution operators. Quantum interference and decoherence are introduced and clarified. A justification for why the evolution operator must be unitary is given. For theoretical purposes, a slightly more general machine called a q-state machine is introduced. We indicate why such machines are easier to work with than a QSM. We then consider a variation of a QSM called a quantum printer. Although they are still very simple, quantum printers have an input and output. Section 3 ends with a discussion of a finalizing QSM. Such machines possess a set of final states and they provide an introduction to the quantum automata of Section 4.

The main part of the paper is Section 4, which discusses quantum automata and their slightly more general versions, q-automata. Response functions for q-automata are characterized. Finalizing q-automata are developed and two types of languages called quantum languages and η-quantum languages, $0 \leq \eta < 1$, are defined. It is shown that the intersection of two quantum languages over the same alphabet is again a quantum language. We next introduce the concept of a generalized finalizing q-automaton and its corresponding generalized languages. Instead of starting at an initial state, these automata can start at a superposition of states. We note that if L is a generalized η'-quantum language for $0 \leq \eta' < 1$, then L is a generalized η-quantum language for every $0 < \eta < 1$. A method is given for reducing the size of the state space for a finalizing q-automaton without changing its language (if this is possible). Functions that can be realized as probability maps for q-automata are characterized.

Section 5 discusses quantum gates. It is shown that any finite-dimensional unitary operator can be decomposed into a product of quantum gates. This is a simplified version of a result given in ref. 6. A quantum pumping lemma [17] is employed to show that there are regular languages that are not η-quantum, $0 \leq \eta < 1$. Finally, Section 6 presents a list of open problems that are suggested by the work of the previous two sections. For brievity, some of the proofs are omitted and will be left for a future paper.

2. ELEMENTS OF HILBERT SPACE THEORY

We assume that the reader has some familiarity with Hilbert space theory. The purpose of this section is to quickly review this theory and to set the notation that will be used in the sequel. Although the present paper is con-

cerned with finite-dimensional Hilbert spaces, for generality and future reference we shall consider arbitrary Hilbert spaces in this section.

A *complex Hilbert space* is a complete inner product space H over the field of complex numbers C. We denote the inner product by $\langle \cdot, \cdot \rangle$ and assume that $\langle \cdot, \cdot \rangle$ is linear in the first argument. We denote the norm of a vector $\psi \in H$ by $\| \psi \|$ and say that two vectors $\psi, \phi \in H$ are *orthogonal* ($\psi \perp \phi$) if $\langle \psi, \phi \rangle = 0$. An *orthonormal basis* for H is a maximal set of mutually orthogonal vectors of norm 1. If S is an orthonormal basis for H, then any $\psi \in H$ has the unique representation.

$$\psi = \sum_{s \in S} \langle \psi, s \rangle s$$

Let H_1 and H_2 be Hilbert spaces. Suppose that there exists a Hilbert space $H_1 \otimes H_2$ with the following properties:

(a) There is a bilinear map from $H_1 \times H_2$ *to* $H_1 \otimes H_2$ written as $(\psi_1, \psi_2) \mapsto \psi_1 \otimes \psi_2$.

(b) For any bilinear map B from $H_1 \times H_2$ to a Hilbert space H, there is a unique linear transformation T: $H_1 \times H_2 \to H$ satisfying

$$T(\psi_1 \otimes \psi_2) = B(\psi_1, \psi_2)$$

Then $H_1 \otimes H_2$ is called the *tensor product* of H_1 and H_2. It can be shown that the tensor product exists and is unique to within an isomorphism. Moreover, the inner product on $H_1 \otimes H_2$ is defined by

$$\langle \phi_1 \otimes \phi_2, \psi_1 \otimes \psi_2 \rangle = \langle \phi_1, \psi_1 \rangle \langle \phi_2, \psi_2 \rangle$$

We next consider another important way of combining two Hilbert spaces H_1 and H_2. Suppose that there exists a Hilbert space $H_1 \oplus H_2$ with the following properties:

(a) There exist linear injections T_1: $H_1 \to H_1 \oplus H_2$ and T_2: $H_2 \to H_1 \oplus H_2$.

(b) If $\psi_1 \in T_1 H_1$ and $\psi_2 \in T_2 H_2$, then $\psi_1 \perp \psi_2$ and every vector $\psi \in H_1 \oplus H_2$ has the form $\psi = \psi_1 + \psi_2$ with $\psi_1 \in T_1 H_1$ and $\psi_2 \in T_2 H_2$.

Then $H_1 \oplus H_2$ is called the *direct sum* of H_1 and H_2. Again, it can be shown that the direct sum exists and is unique to within an isomorphism.

The norm of a bounded linear operator T: $H \to H$ is denoted by $\|T\|$. It is well known that if T_1: $H \to H$ and T_2: $H \to H$ are bounded linear operators, then $\|T_1 T_2\| \leq \|T_1\| \, \|T_2\|$. If T: $H \to H$ is a bounded linear operator, then its adjoint T^* is the unique bounded linear operator on H that satisfies

$$\langle T^* \psi, \phi \rangle = \langle \psi, T\phi \rangle$$

for all $\psi, \phi \in H$. A bounded linear operator T: $H \to H$ is an *isometry* if $T^*T = 1$, where 1 denotes the identity operator on H. It is clear that T is an

isometry if and only if T preserves the inner product; that is, $\langle T\psi, T\phi \rangle = \langle \psi, \phi \rangle$ for every $\psi, \phi \in H$.

Lemma 2.1

A linear operator T: $H \to H$ is an isometry if and only if $\|T\psi\| = \|\psi\|$ for every $\psi \in H$.

Proof.

If T is an isometry, then clearly $\|T\psi\| = \|\psi\|$ for every $\psi \in H$. Conversely, if $\|T\psi\| = \|\psi\|$, then

$$\langle (T^*T - 1)\psi, \psi \rangle = \langle T^*T\psi, \psi \rangle - \langle \psi, \psi \rangle = \|T\psi\|^2 - \|\psi\|^2 = 0$$

For every $\psi \in H$. It follows that $T^*T = 1$, so T is an isometry. ■

A linear operator U: $H \to H$ is *unitary* if $U^*U = UU^* = 1$. Thus, every unitary operator is an isometry. If H is finite dimensional, then $U^* U = 1$ implies that $UU^* = 1$, so every isometry is unitary. However, if H is infinite dimensional, then an isometry need not be unitary. We denote the Kronecker delta symbol by $\delta_{s,t}$.

Theorem 2.2. Let S be an orthonormal basis for the Hilbert space H. (a) A bounded linear operator U: $H \to H$ is an isometry if and only if $\langle Us, Ut \rangle = \delta_{s,t}$ for every $s, t \in S$. (b) A linear operator U: $H \to H$ is unitary if and only if U is an isometry and $\|U^*s\| = 1$ for every $s \in S$.

Proof. (a) If U is an isometry, then clearly $\langle Us, Ut \rangle = \delta_{s,t}$ for every $s, t \in S$. Conversely, suppose that $\langle Us, Ut \rangle = \delta_{s,t}$ for every $s, t \in S$. If $\psi \in H$, then $\psi = \Sigma \alpha_i s_i$, $\alpha_i \in \mathbb{C}$, $s_i \in S$. Hence,

$$\|U\psi\|^2 = \langle U \sum_i \alpha_i s_i, U \sum_j \alpha_j s_j \rangle = \sum_{ij} \alpha_i \alpha_j^* \langle Us_i, Us_j \rangle$$

$$= \sum_{ij} \alpha_i \alpha_j^* \delta_{ij} = \sum_i |\alpha_i|^2 = \|\psi\|^2$$

It follows from Lemma 2.1 that U is an isometry.

(b) If U is unitary, then clearly U is an isometry and $\|U^*s\| = 1$ for every $s \in S$. Conversely, suppose U is an isometry and $\|U^*s\| = 1$ for every $s \in S$. Letting $P = 1 - UU^*$, it is easy to check that P is a projection; that is, $P = P^* = P^2$. Now for every $s \in S$ we have

$$\|Ps\|^2 = \langle Ps, Ps \rangle = \langle P^*Ps, s \rangle = \langle Ps, s \rangle$$

$$= \langle (1 - UU^*)s, s \rangle = \langle s, s \rangle - \langle UU^*s, s \rangle$$

$$= 1 - \langle U^*s, U^*s \rangle = 1 - \|U^*s\|^2 = 0$$

Hence, $Ps = 0$ for every $s \in S$ and it follows that $P = 0$. Since $UU^* = 1$ and U is an isometry, U is unitary. ■

Corollary 2.3. Let H be a finite-dimensional Hilbert space and S be an orthonormal basis for H. A linear operator U: $H \to H$ is unitary if and only if $\langle Us, Ut\rangle = \delta_{s,t}$ for every $s, t \in S$.

In the sequel we shall denote the set of unitary operators on H by $\mathcal{U}(H)$. As we shall see, this type of operator plays an important role in the study of quantum automata.

3. QUANTUM STATE MACHINES

A *quantum state machine* (QSM) is a triple $M = (S, s_0, \delta)$ where S is a finite set of internal states for M, $s_0 \in S$ is a designated start state, and δ: $S \times S \to \mathbb{C}$ is a transition function. We interpret $\delta(s, t)$ as the amplitude that M performs a transition from s to t in one time step and the probability of such a transition is $|\delta(s, t)|^2$. We require that δ satisfies the condition

$$\sum_t \delta(s, t)\delta(s', t)^* = \delta_{s,s'} \tag{3.1}$$

for every $s, s' \in S$, where * denotes complex conjugation and $\delta_{s,s'}$ is the Kronecker delta.

We now give a justification for Eq. (3.1). When $s = s'$, (3.1) gives $\Sigma_t |\delta(s, t)|^2 = 1$, which says that M moves from s to some state with probability one. When $s \neq s'$, the left side of (3.1) vanishes and this is needed for the reversible evolution of M as required by quantum theory. Reversibility says that if an undisturbed machine M moves from state s to state t in n time steps, then after n time steps of running the machine in reverse it will move back to state s. Moreover, $\delta(s, t)^*$ gives the amplitude that M moves from t to s when M is run in reverse. Referring to Eq. (3.1), $\delta(s, t)\delta(s', t)^*$ is the amplitude that M moves from s to t and then back from t to s' in two time steps. Summing over t, $\Sigma_t \delta(s, t)\delta(s', t)^*$ is the total amplitude that M moves forward from s and then backward to s' in two time steps and the probability of this evolution is $|\Sigma_t \delta(s, t)\delta(s', t)^*|^2$. By reversibility, if $s = s'$, this probability is 1 and if $s \neq s'$, this probability is 0. It follows that (3.1) holds.

The machine M begins in its start state s_0 and enters a state $s_1 \in S$ with amplitude $\delta(s_0, s_1)$. At the second step, M scans its current state (say s_1) and enters a state $s_2 \in S$ with amplitude $\delta(s_1, s_2)$. The machine continues to evolve as long as desired. At any given time, M is in a definite state s, but this state cannot be observed without disturbing the later operation of M. (We will discuss this point in more detail later.) All we know is the amplitude (and hence the probability) that M is in state s at a given time.

The amplitude that M is in state s at time n is computed as follows. A *computational path* from s_0 to s is a finite sequence $s_0, s_1, \ldots, s_{n-1}, s$. The *amplitude* of this path is defined to be

$$\delta(s_0, s_1)\delta(s_1,s_2) \ldots \delta(s_{n-2}, s_{n-1})\delta(s_{n-1}, s)$$

and the *amplitude* $A_n(s)$ that M is in state s at time n is defined as the sum of the amplitudes over all paths from s_0 to s:

$$A_n(s) = \sum_{i_1,\ldots,i_{n-1}} \delta(s_0, s_{i_1})\delta(s_{i_1}, s_{i_2}) \cdots \delta(s_{i_{n-2}}, s_{i_{n-1}})\delta(s_{i_{n-1}}, s) \quad (3.2)$$

The *probability* that M is in state s at time n is given by $|A_n(s)|^2$. The expression $|A_n(s)|^2$ can indeed be interpreted as a probability because a repeated application of (3.1) shows that

$$\sum_s |A_n(s)|^2 = 1 \quad (3.3)$$

Quantum theory allows only certain types of predictions such as the probability $|A_n(s)|^2$. For example, we cannot predict the probability that M is in state s at time $m > 0$ and in the state t at time $n > m$. This is because such an event involves only the computational paths of length n that go through s at time m and t at time n. If we sum the amplitudes of these paths and take the square of its absolute value, then this number cannot be interpreted as a probability. In fact, the resulting number could be greater than one. The reason for this is that amplitudes of computational paths are complex numbers and summing them may give cancellations or reinforcements. In physical terms, paths can interfere and in general this phenomenon is called quantum interference. In a similar way, we cannot predict the probability of a particular computational path.

The derivation of (3.3) as well as the expression in (3.2) are cumbersome and can be given in a much more convenient form by introducing an evolution operator for M. Suppose that the cardinality $|S| = N$ and let H be an N-dimensional complex Hilbert space with unit sphere $\hat{H}$. Take an orthonormal basis for H and identify this basis with S. Thus, we can assume that S is an orthonormal basis for H which we call a *computational basis* for M. We call the elements of S *states* and the general elements of $\hat{H}$ *superposition states*. Of course, a state is also a superposition state in a trivial way. We construct the *evolution operator* U for M by defining

$$Us = \sum_t \delta(s, t)\, t \quad (3.4)$$

for all $s \in S$ and extending U to H by linearity. Our first result shows that Eq. (3.1) is precisely the condition needed for U to be unitary.

Lemma 3.1. The operator U: $H \to H$ is unitary if and only if (3.1) holds.

Proof. By Corollary 2.3, U is unitary if and only if $\langle Us, Ut\rangle = \delta_{s,t}$ for every $s, t \in S$. Since

$$\langle Us, Ut\rangle = \langle \sum_s \delta(s, s')s', \sum_t \delta(t, t')t'\rangle$$

$$= \sum_{s't} \delta(s, s')\delta(t, t')^* \langle s', t'\rangle = \sum_t \delta(s, t')\, \delta(t, t')^*$$

the result follows. ■

It is clear that the adjoint U^*: $H \to H$ of U is determined by

$$U^*t = \sum_s \delta(s, t)^* s$$

and since $U^*U = UU^* = 1$, U^* reverses the action of U. Since U^* is also unitary, it follows from Lemma 3.1 that the dual of Eq. (3.1) holds:

$$\sum_s \delta(s, t)\delta(s, t')^* = \delta_{t,t'}$$

The transition function δ determines the evolution operator U via Eq. (3.4). Conversely, δ can be retrieved from U because $\langle Us, t\rangle = \delta(s, t)$. In this way, δ and U contain the same information.

The unitary operator U describes the evolution of M as follows. Starting with state s_0, after one time step M is in the superposition state Us_0. After the second time step M is in the superposition state U^2s_0 and continuing, after n steps M is in the superposition state U^ns_0. In reality, M is always in a specific state $s \in S$ at any given time. However, s is unknown and all we know is that M is in superposition state $\psi = \Sigma\, \alpha_i s_i$, $s_i \in S$. In this case, M is in state s_i with amplitude $\langle \psi, s_i\rangle = \alpha_i$ and probability $\left|\langle \psi, s_i\rangle\right|^2 = \left|\alpha_i\right|^2$. If we try to observe the state at time m, then the superposition state U^ms_0 "collapses" into a definite state $s \in S$. The evolution then restarts at s and this changes the later operation of M. Thus, an observation can disturb the operation of M and this phenomenon is called quantum decoherence.

It follows from our previous discussion that the amplitude that M is in state s at time n becomes

$$\langle U^ns_0, s\rangle = \langle U^{n-1} \sum_{i_1} \delta(s_0, s_{i_1})s_{i_1}, s\rangle = \sum_{i_1} \delta(s_0, s_{i_1})\langle U^{n-1} s_{i_1}, s\rangle$$

$$= \sum_{i_1} \delta(s_0, s_{i_1})\langle U^{n-2} \sum_{i_2} \delta(s_{i_1}, s_{i_2})s_{i_2}, s\rangle$$

$$= \sum_{i_1, i_2} \delta(s_0, s_{i_1})\, \delta(s_{i_1}, s_{i_2})\langle U^{n-2} s_{i_2}, s\rangle$$

$$\vdots$$

$$= \sum_{i_1,\ldots,i_{n-1}} \delta(s_0, s_{i_1})\delta(s_{i_1}, s_{i_2}) \cdots \delta(s_{i_{n-2}}, s_{i_{n-1}})\delta(s_{i_{n-1}}, s)$$

and this agrees with Eq. (3.2). Thus, the complicated equation (3.2) can be replaced by the simple equation $A_n(s) = \langle U^n s_0, s\rangle$. Moreover, since U^n is unitary, we have

$$\sum_s \left|A_n(s)\right|^2 = \sum_s \left|\langle U^n s_0, s\rangle\right|^2 = \|U^n s_0\|^2 = \|s_0\|^2 = 1$$

which is a simple derivation of (3.3).

Due to the close connection between δ and U, we can give an alternative way of viewing a QSM. A *q-state machine* is a triple $M = (H, s_0, U)$, where H is a finite-dimensional Hilbert space, $s_0 \in \hat{H}$, and U: $H \to H$ is a unitary operator. This definition is more general than our previous definition of a QSM. Indeed, if $M' = (S, s_0, \delta)$ is a QSM, then as before we can form a Hilbert space H with computational basis S and a unitary operator U: $H \to H$ given by (3.4) to obtain a q-state machine $M = (H, s_0, U)$. Conversely, suppose $M = (H, s_0, U)$ is a q-state machine. Then there are many QSMs corresponding to M. Just let S be an orthonormal basis for H with $s_0 \in S$, define $\delta(s, t) = \langle Us, t\rangle$, and let $M' = (S, s_0, \delta)$. We may think of a q-state machine as a QSM in which the computational basis is left unspecified. A QSM is more basic than a q-state machine because a QSM is defined in terms of the transition amplitudes of the internal states of the machine M and these are the determining characteristics of M. These characteristic internal states are unspecified (except for s_0) in a q-state machine. However, as we have seen, a q-state machine is easier to work with for theoretical purposes and for this reason we shall frequently employ a q-state machine corresponding to a QSM.

A QSM does not have a very useful purpose, it has no input or output and just evolves. We now consider a more useful variant called a quantum printer. Let I be a finite alphabet that includes a blank symbol #. A *quantum printer* is a 4-tuple $P = (S, s_0, I, \delta)$ where S is a finite set of internal states, $s_0 \in S$ is a start state, I is an alphabet, and δ: $I \times S \times I \times S \to \mathbb{C}$ is a transition function that satisfies

$$\sum_{y,t} \delta(x, s, y, t)\, \delta(x', s', y, t)^* = \delta_{x,x'}\, \delta_{s,s'} \tag{3.5}$$

Of course, a quantum printer can be viewed as just a QSM in which the set of internal states is replaced by $I \times S$. However, we can think of a quantum printer as having an output. Suppose we have a finite tape divided into $N + 2$ cells numbered $-1, 0, 1, \ldots, N$. The quantum printer P has a tape head

that begins at cell 0 and moves one cell to the right at each time step and stops one time step after it enters cell N. (For a realistic printer, the tape head would be fixed and the tape would move to the left. However, our description is traditional and easier to visualize.) The original tape is blank in every cell so P begins in state s_0 with # in every cell. At time 0, P scans its current state s_0 and the # in cell -1. Then P prints letter y in cell 0 and enters state s with amplitude $\delta(\#, s_0, y, s)$ and moves to cell 1. Then P scans the printed letter, say y, in cell 0 and its current state, say s, prints letter z and enters state t with amplitude $\delta(y, s, z, t)$ and moves to cell 2. This process continues until P prints a letter in cell N and stops.

After P stops it certainly produces an output in terms of a printed word $w = x_1x_2 \cdots x_N$. However, quantum theory has nothing to say about the probability of w. This is because w involves only a restricted set of computational paths and quantum interference prevents us from associating probabilities to these paths. But we can predict the probability of the last letter or more generally the probability of any letter in a given cell. As with a QSM these are most easily given in terms of the associated evolution operator.

As before, we form a finite-dimensional complex Hilbert space H with an orthonormal basis identified with the elements of $I \times S$. Thus, $I \times S$ is the computational basis for P and we denote its elements by $x \otimes s$, $x \in I$, $s \in S$. We define the *evolution operator* U: $H \to H$ by

$$Ux \otimes s = \sum_{y,t} \delta(x, s, y, t)y \otimes t$$

and it follows from (3.5) and Lemma 3.1 that U is unitary. Now the superposition state at time $n \leq N$ is given by $U^n \# \otimes s_0$. Hence, the amplitude that P will print the letter x in the nth cell and find itself in state s at time n is

$$A_n(x, s) = \langle U^n \# \otimes s_0, x \otimes s \rangle$$

The corresponding probability becomes $\left|A_n(x, s)\right\}^2$. Now the probability that x is printed in the nth cell is

$$p_n(x) = \sum_s \left|A_n(x, s)\right|^2 = \sum_s \left|\langle U^n \# \otimes s_0, x \otimes s \rangle\right|^2$$

Notice that $p_n(x)$ can indeed be interpreted as a probability because

$$\sum_x p_n(x) = \sum_{x,s} \left|\langle U^n \# \otimes s_0, x \otimes s \rangle\right|^2 = \|U^n \# \otimes s_0\|^2 = \|\# \otimes s_0\|^2 = 1$$

A *finalizing* QSM is a 4-tuple $M_f = (S, s_0, \delta, S_f)$ where $M = (S, s_0, \delta)$ is a QSM and $S_f \subseteq S$ is a set of final states. The machine M_f halts when it is in a final state $s \in S_f$. As before, we can assume that S is a computational basis for M_f in a Hilbert space H. Since $s_0 \in S$ and $S_f \subseteq S$, we have that

either $s_0 \in S_f$ or $s_0 \in S_f^{\perp}$. Letting $F = \text{span}(S)$, we have that $s_0 \in F$ or $s_0 \in F^{\perp}$. We also define the unitary evolution operator U by Eq.(3.4) and arrive at the following definition. A *finalizing* q-state machine is a 4-tuple $M_f = (H, s_0, U, F)$ where $M = (H, s_0, U)$ is a q-state machine and F is a subspace of H such that $s_0 \in F$ or $s_0 \in F^{\perp}$.

We have just seen that any finalizing QSM corresponds to a finalizing q-state machine. Conversely, any finalizing q-state machine M_f corresponds to many finalizing QSMs depending on the chosen computational basis. Indeed, if $M_f = (H, s_0, U, F)$ and $s_0 \in F$, we can choose an orthonormal basis S_f for F that includes s_0 as an element, extend S_f to an orthonormal basis S for H, define $\delta(s, t) = \langle Us, t\rangle$ for every $s, t \in S$, and form the finalizing QSM $M_f' = (S, s_0, \delta, S_f)$. However, if $s_0 \in F^{\perp}$, we can choose an orthonormal basis S_f for F, extend S_f to an orthonormal basis S for H that includes s_0 as an element, define δ as before, and form the finalizing QSM $M_f' = (S, s_0, \delta, S_f)$. Now the probability that M_f is in a final state at time n is given by

$$p_n(F) = \sum_{s \in F} \left|\langle U^n s_0, s\rangle\right|^2 \tag{3.6}$$

Denoting the projection onto F by $P(F)$, we can rewrite (3.6) as

$$p_n(F) = \|P(F)U^n s_0\|^2 \tag{3.7}$$

If $p_n(F) = 1$, then M_f will halt with certainty at time n or earlier. The *certain halting time* for M_f is given by $\inf\{n: p_n(F) = 1\}$. It is sometimes of interest to let the final subspace F vary. In this case, p_n becomes a probability measure on the set of final subspaces. That is, $0 \leq p_n(F) \leq 1$, $p_n(H) = 1$, and

$$p_n(F + G) = p_n(F) + p_n(G)$$

whenever $F \perp G$.

4. QUANTUM AUTOMATA

Let I be a finite nonempty alphabet and let I^* be the set of all words with finitely many letters in I, including the empty word λ. Defining a product on I^* by concatenation and defining $\lambda w = w\lambda = w$ for all $w \in I^*$, I^* becomes a semigroup. A *quantum automaton* (QA) is a 4-tuple $\mathcal{A} = (S, s_0, I, \delta)$, where S is a finite set of internal states, $s_0 \in S$ is the start state, I is a finite input alphabet, and δ: $I \times S \times S \to \mathbb{C}$ is a transition function that satisfies

$$\sum_t \delta(x, s, t)\delta(x, s', t)^* = \delta_{s,s'} \tag{4.1}$$

for all $x \in I$ and $s, s' \in S$. When an input word w is fed into $\mathcal{A}$, $\mathcal{A}$ operates

as follows. After $\mathcal{A}$ scans the first letter x of w and its start state s_0, $\mathcal{A}$ updates its state to s with amplitude $\delta(x, s_0, s)$. Next, $\mathcal{A}$ scans the second letter y of w and its current state, say s, and updates its state to t with amplitude $\delta(y, s, t)$. This process is continued until all the letters of w are scanned. It is convenient to assume that words are read from right to left. Thus, if a word $x_n x_{n-1} \cdots x_1 \in I^*$ is fed into $\mathcal{A}$, then $\mathcal{A}$ first scans x_1, next $\mathcal{A}$ scans $x_2, \ldots,$ and finally $\mathcal{A}$ scans x_n.

Assume that the word $w = x_n x_{n-1} \cdots x_1$ is fed into $\mathcal{A}$. The *amplitude* of a computational path $p = (s_0, s_1, \ldots, s_{n-1}, s)$ is defined to be

$$A(p|w) = \delta(x_1, s_0, s_1)\delta(x_2, s_1, s_2) \cdots \delta(x_n, s_{n-1}, s)$$

The amplitude that $\mathcal{A}$ ends up in state s is the sum of these amplitudes $A(p|w)$ for all such paths p from s_0 to s and is given by

$$A(s|w) = \sum_{i_1,\ldots,i_{n-1}} \delta(x_1, s_0, s_{i_1})\delta(x_2, s_{i_1}, s_{i_2}) \cdots \delta(x_n, s_{i_{n-1}}, s) \tag{4.2}$$

The corresponding probability is $|A(s|w)|^2$ and (4.1) ensures that this can indeed be interpreted as a probability. That is,

$$\sum_{s \in S} |A(s|w)|^2 = 1 \tag{4.3}$$

As in Section 3, we can assume that S is a computational basis for a Hilbert space H. For $x \in I$ define the operator $U(x)$: $H \to H$ by

$$U(x)s = \sum_t \delta(x, s, t)t$$

for all $s \in S$. As in Lemma 3.1, Eq. (4.1) is a necessary and sufficient condition for $U(x)$ to be unitary. We call the map U: $I \to \mathcal{U}(H)$ given by $x \mapsto U(x)$ a *transition operator*. Of course, δ can be retrieved from U because $\delta(x, s, t) = \langle U(x)s, t\rangle$. We extend the domain of U from I to I^* by defining $U(\lambda) = 1$ and

$$U(x_n x_{n-1} \cdots x_1) = U(x_n)U(x_{n-1}) \cdots U(x_1)$$

Since a product of unitary operators is unitary, $U(w) \in \mathcal{U}(H)$ for all $w \in I^*$. Notice that $U(uv) = U(u)U(v)$ for all $u, v \in I^*$. The amplitude that $\mathcal{A}$ ends up in state s after being fed a word w becomes

$$A(s, w) = \langle U(w)s_0, s\rangle$$

and this is consistent with (4.2). Moreover, (4.3) holds because

$$\sum_{s \in S} |A(s|w)|^2 = \sum_{s \in S} |\langle U(w)s_0, s\rangle|^2 = \|U(w)s_0\|^2 = \|s_0\|^2 = 1$$

A *q-automaton* is a 4-tuple $\mathcal{A} = (H, s_0, I, U)$ where H is a finite-

dimensional complex Hilbert space, $s_0 \in \hat{H}$, I is a finite alphabet, and U: $I \to \mathcal{U}(H)$. As in Section 3, there is a close connection between QAs and q-automata. A *finalizing* QA is a pair $\mathcal{A}_f' = (\mathcal{A}', S_t) = (S, s_0, I, \delta, S_f)$ where $\mathcal{A} = (S, s_0, I, \delta)$ is a QA and $S_f \subseteq S$ is a set of final states. Similarly, a *finalizing* q-automaton is a pair $\mathcal{A}_f = (\mathcal{A}, F) = (H, s_0, I, U, F)$ where $\mathcal{A}' = (H, s_0, I, U)$ is a q-automaton and F is a subspace of H such that $s_0 \in F$ or $s_0 \in F^\perp$. Again, there is a close connection between $\mathcal{A}_f$ and $\mathcal{A}_f'$. Because of their convenience we shall usually work with q-automata and our definitions and results can be easily translated for QAs. If $\mathcal{A}_f$ is a finalizing q-automaton, then as in (3.7), the probability that $\mathcal{A}_f$ reaches a final state when fed a word w is given by

$$p_{\mathcal{A}}(F|w) = \|P(F)U(w)s_0\|^2 \tag{4.4}$$

The *response function* for a q-automaton $\mathcal{A} = (H, s_0, I, U)$ is the function $R_{\mathcal{A}}$: $I^* \to \hat{H}$ given by $R_{\mathcal{A}}(w) = U(w)s_0$. The superposition state $R_{\mathcal{A}}(w)$ is the one in which $\mathcal{A}$ finds itself when fed the word w. A function R: $I^* \to \hat{H}$ is *realizable* by a q-automaton $\mathcal{A}$ if $R = R_{\mathcal{A}}$. A map G: $\hat{H} \to \hat{H}$ is $\perp$-*preserving* if $G(\psi_1) \perp G(\psi_2)$ whenever $\psi_1 \perp \psi_2$. We omit the proof of the following theorem.

Theorem 4.1. For a function R: $I^* \to \hat{H}$, the following statements are equivalent.

(a) R is realizable by a q-automaton.

(b) There exists a map U: $I \to \mathcal{U}(H)$ such that $R(xw) = U(x)R(w)$ for every $x \in I$, $w \in I^*$.

(c) There exists an orthonormal basis ψ_i for H such that for every $x \in I$ there is an orthonormal basis $\psi_i(x)$ with the property that $\langle R(xw), \psi_i\rangle = \langle R(w), \psi_i(x)\rangle$ for all $w \in I^*$.

(d) There exists a map G: $I \times \hat{H} \to \hat{H}$ such that $G(x, \cdot)$ is $\perp$-preserving and $\langle R(xw), \psi\rangle = \langle R(w), G(x, \psi)\rangle$ for every $x \in I$, $w \in I$,* $\psi \in \hat{H}$.

A word w is *accepted* by a finalizing q-automaton $\mathcal{A} = (H, s_0, I, U, F)$ if $R_{\mathcal{A}}(w) = U(w)s_0 \in F$. The proof of the following lemma is straightforward.

Lemma 4.2. If $\mathcal{A} = (H, s_0, I, U, F)$ is a finalizing q-automaton, then the following statements are equivalent.

(a) A word w is accepted by $\mathcal{A}$.

(b) $P(F)R_{\mathcal{A}}(w) = R_{\mathcal{A}}(w)$.

(c) $p_{\mathcal{A}}(F|w) = \|P(F)U(w)s_0\|^2 = 1$.

Thus, w is accepted by $\mathcal{A} = (H, s_0, I, U, F)$ if and only if $\mathcal{A}$ enters F with certainty upon receiving w. The *language accepted by* $\mathcal{A}$ is the set $L(\mathcal{A})$ of all words in I^* that are accepted by $\mathcal{A}$. Hence,

$$L(\mathcal{A}) = \{w \in I^*: R_{\mathcal{A}}(w) \in F\}$$

A language L is a *quantum language* if $L = L(\mathcal{A})$ for some finalizing q-automaton $\mathcal{A}$.

If $\mathcal{A}_i = (H_i, s_i, I, U_i, F_i)$ are finalizing q-automata, $i = 1, 2$, then their *tensor product* is the finalizing q-automaton given by

$$\mathcal{A}_1 \otimes \mathcal{A}_2 = (H_1 \otimes H_2, s_1 \otimes s_2, I, U_1 \otimes U_2, F_1 \otimes F_2)$$

where $(U_1 \otimes U_2)(x) = U_1(x) \otimes U_2(x)$.

Lemma 4.3. $L(\mathcal{A}_1 \otimes \mathcal{A}_2) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$.

Proof. For $w \in I^*$ we have

$$(U_1 \otimes U_2)(w)s_1 \otimes s_2 = U_1(w)s_1 \otimes U_2(w)s_2$$

Hence, $(U_1 \otimes U_2)(w)s_1 \otimes s_2 \in F_1 \otimes F_2$ if and only if $U_1(w)s_1 \in F_1$ and $U_2(w)s_2 \in F_2$. Thus, $w \in L(\mathcal{A}_1 \otimes \mathcal{A}_2)$ if and only if $w \in L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$ and the result follows. ■

Corollary 4.4. If L_1 and L_2 are quantum languages over the same alphabet, then $L_1 \cap L_2$ is a quantum language.

For a finalizing q-automaton $\mathcal{A} = (H, s_0, I, U, F)$ we required that $s_0 \in F \cup F^\perp$. If this requirement is relaxed and we allow F to be an arbitrary subspace of H, then $\mathcal{A}$ is called a *generalized finalizing* (*g-finalizing*) q-automaton. In general, a g-finalizing q-automaton does not directly correspond to a finalizing QA. The language $L(\mathcal{A})$ accepted by a g-finalizing q-automaton $\mathcal{A}$ is defined as before. Moreover, a language L is a *generalized quantum language* if $L = L(\mathcal{A})$ for some g-finalizing q-automaton. Let $\mathcal{A} = (H, s_0, I, U)$ be a q-automaton and let F be a subspace of H. We say that $\mathcal{A}$ *accepts* $w \in I^*$ *relative to* F if the g-finalizing q-automaton (H, s_0, I, U, F) accepts w. The set of words that $\mathcal{A}$ accepts relative to F is denoted by $L(\mathcal{A}; F)$. Of course, $L(\mathcal{A}; F)$ is a generalized quantum language. We denote the span of two subspaces E and F by $E \vee F$.

Lemma 4.5. Let E and F be subspaces of H and let $\mathcal{A} = (H, s_0, I, U)$ be a q-automaton.

(a) $L(\mathcal{A}; E \cap F) = L(\mathcal{A}; E) \cap L(\mathcal{A}; F)$.

(b) $L(\mathcal{A}; F^\perp) \subseteq I^* \backslash L(\mathcal{A}; F)$.

(c) $L(\mathcal{A}; E \vee F) \supseteq L(\mathcal{A}; E) \cup L(\mathcal{A}; F)$.

Proof. (a) Since $U(w)s_0 \in E \cap F$ if and only if $U(w)s_0 \in E$ and $U(w)s_0 \in F$, we have $w \in L(\mathcal{A}; E \cap F)$ if and only if $w \in L(\mathcal{A}; E) \cap L(\mathcal{A}; F)$. (b) Since $U(w)s_0 \in F^\perp$ implies that $U(w)s_0 \notin F$, we have that $w \in L(\mathcal{A}; F^\perp)$ implies $w \notin L(\mathcal{A}; F)$.

(c) Assume that $w \in L(\mathcal{A}; E) \cup L(\mathcal{A}; F)$. We then have that

$$U(w)s_0 \in E \cup F \subseteq E \vee F$$

Hence, $w \in L(\mathcal{A}; E \vee F)$. ■

Let $\mathcal{A}_i = (H_i, s_i, I, U_i, F_i)$ be g-finalizing q-automata over the same alphabet I. For $\alpha, \beta \in \mathbb{C}$ with $\alpha, \beta \neq 0$, $|\alpha|^2 + |\beta|^2 = 1$, form the *linear combination*

$$\alpha\mathcal{A}_1 + \beta\mathcal{A}_2 = (H_1 \oplus H_2, \alpha s_1 \oplus \beta s_2, I, U_1 \oplus U_2, F_1 \oplus F_2)$$

where $(U_1 \oplus U_2)(x) = U_1(x) \oplus U_2(x)$ for every $x \in I$. Then $\alpha\mathcal{A}_1 + \beta\mathcal{A}_2$ is again a g-finalizing q-automaton.

Lemma 4.6. $L(\alpha\mathcal{A}_1 + \beta\mathcal{A}_2) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$.

Proof. For $w \in L^*$ we have

$$\begin{aligned}(U_1 \oplus U_2)(w)(\alpha s_1 \oplus \beta s_2) &= (U_1(w) \oplus U_2(w))(\alpha s_1 \oplus \beta s_2) \\ &= \alpha U_1(w)s_1 \oplus \beta U_2(w)s_2\end{aligned}$$

Hence, $(U_1 \oplus U_2)(w)(\alpha s_1 \oplus \beta s_2) \in F_1 \oplus F_2$ if and only if $\alpha U_1(w)s_1 \in F_1$ and $\beta U_2(w)s_2 \in F_2$. Hence, $w \in L(\alpha\mathcal{A}_1 + \beta\mathcal{A}_2)$ if and only if $w \in L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$. ■

Corollary 4.7. If L_1 and L_2 are generalized quantum languages over the same alphabet, then $L_1 \cap L_2$ is a generalized quantum language.

The tensor product of two g-finalizing q-automata over the same alphabet is defined as before. Moreover, since the proof of Lemma 4.3 still holds, this gives another demonstration of Corollary 4.7.

Let $\mathcal{A} = (H, s_0, I, U, F)$ be a finalizing q-automaton. A word $w \in I^*$ is η-*accepted by* $\mathcal{A}$ where $0 \leq \eta < 1$ if

$$p_{\mathcal{A}}(F|w) = \|P(F)U(w)s_0\|^2 > \eta$$

For example, if w is accepted by $\mathcal{A}$, then w is η-accepted by $\mathcal{A}$ for every η with $0 \leq \eta < 1$. The *language* $L(\mathcal{A}, \eta)$ η-*accepted by* $\mathcal{A}$ is the set of all words η-accepted by $\mathcal{A}$. A language L is η-*quantum* if $L = L(\mathcal{A}, \eta)$ for some finalizing q-automaton $\mathcal{A}$. Generalized η-quantum languages are defined in the obvious way. The proof of the following theorem is omitted.

Theorem 4.8. If L is a generalized η'-quantum language for $0 \leq \eta' < 1$, then L is a generalized η-quantum language for every $0 < \eta < 1$.

Let $I = \{x_1, \ldots, x_n\}$ be a finite alphabet and form the alphabet

$$\hat{I} = \{x_1, \ldots, x_n, x_1', \ldots, x_n'\}$$

If we identify $x_i x_i'$ and $x_i' x_i$ with λ, $i = 1, \ldots, n$, then $\hat{I}^*$ becomes a group

(the free group over I). If $\mathscr{A} = (H, s_0, I, U, F)$ is a finalizing q-automaton, we can form $\hat{\mathscr{A}} = (H, s_0, \hat{I}, \hat{U}, F)$, where $\hat{U}(x_i) = U(x_i)$ and $\hat{U}(x_i') = U(x_i)^*$, $i = 1, \ldots, n$. We call $\hat{\mathscr{A}}$ a *finalizing group q-automaton*. As before, we extend $\hat{U}$: $I \to \mathscr{U}(H)$ to a map (also denoted by $\hat{U}$) from $\hat{I}^*$ to $\mathscr{U}(H)$. Then $\hat{U}$: $\hat{I}^* \to \mathscr{U}(H)$ is a unitary representation of the group $\hat{I}^*$ in H. Notice that $L(\hat{\mathscr{A}}) \cap I^* = L(\mathscr{A})$ and $L(\hat{\mathscr{A}}, \eta) \cap I^* = L(\mathscr{A}, \eta)$. Thus, any quantum or η-quantum language can be obtained from a finalizing group q-automaton. These definitions and the results to follow easily extend to g-finalizing q-automata.

To simplify notation, let $\mathscr{A} = (H, s_0, I, U, F)$ be a finalizing group q-automaton. Then $U(w^{-1}) = U(w)^*$ for every $w \in I^*$. Letting

$$H_0 = \text{span}\{U(w)s_0 \colon w \in I^*\}$$

we have that H_0 is a Hilbert space containing s_0. Form

$$\mathscr{A}_0 = (H_0, s_0, I, U_0, F \cap H_0)$$

where $U_0(x) = U(x)\big|H_0$ for all $x \in I$. The next result shows that $\mathscr{A}_0$ is a finalizing q-automaton that accepts the same language as $\mathscr{A}$.

Theorem 4.9. (a) U_0: $I^* \to \mathscr{U}(H_0)$ is a unitary representation of I^* in H_0 so that $\mathscr{A}_0$ is a finalizing q-automaton. (b) $L(\mathscr{A}) = L(\mathscr{A}_0)$. (c) If $P(F)P(H_0) = P(H_0)P(F)$, then $L(\mathscr{A}, \eta) = L(\mathscr{A}_0, \eta)$ for all $0 \leq \eta < 1$.

Proof. (a) It is clear that $U_0(w)H_0 \subseteq H_0$ and $U_0(w)^* H_0 \subseteq H_0$ for every $w \in I^*$. Moreover, for every $x \in I$ and $\psi \in H_0$ we have

$$U_0(x)U_0(x)^*\psi = U(x)U(x)^*\psi = \psi$$

so that $U_0(x)\, U_0(x)^* = 1_{H_0}$. Hence, $U_0(x) \in \mathscr{U}(H_0)$ for every $x \in I$. Since $F \cap H_0$ is a subspace of H_0 and $s_0 \in (F \cap H_0) \cup (F \cap H_0)^\perp$, it follows that $\mathscr{A}_0$ is a finalizing q-automaton. (b) It is clear that $L(\mathscr{A}_0) \subseteq L(\mathscr{A})$. If $w \in L(\mathscr{A})$, then

$$U_0(w)s_0 = U(w)s_0 \in F$$

Since $U_0(w)s_0 \in H_0$, we have $U_0(w)s_0 \in F \cap H_0$, so that $w \in L(\mathscr{A}_0)$. Hence, $L(\mathscr{A}) = L(\mathscr{A}_0)$. (c) Since $P(F \cap H_0) = P(F)P(H_0)$, we have

$$P(F \cap H_0)U_0\,(w)s_0 = P(F)P(H_0)U_0\,(w)s_0 = P(F)U\,(w)s_0$$

for every $w \in I^*$ and the result follows. ∎

Theorem 4.9 shows that if $H_0 \neq H$, then there exists a finalizing q-automaton $\mathscr{A}_0$ that has a smaller set of states than $\mathscr{A}$ but accepts the same quantum language as $\mathscr{A}$. In this case, the number of states for a computational

basis of $\mathcal{A}$ can be reduced. The next result shows that we can obtain the same result for an arbitrary q-automaton.

Corollary 4.10. Let $\mathcal{A} = (H, s_0, I, U, F)$ be a finalizing q-automaton and let

$$H_0 = \text{span}\{U(w)s_0, U(w)^*s_0, w \in I^*\}$$

Then $\mathcal{A}_0 = (H_0, s_0, I, U_0, F \cap H_0)$, where $U_0(x) = U(x)\big|H_0$ for every $x \in I$, is a finalizing q-automaton for which $L(\mathcal{A}) = L(\mathcal{A}_0)$. Moreover, if $P(F)P(H_0) = P(H_0)P(F)$, then $L(\mathcal{A}, \eta) = L(\mathcal{A}_0, \eta)$ for all $0 \leq \eta < 1$.

5. QUANTUM GATES

It is now clear that finite-dimensional unitary operators play an important role in the theory of quantum computers. However, in order to build an actual physical quantum computer a unitary operator must be implemented in practice. It appears that the best way to do this is to break a unitary operator down into simpler components called quantum gates.

Of course, a finite-dimensional unitary operator can be considered to be a unitary matrix, so for simplicity we shall restrict our attention to unitary matrices. We denote the standard basis on the n-dimensional Hilbert space $\mathbb{C}^n$ by $e_1, \ldots, e_n$. An $n \times n$ unitary matrix M is *basic* if M satisfies one of the following conditions.

1. M is the identity matrix except that one of its diagonal entries is $e^{i\theta}$ for some $\theta \in (-\pi, \pi]$.

2. M is the identity matrix except that the submatrix in one pair of distinct indices j and k of M is the rotation by some angle $\theta \in (-\pi, \pi]$:

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

A matrix of type 1 has the form $Me_j = e_j$ for $j \neq k$ and $Me_k = e^{i\theta}e_k$. A matrix of type (2) has the form $Me_r = e_r$ for $r \neq j, k$ and

$$Me_j = (\cos\theta)e_j + (\sin\theta)e_k$$

$$Me_k = (-\sin\theta)e_j + (\cos\theta)e_k$$

These basic unitary matrices are also called *quantum gates*. If M is of type 1, we call M a *basic phase shift*, and if M is of type 2, we call M a *basic rotation*. When M is a basic phase shift of $e^{i\theta}$ in the index j, we write $M = [j, j, \theta]$, and when M is a basic rotation of angle θ between indices j and k, we write $M = [j, k, \theta]$. The next result gives a well-known compact form for a basic rotation.

Lemma 5.1. For $\theta \in (-\pi, \pi]$ we have

$$M = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \exp\left\{\theta \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\right\}$$

Proof. The spectral representation of the rotation is

$$M = (\cos\theta + i\sin\theta)P_1 + (\cos\theta - i\sin\theta)P_2 = e^{i\theta}P_1 + e^{-i\theta}P_2$$

where P_1 and P_2 are the projections

$$P_1 = \frac{1}{2}\begin{bmatrix} 1 & i \\ -i & 1 \end{bmatrix}, \qquad P_2 = \frac{1}{2}\begin{bmatrix} 1 & -i \\ i & 1 \end{bmatrix}$$

Let

$$M_1 = \theta P_1 - \theta P_2 = i\theta \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Then

$$M = e^{iM_1} = \exp\left\{\theta \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\right\} \quad \blacksquare$$

We shall show that any unitary matrix can be written as a product of quantum gates. Before doing this, we consider the following example [10]. Deutsch has introduced the not gate

$$N = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Although N is not a quantum gate, we can write N as a product of quantum gates

$$N = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = [1, 1, \pi][1, 2, \pi/2]$$

More generally, let $\alpha \in [0, 2]$ and let N^α be a power of the not gate. Notice that since $N^2 = 1$, this includes all real powers of N. The spectral representation of N is $N = P_1 - P_2$, where P_1 and P_2 are the projections

$$P_1 = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \qquad P_2 = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Hence,

$$N^\alpha = P_1 + (-1)^\alpha P_2 = P_1 + e^{i\pi\alpha}P_2$$

$$= \frac{1}{2}\begin{bmatrix} 1 + e^{i\pi\alpha} & 1 - e^{i\pi\alpha} \\ 1 - e^{i\pi\alpha} & 1 + e^{i\pi\alpha} \end{bmatrix} = e^{i\pi\alpha/2}\begin{bmatrix} \cos \pi\alpha/2 & -i \sin \pi\alpha/2 \\ -i \sin \pi\alpha/2 & \cos \pi\alpha/2 \end{bmatrix}$$

For example, the square root of N is

$$N^{1/2} = \frac{1}{2}\begin{bmatrix} 1 + i & 1 - i \\ 1 - i & 1 + i \end{bmatrix}$$

We can write N^α as a product of quantum gates

$$N^\alpha = \begin{bmatrix} e^{i\pi\alpha/2} & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi\alpha/2} \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}\begin{bmatrix} \cos \pi\alpha/2 & -\sin \pi\alpha/2 \\ \sin \pi\alpha/2 & \cos \pi\alpha/2 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

$$= [1, 1, \pi\alpha/2][2, 2, \pi(\alpha - 1)/2]\,[1, 2, \pi\alpha/2][2, 2, \pi/2]$$

The following two results are based on work in ref. 6. We shall present a simpler version (and correct some minor errors) of their proofs. The reason that our version is simpler is that they show that this construction is computationally efficient, which we do not discuss here.

Lemma 5.2. (a) If $v = (v_1, v_2) \in \mathbb{R}^2$ with $v_1, v_2 \geq 0$, then there exists a basic rotation U such that $Uv = \|v\|e_1$. (b) For any $v \in \mathbb{C}^n$ with $\|v\| = 1$ there exist quantum gates $U_1, \ldots, U_{2n-1}$ such that $U_1 \cdots U_{2n-1}v = e_1$.

Proof. (a) If $v = 0$, then $U = 1$ will do, so suppose $v \neq 0$. If $v_1 = 0$, let $\theta = -\pi/2$ and otherwise let $\theta = \tan^{-1}(-v_2/v_1)$. Then $\sin \theta = -v_2/\|v\|$, $\cos \theta = v_1/\|v\|$, and letting $U = [1, 2, \theta]$, we have

$$Uv = \frac{1}{\|v\|}\begin{bmatrix} v_1 & v_2 \\ -v_2 & v_1 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \|v\|e_1$$

(b) If

$$v = (v_1, \ldots, v_n) = (|v_1|e^{i\theta_1}, \ldots, |v_n|e^{i\theta_n})$$

let $P_j = [j, j, -\theta_j], j = 1, \ldots, n$. Then $P_1 \cdots P_n v = (|v_1|, \ldots, |v_n|)$. Applying part (a), we have that there exist basic rotations $R_1, \ldots, R_{n-1}$ such that

$$R_j(|v_1|, \ldots, |v_n|) = (|v_1|, \ldots, |v_{j-1}|, (|v_j|^2 + |v_{j+1}|^2)^{1/2}, 0, |v_{j+2}|, \ldots, |v_n|)$$

$j = 1, \ldots, n - 1$. We then have

$$R_1 \cdots R_{n-1}P_1 \cdots P_n v$$

$$= R_1 \cdots R_{n-1}(|v_1|, \ldots, |v_n|)$$

$$= R_1 \cdots R_{n-2}(|v_1|, \ldots, |v_{n-2}|, (|v_{n-1}|^2 + |v_n|^2)^{1/2}, 0)$$
$$= R_1 \cdots R_{n-3}(|v_1|, \ldots, |v_{n-3}|, (|v_{n-2}|^2 + |v_{n-1}|^2 + |v_n|^2)^{1/2}, 0, 0)$$
$$\vdots$$
$$= ((|v_1|^2 + \cdots + |v_2|^2)^{1/2}, 0, \ldots, 0) = e_1 \quad \blacksquare$$

Theorem 5.3. If U is an $n \times n$ unitary matrix, there exist quantum gates $U_1, \ldots, U_m$ such that $U = U_1 \cdots U_m$.

Proof. A $n \times n$ unitary matrix M is *k-simple* if the first k rows and columns of M are the same as those of the n-dimensional identity matrix 1_n. In this case, we can write $m = 1_k \oplus A$ where A is an $(n - k) \times (n - k)$ unitary matrix. In particular, any unitary matrix is 0-simple and if M is n-simple, then $M = 1_n$. Notice that the product of two $n \times n$ k-simple matrices is also k-simple. Suppose that $U = 1_k \oplus A$ is k-simple, $0 \leq k < n$, and let A_1 be the first row of A. By Lemma 5.2(b), there exist $(n - k)$-dimensional quantum gates $V_1, \ldots, V_{2(n-k)-1}$ with product $V = V_1 \cdots V_{2(n-k)-1}$ such that $V A_1^* = \tilde{e}_1$, where $\tilde{e}_1$ is the first standard basis element of $\mathbb{C}^{n-k}$. Then $\hat{V} = 1_k \oplus V$ is k-simple and hence $W = U\hat{V}^*$ is also k-simple. We now show that W is $(k + 1)$-simple. Letting W_{k+1} be the $(k + 1)$th row of W, we must show that W_{k+1} is the standard basis element $e_{k+1} \in \mathbb{C}^n$. Now

$$W = U\hat{V}^* = (1_k \oplus A)(1_k \oplus V^*) = 1_k \oplus AV^*$$

Hence, letting 0_k be the zero vector in $\mathbb{C}^k$, we have

$$W_{k+1,j} = \langle We_j, e_{k+1}\rangle = \langle (1_k \oplus AV^*)e_j, e_{k+1}\rangle$$
$$= \langle e_j, (1_k \oplus VA^*)e_{k+1}\rangle = \langle e_j, (1_k \oplus VA^*)0_k \oplus \tilde{e}_1\rangle$$
$$= \langle e_j, 0_k \oplus VA^*\tilde{e}_1\rangle = \langle e_j, 0_k \oplus VA_1^*\rangle$$
$$= \langle e_j, 0_k \oplus \tilde{e}_1\rangle = \langle e_j, e_{k+1}\rangle = \delta_{j,k+1}$$

Thus, $W_{k+1} = e_{k+1}$.

We have shown that if U is k-simple, then $U = W_1\hat{V}_1$, where V_1 is a product of quantum gates and W_1 is $(k + 1)$-simple. Repeating this process, we have $W_1 = W_2\hat{V}_2$, where $\hat{V}_2$ is a product of quantum gates and W_2 is $(k + 2)$-simple. Hence, $U = W_2\hat{V}_2\hat{V}_1$ and eventually $U = W_{n-k}\hat{V}_{n-k} \cdots \hat{V}_1$ Since W_{n-k} is n-simple, we have $U = \hat{V}_{n-k} \cdots \hat{V}_1$. $\blacksquare$

Recall the pumping lemma for regular languages. If L is a regular language, then any sufficiently long word $w \in L$ can be written $w = xyz$ such that $xy^kz \in L$ for every $k \in \mathbb{N}$. The following lemma is a variation of a result in ref. 17.

Lemma 5.4 (Quantum pumping lemma). Let $\mathcal{A} = (H, s_0, I, U)$ be a q-automaton. For any $\varepsilon > 0$ and $w \in I^*$ there exists a $k \in \mathbb{N}$ such that

$$\left\|U(uw^k v) - U(uv)\right\| < \varepsilon \tag{5.3}$$

for all $u, v \in I^*$.

Proof. By Theorem 6 [17] there exists a $k \in \mathbb{N}$ such that $U(w)^k = 1 + \varepsilon J$, where $\|J\| < 1$. Hence,

$$\begin{aligned}\left\|U(uw^k v) - U(uv)\right\| &= \left\|U(u)U(w)^k U(v) - U(u)U(v)\right\| \\ &= \left\|U(u)[U(w)^k - 1]U(v)\right\| \\ &\leq \left\|U(w)^k - 1\right\| = \varepsilon\|J\| < \varepsilon \quad \blacksquare\end{aligned}$$

Corollary 5.5. Let $\mathcal{A} = (H, s_0, I, U, F)$ be a finalizing q-automaton. If $uv \in L(\mathcal{A}, \eta)$ and $w \in I^*$, then there exists a $k \in \mathbb{N}$ such that $u w^k v \in L(\mathcal{A}, \eta)$.

Proof. Since $\|P(F)U(uv)s_0\| > \eta$, there exists an $\varepsilon > 0$ such that

$$\|P(F)U(uv)s_0\| > \eta + \varepsilon$$

By Lemma 5.4, there exists a $k \in \mathbb{N}$ such that (5.3) holds. Hence,

$$\begin{aligned}\left\|P(F)U(uw^k v)s_0\right\| &= \left\|P(F)U(uv)s_0 + P(F)U(uw^k v)s_0 - P(F)U(uv)s_0\right\| \\ &\geq \|P(F)U(uv)s_0\| - \left\|P(F)U(uw^k v)\, s_0 - P(F)U(uv)s_0\right\| \\ &> \eta + \varepsilon - \left\|U(uw^k v) - U(uv)\right\| > \eta \quad \blacksquare\end{aligned}$$

Corollary 5.6. There are regular languages that are not η-quantum for any $0 \leq \eta < 1$.

Proof. By Corollary 5.5, no letter is forbidden in $L(\mathcal{A}, \eta)$ if $L(\mathcal{A}, \eta) \neq \{\lambda\}$. But there exist regular languages other than $\{\lambda\}$ with a forbidden letter. $\blacksquare$

For related results in a slightly different approach, we refer the reader to ref. 16.

6. OPEN PROBLEMS

The work in Sections 3–5 suggest many open problems. We now list the ones that we consider to be the most interesting and important.

If $\mathcal{A} = (H, s_0, I, U, F)$ is a finalizing q-automaton, the *probability function* $p_{\mathcal{A}}$: $I^* \to [0, 1]$ for $\mathcal{A}$ is defined by

$$p_{\mathcal{A}}(w) = \|P(F)U(w)s_0\|^2$$

We say that p: $I^* \to [0, 1]$ is *realizable* if $p = p_{\mathcal{A}}$ for some finalizing q-automaton $\mathcal{A}$.

Problem 1. Characterize the realizable functions p: $I^* \to [0, 1]$.

Problem 2. Characterize the finalizing q-automaton $\mathcal{A}$ that satisfies $p_{\mathcal{A}}(w) \in \{0, 1\}$ for every w; that is, $p_{\mathcal{A}}$ is a characteristic function.

The next set of problems deal with quantum languages.

Problem 3. Can the requirement that L_1 and L_2 have the same alphabet be removed in Corollary 4.4?

Problem 4. If L is a quantum language over the alphabet I, is $I^* \setminus L$ a quantum language?

Problem 5. If L_1 and L_2 are quantum languages over the same alphabet, is $L_1 \cup L_2$ a quantum language?

Problem 6. If the answer to Problem 5 is yes, can the same alphabet requirement be removed?

Problem 7. Is every generalized quantum language a quantum language?

Problem 8. If the answer to Problem 7 is no, answer Problems 3–6 for generalized quantum languages.

Problem 9. Is $L(\mathcal{A}; E) \cup L(\mathcal{A}; F)$ a quantum language? If not, what if $E \perp F$?

Problem 10. If L_1 and L_2 are η-quantum languages, is $L_1 \cap L_2$ an η-quantum language?

Problem 11. Answer Problems 3–9 for η-quantum languages.

Problem 12. Are 0-quantum, η-quantum for $0 < \eta < 1$, and quantum languages the same thing? If not, how do they compare?

Two finalizing q-automata $\mathcal{A}_1$, $\mathcal{A}_2$ over the same alphabet are *equivalent* if $L(\mathcal{A}_1) = L(\mathcal{A}_2)$ and η-*equivalent* if $L(\mathcal{A}_1, \eta) = L(\mathcal{A}_2, \eta)$.

Problem 13. Characterize the pairs $\mathcal{A}_1$, $\mathcal{A}_2$ that are equivalent, η-equivalent.

The next set of problems deal with the material of Section 5.

Problem 14. Does Corollary 5.5 hold with $L(\mathcal{A}, \eta)$ replaced by $L(\mathcal{A})$?

Problem 15. Are there regular languages that are not quantum languages? (See Problem 12.)

Problem 16. Are there quantum languages (η-quantum languages) that are not regular?

As the reader can see from this long list, the present paper opens more problems than it has solved. Thus there is much more interesting work to be done.

REFERENCES

1. L. Adelman, J. DeMarrais, and M. Huang, Quantum computability, *SIAM J. Comput.* **26**, 1524–1540 (1997).
2. A. Barenco, C. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev. A* **52**, 3457–3467 (1995).
3. P. Benioff, Quantum Hamiltonian models of Turing machines, *Int. J. Stat. Phys.* **29**, 515–546 (1982).
4. P. Benioff, Quantum mechanical Hamiltonian models of Turing machines that dissipate no energy, *Phys. Rev. Lett.* **48**, 1581–1585 (1982).
5. C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, Strengths and weaknesses of quantum computing, *SIAM J. Comput.* **26**, 1510–1523 (1997).
6. E. Bernstein and U. Vazirani, Quantum complexity theory, *SIAM J. Comput.* **26**, 1411–1473 (1997).
7. A. Berthiawme and G. Brassard, Oracle quantum computing, *J. Mod. Optics* **41**, 2521–2535 (1994).
8. J. Chuang, R. LaFlamme, P. Shor, and W. Zurek, Quantum computers, factoring and decoherence, *Science* **1995** (December 8), 1633–1635 (1995).
9. D. Deutsch, Quantum theory, the Church–Turing principle and the universal quantum computer, *Proc. R. Soc. Lond. A* **400**, 97–117 (1985).
10. D. Deutsch, Quantum computational networks, *Proc. R. Soc. Lond. A* **425**, 73–90 (1989).
11. D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computation, *Proc. R. Soc. Lond. A* **439**, 553–558 (1992).
12. D. DiVincenzo, Two-bit gates are universal for quantum computation, *Phys. Rev. A* **51**, 1015–1022 (1995).
13. R. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.* **21**, 467–488 (1982).
14. R. Feynman, Quantum mechanical computers, *Found. Phys.* **16**, 507–531 (1986).
15. A. Gleason, Measures on closed subspaces of a Hilbert space, *J. Rat. Mech. Anal.* **6**, 885–893 (1957).
16. A. Kondacs and J. Watrous, On the power of quantum finite state automata, in *Proceedings 38th Symposium on Foundations of Computer Science* (1997).
17. C. Moore and J. Crutchfield, Quantum automata and quantum grammars, *Theor. Comp. Sci.* (to appear).
18. G. Palma, K. Suominen, and A. Ekert, Quantum computers and dissipation, *Proc. R. Soc. Lond. A* **452**, 567–584 (1996).
19. A. Paz, *Introduction to Probabilistic Automata*, Academic Press, New York (1971).
20. P. Shor, Scheme for reducing decoherence in quantum computer memory, *Phys. Rev. A* **52**, 2493–2496 (1995).
21. P. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Comput.* **26**, 1484–1509 (1997).
22. D. Simon, On the power of quantum computation, *SIAM J. Comput.* **26**, 1474–1483 (1997).
23. A. Steane, Active stabilization, quantum computation, and quantum state synthesis, *Phys. Rev. Lett.* **78**, 2252–2255 (1997).