International Journal of Theoretical Physics, Vol. 44, No. 9, September 2005 (© 2005)
DOI: 10.1007/s10773-005-4781-z

Finite State and Finite Stop Quantum Languages

Ruqian Lu[1,2,3,4,5,7] and Hong Zheng[6]

Received January 10, 2005; accepted March 30, 2005

We propose the concept of finite stop quantum automata (ftqa) based on Hilbert space and compare it with the finite state quantum automata (fsqa) proposed by Moore and Crutchfield (*Theoretical Computer Science* **237**(1–2), 2000, 275–306). The languages accepted by fsqa form a proper subset of the languages accepted by ftqa. In addition, the fsqa form an infinite hierarchy of language inclusion with respect to the dimensionality of unitary matrices. We introduce complex-valued acceptance degrees and two types of finite stop quantum automata based on them: the invariant ftqa (icftq) and the variant ftqa (vcftq). The languages accepted by icftq form a proper subset of the languages accepted by vcftq. In addition, the icftq form an infinite hierarchy of language inclusion with respect to the dimensionality of unitary matrices. In this way, we establish two proper inclusion relations $\mathcal{L}$ (fsqa) $\subset$ $\mathcal{L}$ (ftqa) and $\mathcal{L}$ (icftq) $\subset$ $\mathcal{L}$ (vcftq), where the symbol $\mathcal{L}$ means languages, and two infinite language hierarchies $\mathcal{L}_n$ (fsqa) $\subset$ $\mathcal{L}_{n+1}$ (fsqa), $\mathcal{L}_n$ (icftq) $\subset$ $\mathcal{L}_{n+1}$ (icftq).

KEY WORDS: finite state quantum automata; finite stop quantum automata; complex valued quantum automata; quantum languages.

1. INTRODUCTION

Research on quantum computation not only stimulated people to find efficient quantum algorithms, but also to do some research on basic theories of quantum computation. It is well known that automata are simple theoretical models of computers. So quantum automata have become an active research area gradually.

[1] Shanghai Key Lab of Intelligent Information Processing, Fudan University, Shanghai 200433, P.R. China.
[2] Beijing Key Lab of Multimedia and Intelligent Software, Beijing University of Technology, Beijing 100022, P.R. China.
[3] Key Lab of IIP, Institute of Computing Technology, Academia Sinica, Beijing 100080, P.R. China.
[4] Key Lab of MADIS, Academia Sinica, Beijing 100080, P.R. China.
[5] Institute of Mathematics, AMSS, Academia Sinica, Beijing 100080, P.R. China.
[6] Patent Examination Cooperation Center, State Intellectual Property Office of the People's Republic of China, Xuezhixuan Building, 16 Xueqing Road, Haidian District, Beijing 100080, P.R. China.
[7] To whom correspondence should be addressed at Institute of Mathematics, Academy of Mathematics and System Sciences, Academia Sinica, Beijing 100080, P.R. China.

Moore and Crutchfield (2000) introduced several types of Quantum automata based on Hilbert space in their recent paper, in which the authors defined quantum finite state and quantum push-down automata (QFAs and QPDAs) as special cases of a more general object, a real-time quantum automaton. In this paper, first, we proposed a new type of finite state quantum automata (we call them finite stop quantum automata) based on Hilbert space and compared it with the QFAs proposed by Moore and Crutchfield (2000). Then we investigated the relations between the languages accepted by their automata and the languages accepted by ours. Later, the relations between the languages accepted by n-dimensional finite state quantum automata and $n+1$-dimensional finite state quantum automata have been studied. In this way, we gave a hierarchy of all finite state quantum automata. Finally, two new types of quantum automata with complex acceptance degrees have been defined and the relations between the languages accepted by them have been discussed.

2. FINITE STATE QUANTUM AUTOMATA

Definition 2.1. (Moore and Crutchfield, 2000). A real-time quantum automaton $\mathcal{R}$ can be represented with a six tuple $\mathcal{R} = (H, H_{\text{accept}}, S_{\text{init}}, \mathrm{P}_{\text{accept}}, \Sigma, \Delta)$, with

1. A Hilbert space H;
2. An initial vector $S_{\text{init}} \in H$, where $|S_{\text{init}}|^2 = 1$;
3. A sub-space $H_{\text{accept}} \subset H$ and an operator P_{accept} projecting elements of H into H_{accept};
4. A finite alphabet Σ of input symbols;
5. A finite set of unitary matrices $\Delta = \{U_x |$ for each $x \in \Sigma\}$;

   The quantum language $\mathcal{L}(\mathcal{R})$ recognized by $\mathcal{R}$ is defined as the following function:

$$f_{\mathcal{R}}(w) = |S_{\text{init}} U_w P_{\text{accept}}|^2 \tag{1}$$

   where $w = x_1 x_2, \ldots, x_n$ is an input string, $U_w = U_{x_1} U_{x_2}, \ldots, U_{x_n}$ is the matrix product of the input symbols. This function maps each string w in the interval [0, 1].

   In this paper, we also use the notation

$$\text{Accept}(\mathcal{R}, w) = |S_{\text{init}} U_w P_{\text{accept}}|^2 \tag{2}$$

   to replace (1).

We make the following remarks to the previous definition. First, a Hilbert space has in general infinitely many dimensions. Second, there is actually no definition of states. Third, the unitary matrix corresponding to each input symbol is unique and the application of it is thus history insensitive. Fourth, the item H_{accept} in the previous definition is determined by other factors, since if we let:

$$P' = \text{closure}\,\{v \mid v = S_{\text{init}} U_w P_{\text{accept}},\ \text{where } U_w \text{ is a finite product of matrices from } \Delta\}$$

where closure means the smallest linear sub-space of H containing all v, then there will be no problem to use P' as the sub-space H_{accept}. Due to this reason, the item H_{accept} will not appear in the definitions later.

Moore and Crutchfield (2000) have paid attention to the first point and have given a second definition about quantum automata:

Definition 2.2. (Moore and Crutchfield, 2000). A real-time quantum automaton $\mathcal{R}$ is called a finite state automaton if H, S_{init} and all U_x are finite dimensional and have the same dimension n.

To be of practical meaning, we consider the operator P_{accept} as an $n \times m$ matrix of complex numbers, where $1 \leq m \leq n$.

There is a significant difference between the notion "finite state automaton" used in this context and that in the classical sense. The notion "finite state" refers only to the finite number of basic states of a Hilbert Space. It is questionable if we count the number of all mixed states, of which each is a linear combination of the basic states. In general, there may be infinitely many mixed states produces by such an automaton. We try to make the concept of a mixed state explicit with the following definition.

Definition 2.3. A mixed state of a real-time quantum automaton $\mathcal{R}$ based on a finite (n)-dimensional Hilbert space, where H, S_{init} and all U_x are finite dimensional and have the same dimension n, is defined as follows:

1. S_{init} is a mixed state, called the initial mixed state;
2. If a is a mixed state, and U_x is an n-dimensional unitary matrix corresponding to the input symbol x, then aU_x is also a mixed state, which is reachable from a by accepting the input symbol x.

Theorem 2.1.

1. *All mixed states are normalized, i.e, if a is a mixed state, then $|a|^2 = 1$.*
2. *There are real-time quantum automata based on finite-dimensional Hilbert space, which have infinitely many mixed states.*

Proof: Let a be a mixed state, then a is produced by a product of the initial mixed state S_{init} with finitely many unitary matrices, i.e. $a = S_{\text{init}} U_w$, where U_w is the matrix product. Thus, □

$$a\overline{a'} = S_{\text{init}} U_w \overline{U'_w} \overline{S'_{\text{init}}} = S_{\text{init}} \overline{S'_{\text{init}}} = 1 \tag{3}$$

This shows that a is normalized.

Construct a real-time automaton $\mathcal{R}$ based on n-dimensional Hilbert space H as follows:

(1) $S_{\text{init}} = \frac{1}{\sqrt{n}}(1, 1, \ldots, 1)$
(2) $P_{\text{accept}} = \frac{1}{\sqrt{n}}(1, 1, \ldots, 1)'$
(3) $\Sigma = \{x\}$
(4) $U_x = \begin{pmatrix} e^{ic_1\pi} & & & \\ & e^{ic_2\pi} & & \\ & & \cdots & \\ & & & e^{ic_n\pi} \end{pmatrix}$, where all c_j, $1 \leq j \leq n$, are real numbers.

After receiving the input string x^m, the automaton enters in the mixed state:

$$s(x^m) = \frac{1}{\sqrt{n}}(e^{imc_1\pi}, e^{imc_2\pi}, \ldots, e^{imc_n\pi}) \tag{4}$$

We will check the condition when it is possible for the automaton to produce two identical mixed states, i.e. when it is possible that

$$s(x^m) = s(x^k) \quad \text{for } m \neq k$$

For that, it must be

$$\begin{aligned} e^{imc_1\pi} &= e^{ikc_1\pi}, \\ &\cdots \\ e^{imc_n\pi} &= e^{ikc_n\pi}. \end{aligned}$$

That means

$$\begin{aligned} mc_1 &= kc_1 + 2j_1 \\ &\cdots \\ mc_n &= kc_n + 2j_n \end{aligned}$$

where for all g, j_g are integers. Therefore, for $m \neq k$ we have the equation:

$$c_g = \frac{2j_g}{m-k}, \qquad 1 \leq g \leq n \tag{5}$$

This shows that in order to produce identical mixed states, all c_g must be rational numbers. Therefore, we need only to choose irrational numbers for the c_g to let the automaton always produce new mixed states. This proves that the automaton will produce infinitely many mixed states. □

Due to this fact, we will introduce a new concept of automata which can serve as a bridge between the classical finite state automata and the quantum finite state automata as they are defined in (Moore and Crutchfield, 2000). This new type of finite automata will be introduced and discussed in detail in the next section.

3. FINITE STOP QUANTUM AUTOMATA

Parallel to the finite state quantum automata discussed in previous section, we propose a definition of finite stop quantum automata as follows.

Definition 3.1. An n-dimensional finite stop quantum automaton $\mathcal{R}$ can be represented with an eight-tuple $\mathcal{R} = (H, S_{\text{init}}, S_{\text{term}}, \Sigma, Q, q_0, T, \Delta)$, with:

1. An n-dimensional Hilbert space H;
2. An initial vector $S_{\text{init}} \in H$, where $|S_{\text{init}}|^2 = 1$;
3. An operator S_{term} projecting elements of H into an m-dimensional subspace, $1 \leq m \leq n$. In practice, we consider it to be an $n \times m$ dimensional matrix of complex numbers with $|p|^2 \leq 1$, where p is a column vector of S_{term};
4. A finite set Σ of input symbols;
5. A finite set Q of stops, an initial stop $q_0 \in Q$ and a set $T \subseteq Q$ of terminal stops;
6. A (partial) mapping: $\Delta : Q \times \Sigma \times Q \to \text{UN}$,
   where UN is a set of unitary matrices. We write it as

$$\Delta = \{\delta(q_1, x, q_2)|\ \textit{if}\ \delta(q_1, x, q_2) \text{ is defined and equals to } U(q_1, x, q_2), \\ \text{which is unitary}\}$$

For each input string $w = x_1x_2, \ldots, x_m$, Accept $(\mathcal{R}, w) = a$ if and only if there is a chain of stops in $\mathcal{R} : q_0, q_1, \ldots, q_m$, such that $q_m \in T$, and for all $j, 0 \leq j \leq m-1, \delta(q_j, x, q_{j+1}) = U(q_j, x, q_{j+1}) \in \Delta$, and $|S_{\text{init}} U_w S_{\text{term}}|^2 = a$, where

$$U_w = \prod_{j=0}^{m-1} U(q_j, x, q_{j+1})$$

The quantum language $\mathcal{L}(\mathcal{R})$ recognized by $\mathcal{R}$ is defined as the following set of pairs:

$$\mathcal{L}(\mathcal{R}) = \{(s \in \Sigma^+,\ \text{Accept}\,(\mathcal{R}, s))|s \text{ is accepted by } \mathcal{R}\}$$

Compare this definition of finite stop quantum automata with those given in literature, e.g. those given in (Ambainis *et al.*, 1999) or (Kondacs and Watrous, 1997), we see there is a difference that in our definition the stops are not classified as accepting, rejecting and non-halting. Our definition of stops is more close to that in traditional finite state automata.

4. RELATIONS BETWEEN THE LANGUAGES ACCEPTED BY FINITE STATE QUANTUM AUTOMATA AND FINITE STOP QUANTUM AUTOMATA

Theorem 4.1. *For each finite state quantum automaton $\mathcal{R}$, there is a finite stop quantum automaton $\mathcal{R}'$, which accepts the same language accepted by $\mathcal{R}$.*

Proof: In fact, we can construct a "universal" finite stop quantum automaton $\mathcal{R}'$, which can simulate any finite state quantum automaton by modifying the partial mapping δ appropriately.

Assume a finite state quantum automaton $\mathcal{R} = (H, S_{\text{init}}, P_{\text{accept}}, \Sigma, \Delta = \{U_x | \forall x \in \Sigma\})$ as described in Definitions 2.1 and 2.2 is given. Construct the finite stop quantum automaton $\mathcal{R}' = (H, S'_{\text{init}}, S'_{\text{term}}, \Sigma', Q', q_0, T', \Delta')$ as follows: $S'_{\text{init}} = S_{\text{init}}$, $S'_{\text{term}} = P_{\text{accept}}$, $\Sigma' = \Sigma$, $Q' = \{q_0, q_1\}$, $T' = \{q_1\}$, $\Delta' = \{\delta'(q_0, x, q_1) = U_x, \delta'(q_1, x, q_1) = U_x | \forall x \in \Sigma\}$. The structure of $\mathcal{R}'$ is shown in Figure 1.

It is easy to see that for any input string w, which is accepted by $\mathcal{R}$ with $|S_{\text{init}} U_w P_{\text{accept}}|^2$, the automaton $\mathcal{R}'$ accepts the same w with $|S'_{\text{init}} U_w S'_{\text{term}}|^2$, and vice versa. □

Now we pose a question in another direction: Does there exist an equivalent finite state quantum automaton to any given finite stop quantum automaton? Let us first consider the case $n = 1$.

Lemma 4.1. *For one-dimensional Hilbert space, there is a finite stop quantum automaton $\mathcal{R}$, for which there does not exist any equivalent finite state quantum automaton $\mathcal{R}'$.*

Proof: In case of one-dimensional Hilbert space H_1, the initial state (or stop) S_{init}, the projecting state (or stop) S_{term}, and the unitary matrices

Fig. 1. "Universal" finite stop quantum automaton.

are all reduced to some complex number. Consider the one-dimensional finite stop quantum automaton $\mathcal{R} = (H_1, 1, 1, \{x\}, \{q_0, q_1, q_2\}, q_0, \{q_2\}, \Delta = \{\delta(q_0, x, q_1) = 1, \delta(q_1, x, q_2) = 1\})$. It is easy to see that

$$\text{Accept}(\mathcal{R}, x) = 0,\ \text{Accept}\,(\mathcal{R}, xx) = 1 \tag{6}$$

If there exists a finite state quantum automaton $\mathcal{R}'$, then $\mathcal{R}'$ should take the following form:

$$\begin{aligned} \mathcal{R}' &= (H_1, S'_{\text{init}}, P_{\text{accept}}, \{x\}, \Delta'), \\ \text{Accept}(\mathcal{R}', x) &= |S'_{\text{init}} U_x P_{\text{accept}}|^2 = |P_{\text{accept}}|^2 \\ \text{Accept}(\mathcal{R}', xx) &= |S'_{\text{init}} U_x U_x P_{\text{accept}}|^2 = |P_{\text{accept}}|^2 \end{aligned} \tag{7}$$

since S'_{init} and the unitary matrix U_x must be complex numbers with the absolute value 1. It is impossible to choose the value of P_{accept} such that both equations of (6) are satisfied. □

Now we will generalize the result obtained above to n-dimensional Hilbert spaces for arbitrary n.

Theorem 4.2. *For an n-dimensional Hilbert space of arbitrary positive integer n, there exists at least one finite stop quantum automaton, for which there is no equivalent n-dimensional finite state quantum automaton.*

We will not prove this theorem directly. Rather, we will prove a more general and more powerful theorem stated later.

Theorem 4.3. *For any n-dimensional Hilbert space of arbitrary positive integer n, there exists at least one n-dimensional finite stop quantum automaton $\mathcal{R}$, such that for any positive integer m and m-dimensional Hilbert space, there is no m-dimensional finite state quantum automaton $\mathcal{R}'$ equivalent to $\mathcal{R}$.*

Proof: Given $n > 0$, construct an n-dimensional finite stop quantum automaton $\mathcal{R}$ as follows: $S_{\text{init}} = \frac{1}{\sqrt{n}}(1, 1, \ldots, 1)$, $S_{\text{term}} = \frac{1}{\sqrt{n}}(1, 1, \ldots, 1)'$, $\Sigma = \{x\}$, $Q = \{q_0, q_1, \ldots, q_{f-1}, q_f\}$, q_0 is the initial stop, $T = \{q_f\}$, $f > 0$, $\forall j, 0 \le j \le f - 1$, $\delta(q_j, x, q_{j+1}) = U_I$, where U_I is the identity matrix of degree n, which is also unitary. The structure of this automaton is shown in Figure 2. □

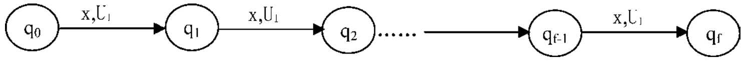

Fig. 2. An n-dimensional finite stop automaton.

It is easy to see that $\mathcal{R}$ does not accept any string other than x^f, which means

$$\begin{aligned}\text{Accept}(\mathcal{R}, x^f) &= \left|S_{\text{init}} U_I^f S_{\text{term}}\right|^2 = |S_{\text{init}} S_{\text{term}}|^2 = 1\\ \text{Accept}(\mathcal{R}, x^j) &= 0, \qquad j \neq f\end{aligned}$$

If there is an equivalent m-dimensional finite state quantum automaton $\mathcal{R}'$ for $m > 0$, then it should have the following form:

$$\begin{aligned}S'_{\text{init}} &= (s_1, s_2, \ldots, s_m),\\ P_{\text{accept}} &= \begin{pmatrix} p_{11} & \cdots & p_{1k}\\ \cdots & \cdots & \cdots\\ p_{m1} & \cdots & p_{mk}\end{pmatrix}, \quad m \geq k > 0,\\ \Sigma &= \{x\},\\ \Delta &= \{U_x\},\\ \left|S'_{\text{init}} U_x^f P_{\text{accept}}\right|^2 &= 1,\\ \left|S'_{\text{init}} U_x^j P_{\text{accept}}\right|^2 &= 0, \quad \text{for all} \quad j \neq f.\end{aligned}$$

According to the matrix theory, there must be a unitary matrix V, such that U_x can be represented in

$$U_x = V \Theta V^{-1},$$

where Θ is a diagonal matrix:

$$\Theta = \begin{pmatrix} e^{i\varphi_1} & & & \\ & e^{i\varphi_2} & & \\ & & \cdots & \\ & & & e^{i\varphi_m}\end{pmatrix},$$

where i is the unit imaginary number.

Θ is the eigenvalue matrix of U_x. Thus,

$$\begin{aligned}U_x^j &= V\Theta V^{-1} V\Theta V^{-1} \cdots V\Theta V^{-1}\\ &= V\Theta^j V^{-1}, \quad j > 0\\ S'_{\text{init}} U_x^j P_{\text{accept}} &= S'_{\text{init}} V \Theta^j V^{-1} P_{\text{accept}} = s_1 \Theta^j t_1\end{aligned}$$

where

$$s_1 = S'_{\text{init}} V, \qquad t_1 = V^{-1} P_{\text{accept}},$$

Let

$$s_1 = (s_{11}, s_{12}, \ldots, s_{1m}),$$

$$t_1 = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1k} \\ t_{21} & t_{22} & \cdots & t_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ t_{m1} & t_{m2} & \cdots & t_{mk} \end{pmatrix},$$

then

$$s_1 \Theta^j t_1 = \left(\sum_{h=1}^{m} s_{1h} t_{h1} e^{ji\varphi_h}, \ldots, \sum_{h=1}^{m} s_{1h} t_{hk} e^{ji\varphi_h} \right), \tag{8}$$

Note that i is the unit imaginary number.

In order to satisfy the condition $|s_1 \Theta^j t_1|^2 = 0$ for $j \neq f$, it should be:

$$\forall 1 \leq g \leq k, \quad \sum_{h=1}^{m} s_{1h} t_{hg} e^{ji\varphi_h} = 0. \tag{9}$$

On the other hand, in order to satisfy the condition $|s_1 \Theta^f t_1|^2 = 1$ for $j = f$, there should be:

$$\exists g, 1 \leq g \leq k, \quad \sum_{h=1}^{m} s_{1h} t_{hg} e^{ji\varphi_h} \neq 0. \tag{10}$$

Take this g (if there are more than one g, take any of them), for $j = 1, 2, 3, \ldots$ write down the linear equations as follows:

$$\begin{gathered}
s_{11} t_{1g} e^{i\varphi_1} + s_{12} t_{2g} e^{i\varphi_2} + \cdots + s_{1m} t_{mg} e^{i\varphi_m} = 0 \\
s_{11} t_{1g} e^{2i\varphi_1} + s_{12} t_{2g} e^{2i\varphi_2} + \cdots + s_{1m} t_{mg} e^{2i\varphi_m} = 0 \\
\cdots \\
s_{11} t_{1g} e^{(f-1)i\varphi_1} + s_{12} t_{2g} e^{(f-1)i\varphi_2} + \cdots + s_{1m} t_{mg} e^{(f-1)i\varphi_m} = 0 \\
s_{11} t_{1g} e^{fi\varphi_1} + s_{12} t_{2g} e^{fi\varphi_2} + \cdots + s_{1m} t_{mg} e^{fi\varphi_m} = r \neq 0 \\
s_{11} t_{1g} e^{(f+1)i\varphi_1} + s_{12} t_{2g} e^{(f+1)i\varphi_2} + \cdots + s_{1m} t_{mg} e^{(f+1)i\varphi_m} = 0 \\
\cdots \\
s_{11} t_{1g} e^{(f+p)i\varphi_1} + s_{12} t_{2g} e^{(f+p)i\varphi_2} + \cdots + s_{1m} t_{mg} e^{(f+p)i\varphi_m} = 0 \\
\cdots
\end{gathered} \tag{11}$$

To simplify the notation, let $s_{1h} t_{hg} = c_h$ for all h. Then we can consider the equation system (11) as one with the set of variables $(c_1, c_2, \ldots, c_m)$. So what we get is a system with infinitely many linear equations, where only one equation has a non-zero right side, all other equations have a zero right side.

Furthermore, we let $e^{i\varphi_h} = a_h$, then the coefficient matrix of the equation system (11) is a matrix with infinitely many rows. It has the following form:

$$C = \begin{pmatrix} a_1 & a_2 & \cdots & a_m \\ a_1^2 & a_2^2 & \cdots & a_m^2 \\ a_1^3 & a_2^3 & \cdots & a_m^3 \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix} \tag{12}$$

Construct the determinate of a finite sub-matrix C_1 of C:

$$\det C_1 = \det \begin{pmatrix} a_1 & a_2 & \cdots & a_m \\ a_1^2 & a_2^2 & \cdots & a_m^2 \\ \cdots & \cdots & \cdots & \cdots \\ a_1^m & a_2^m & \cdots & a_m^m \end{pmatrix} = \prod_{j=1}^{m} a_j \det C_1' \tag{13}$$

where

$$C_1' = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ a_1 & a_2 & \cdots & a_m \\ \cdots & \cdots & \cdots & \cdots \\ a_1^{m-1} & a_2^{m-1} & \cdots & a_m^{m-1} \end{pmatrix} \tag{14}$$

We are familiar with this matrix. Its determinate is the so-called Van de Monde determinate. According to a classical result:

$$\det C_1' = \prod_{m \geq u > v \geq 1} (a_u - a_v) \tag{15}$$

$$\det C_1 = \prod_{j=1}^{m} a_j \prod_{m \geq u > v \geq 1} (a_u - a_v) \tag{16}$$

Therefore, det $C_1 = 0$ if and only if $a_u = a_v$ for at least one pair of (a_u, a_v) (We know already that $a_u \neq 0$ for all u). That means, the rank of C_1 is less than m if and only if $a_u = a_v$ for at least one pair of (a_u, a_v). It follows that the infinite matrix C has a rank less than m if and only if $a_u = a_v$ for at least one pair of (a_u, a_v).

In order to study the case where $a_u = a_v$ for more than one pair of (a_u, a_v), and to determine the exact rank of C in this case, we assume that the m numbers $a_1, a_2, \ldots, a_m$ are divided in q groups such that the members in the same group are equal to each other, and the elements in different groups are mutually different.

Then we take only one element from each group as a representative. Without loss of generality we may assume these representatives are the first q elements: $a_1, a_2, \ldots, a_q$, otherwise we can achieve this by rearranging the columns of C, since such a rearrangement would not modify its rank. Thus, we have:

$$C = \left(\begin{bmatrix} \text{the-first-q-} \\ \text{mutual-different-} \\ \text{columns} \end{bmatrix} \begin{bmatrix} \text{the-last-}(m-q)\text{-} \\ \text{columns-which-are-} \\ \text{copies-of-the-first-}q \end{bmatrix} \right) = (C_3\, C_4) \tag{17}$$

Based on the result about Van de Monde determinate mentioned earlier and on the basic theory of matrices, it follows then

$$\text{rank of } C = \text{rank of } C_3 = q \tag{18}$$

That means, the rank of C is equal to the maximal number of its mutual different columns. And each sub-matrix of C with degree $q' \times m$, where $m \geq q' \geq q$, has also a rank of q.

With these preparations we are now ready for further discussing the equation system (11). Starting from the $f+1$th equation, we construct a finite system of m equations as follows:

$$\begin{aligned} c_1 e^{(f+1)i\varphi_1} + c_2 e^{(f+1)i\varphi_2} + \cdots + c_m e^{(f+1)i\varphi_m} &= 0 \\ c_1 e^{(f+2)i\varphi_1} + c_2 e^{(f+2)i\varphi_2} + \cdots + c_m e^{(f+2)i\varphi_m} &= 0 \\ \cdots \\ c_1 e^{(f+m)i\varphi_1} + c_2 e^{(f+m)i\varphi_2} + \cdots + c_m e^{(f+m)i\varphi_m} &= 0 \end{aligned} \tag{19}$$

Then we can consider (19) as one linear equation system with the set of variables $(c_1, c_2, \ldots, c_m)$ and the coefficient matrix of the equation system (19) is as follows:

$$C'' = \begin{pmatrix} e^{(f+1)i\varphi_1} & e^{(f+1)i\varphi_2} & \ldots & e^{(f+1)i\varphi_m} \\ e^{(f+2)i\varphi_1} & e^{(f+2)i\varphi_2} & \ldots & e^{(f+2)i\varphi_m} \\ \ldots & \ldots & \ldots & \ldots \\ e^{(f+m)i\varphi_1} & e^{(f+m)i\varphi_2} & \ldots & e^{(f+m)i\varphi_m} \end{pmatrix}$$

$$\det C'' = \prod_{j=1}^{m} e^{(f+1)i\varphi_j} \cdot \det C'''$$

where

$$C''' = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ e^{i\varphi_1} & e^{i\varphi_2} & \ldots & e^{i\varphi_m} \\ \ldots & \ldots & \ldots & \ldots \\ e^{(m-1)i\varphi_1} & e^{(m-1)i\varphi_2} & \ldots & e^{(m-1)i\varphi_m} \end{pmatrix} \tag{20}$$

Let $e^{i\varphi_j}$ be a_j, it is easy to see that the determinate of (20) is a Van de Monde determinate. We differentiate two cases.

Case 1. The rank of C is m, then the rank of the coefficient matrix of (19) is also m. It follows then that (19) has only trivial solutions (all variables have the value zero). That means, for all h, $c_h = 0$. Then we must have:

$$c_1 e^{fi\varphi_1} + c_2 e^{fi\varphi_2} + \cdots + c_m e^{fi\varphi_m} = 0$$

which contradicts (11).

Case 2. The rank of C is $q < m$. Consider the equation system consisting of m equations starting from the fth one until the $(f+m-1)$th one:

$$\begin{aligned} c_1 e^{fi\varphi_1} + c_2 e^{fi\varphi_2} + \cdots + c_m e^{fi\varphi_m} &= r \neq 0 \\ c_1 e^{(f+1)i\varphi_1} + c_2 e^{(f+1)i\varphi_2} + \cdots + c_m e^{(f+1)i\varphi_m} &= 0 \\ \cdots \\ c_1 e^{(f+m-1)i\varphi_1} + c_2 e^{(f+m-1)i\varphi_2} + \cdots + c_m e^{(f+m-1)i\varphi_m} &= 0 \end{aligned} \tag{21}$$

From discussions made earlier we can conclude that the rank q of C is less than m, if and only if the maximal number of different columns in C is $q < m$. Furthermore, it is equal to q if and only if the maximal number of different columns of the coefficient matrix of (21) is equal to q. Furthermore, it is equal to q if and only if the rank of this coefficient matrix (which is of type of C'' discussed earlier) is q.

On the other hand, we consider the sub-matrix C'''' formed by deleting the first row from the coefficient matrix. (Obviously, C'''' is of type of C_1. Note also $m - 1 \geq q$.) Since $r \neq 0$ we know that the extended matrix of (21), obtained by adding the right side constants of (21) to the coefficient matrix, has the rank $q + 1$. From the theory of linear equation systems we know that it is impossible that the system (21) has a solution.

In summary, we have proved that the m-dimensional unitary matrices needed for constructing an equivalent finite state quantum automaton do not exist. The theorem is thus proved. □

Note that what we have proved is actually more than it was said in the statement of Theorem 4.2. We have actually proved the following theorem.

Theorem 4.4. *If it is additionally prescribed that to each input symbol x there corresponds only one unitary matrix U_x (independent of the stop the automaton is in), then the conclusion of Theorem 4.2 is still valid.*

5. THE HIERARCHY OF FINITE STATE QUANTUM LANGUAGES

Definition 5.1. The class of languages accepted by finite state quantum automata is called finite state quantum languages. The class of languages accepted by n-dimensional finite state quantum automata is called n-dimensional finite state quantum languages.

Theorem 5.1. *For $0 < m < n$, the m-dimensional finite state quantum languages form a subclass of n-dimensional finite state quantum languages.*

Proof: Assume a finite state quantum automaton $\mathcal{R}$ is given by:

(1) $S_{\text{init}} = (a_1, a_2, \ldots, a_m)$,

(2) $P_{\text{accept}} = \begin{pmatrix} b_{11} & \cdots & b_{1k} \\ \cdots & \cdots & \cdots \\ b_{m1} & \cdots & b_{mk} \end{pmatrix} \qquad 1 \le k \le m$

(3) $\Sigma = \{x_j | 1 \le j \le g\}$,

(4) $\Delta = (U_x | x \in \Sigma)$

Construct a finite state quantum automaton $\mathcal{R}'$ as follows:

$$S'_{\text{init}} = ([S_{\text{init}}][n - m \ \ \text{zeros}]),$$

$$P'_{\text{accept}} = \begin{pmatrix} \lfloor P_{\text{accept}} \rfloor \\ \begin{bmatrix} (n-m)\text{-rows} \\ \text{of-zero-elements} \end{bmatrix} \end{pmatrix}$$

$$\Sigma' = \Sigma,$$

$$\Delta' = \left\{ U'_x = \begin{pmatrix} U_x & \begin{bmatrix} \text{zero-} \\ \text{elements} \end{bmatrix} \\ \begin{bmatrix} \text{zero-} \\ \text{elements} \end{bmatrix} & \begin{bmatrix} (n-m)\text{-dimensional-} \\ \text{identity-matrix} \end{bmatrix} \end{pmatrix} |x \in \Sigma' \right\}$$

Then for an arbitrary input string $w = x_1 x_2 \ldots x_k$ we have

$$\text{Accept}\ (\mathcal{R}', w) = |S'_{\text{init}} U'_{x_1} U'_{x_2}, \ldots, U'_{x_k} P'_{\text{accept}}|^2$$

$$= \left| (S_{\text{init}}, 0, \ \cdots, \ 0) \begin{pmatrix} U_{x_1} & & 0 & \\ & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 \\ & 0 & 0 & 1 \end{pmatrix} \cdots \begin{pmatrix} U_{x_k} & 0 & & \\ & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 \\ & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_{\text{accept}} \\ 0(n-m, k) \end{pmatrix} \right|^2$$

$$= \left| (S_{\text{init}}, 0, \cdots, 0) \begin{pmatrix} U_{x_1} U_{x_2} & \cdots & U_{x_k} & 0 \\ & & \ddots & \\ & 0 & & 1 \end{pmatrix} \begin{pmatrix} P_{\text{accept}} \\ 0(n-m,k) \end{pmatrix} \right|^2$$

$$= \left| (S_{\text{init}} U_{x_1} U_{x_2} \ldots U_{x_k} P_{\text{accept}}, 0, \cdots, 0 \right|^2$$

$$= \text{Accept}(\mathcal{R}, w)$$

where $0(n-m,k)$ denotes a matrix of degree $(n-m) \times k$ with zero elements only. □

Lemma 5.1. *The one-dimensional finite state quantum languages form a proper subset of the two-dimensional finite state quantum languages.*

Proof: Remember that in case of one-dimensional Hilbert space, the unitary matrices are unit complex numbers in form of $e^{i\varphi}$. The initial vector S_{init} is also a unit complex number in form of $e^{i\psi}$. Then

$$\text{Accept}(\mathcal{R}, w) = |S_{\text{init}} e^{\varphi_1} e^{\varphi_2} \ldots e^{\varphi_n} P_{\text{accept}}|^2 = |P_{\text{accept}}|^2$$

This is evidence that the acceptance degree is the same for all input strings. In another word, these automata do not differentiate between different input strings.

Now we construct a two-dimensional finite state quantum automaton $\mathcal{R}'$ with

$$\begin{aligned} S_{\text{init}} &= \tfrac{1}{\sqrt{2}}(1, 1), \\ P_{\text{accept}} &= \tfrac{1}{\sqrt{2}}(1, 1)' \\ \Sigma &= \{x, y\}, \\ \Delta &= \left\{ U_x = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; U_y = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \right\} \end{aligned}$$

then

$$\begin{aligned} \text{Accept}(\mathcal{R}, x) &= 1, \\ \text{Accept}(\mathcal{R}, y) &= 0 \end{aligned}$$

Obviously, there is no one-dimensional finite state quantum automaton, which can have this behavior. □

We will now generalize this result to the case of arbitrary n-dimensional finite state quantum automaton.

Definition 5.2. The language $\mathcal{L}(\mathcal{R})$ accepted by a finite state quantum automaton $\mathcal{R}$ is said to possess the n-vector property, if

$$\{(x, 0), (x^2, 0), \ldots, (x^{n-1}, 0), (x^n, 1)\} \subseteq \mathcal{L}(\mathcal{R}) \tag{22}$$

holds, where $x \in \Sigma$ is some input symbol of $\mathcal{R}$. In words, possessing the n-vector property means that $\mathcal{R}$ rejects all strings x^j, $1 \leq j \leq n-1$, but accepts x^n.

Theorem 5.2. *For each $n > 0$, there exists an n-dimensional finite state quantum automaton, whose language possesses the n-vector property. On the other hand, there is no $n-1$-dimensional finite state quantum automaton, whose language possesses the n-vector property.*

Proof: We go ahead along the similar idea of the proof of Theorem 4.3. In order to determine the structure of the n-dimensional automaton $\mathcal{R}$, we list now the equations, which must be satisfied by $\mathcal{R}$. They are

$$\begin{aligned} |S_{\text{init}} U_x P_{\text{accept}}|^2 &= 0, \\ \left|S_{\text{init}} U_x^2 P_{\text{accept}}\right|^2 &= 0, \\ &\cdots \\ \left|S_{\text{init}} U_x^{n-1} P_{\text{accept}}\right|^2 &= 0, \\ \left|S_{\text{init}} U_x^{n} P_{\text{accept}}\right|^2 &= 1, \end{aligned} \tag{23}$$

where S_{init} is an n-dimensional normalized vector, U_x is n-dimensional unitary matrix, P_{accept} is a matrix of degree $n \times k$, $1 \leq k \leq n$.

With the technique used in the proof of theorem 4.3, we obtain a system of n linear equations:

$$\begin{aligned} s_{11}t_{1g}e^{i\varphi_1} + s_{12}t_{2g}e^{i\varphi_2} + \cdots + s_{1n}t_{ng}e^{i\varphi_n} &= 0 \\ s_{11}t_{1g}e^{2i\varphi_1} + s_{12}t_{2g}e^{2i\varphi_2} + \cdots + s_{1n}t_{ng}e^{2i\varphi_n} &= 0 \\ &\cdots \\ s_{11}t_{1g}e^{(n-1)i\varphi_1} + s_{12}t_{2g}e^{(n-1)i\varphi_2} + \cdots + s_{1n}t_{ng}e^{(n-1)i\varphi_n} &= 0 \\ s_{11}t_{1g}e^{ni\varphi_1} + s_{12}t_{2g}e^{ni\varphi_2} + \cdots + s_{1n}t_{ng}e^{ni\varphi_n} &= r \neq 0 \end{aligned} \tag{24}$$

where g is some fixed number between 1 and n. For the meanings of s_{1j}, t_{jg} and $e^{ji\varphi_k}$, $1 \leq j \leq n$, $1 \leq k \leq n$, please refer to the proof of Theorem 4.3. The equation system (24) looks somehow similar to equation system (11), but (24) is a system with only finitely many equations, while (11) is an infinite system.

$$\text{Let } s_{1j}t_{jg} = c_j, \quad 1 \leq j \leq n.$$

If we can find a solution for the c_j, then we have also a solution for the s_{1j} and the t_{jg}, and thus a solution for the wanted n-dimensional automaton $\mathcal{R}$. For finding the solution, we first have to determine the coefficient matrix. Here we see that we have some degree of freedom. Namely, we can assign arbitrary values to the φ_j in the way that for $j \neq k$, $\varphi_j \neq \varphi_k$. From the theory of Van de Monde determinate we know that the rank of (24) is equal to n. Therefore, system (24) has a unique solution for the c_j, where the c_j are complex numbers. Let:

$$s_{1j} = \frac{1}{\sqrt{n}} \qquad t_{jg} = \sqrt{n} c_j \tag{25}$$

we get an n-dimensional finite state quantum automaton $\mathcal{R}$ with the required n-vector property.

On the other hand, if there were an $n-1$-dimensional finite state quantum automaton $\mathcal{R}'$, whose language possesses the n-vector property, then $\mathcal{R}'$ must satisfy the following equations:

$$\begin{aligned} |S'_{\text{init}} U'_x P'_{\text{accept}}|^2 &= 0, \\ |S'_{\text{init}} U'^2_x P'_{\text{accept}}|^2 &= 0, \\ &\cdots \\ |S'_{\text{init}} U'^{n-1}_x P'_{\text{accept}}|^2 &= 0, \\ |S'_{\text{init}} U'^n_x P'_{\text{accept}}|^2 &= 1, \end{aligned} \tag{26}$$

where S'_{init} is an $n-1$-dimensional vector, U'_x is an $n-1$-dimensional unitary matrix, P'_{accept} is a matrix of degree $(n-1) \times k$, $1 \leq k \leq n-1$.

By making use of the technique of proving Theorem 4.3, we get the following system of equations:

$$\begin{aligned} s_{11}t_{1g}e^{i\varphi_1} + s_{12}t_{2g}e^{i\varphi_2} + \cdots + s_{1,(n-1)}t_{(n-1),g}e^{i\varphi_{n-1}} &= 0 \\ s_{11}t_{1g}e^{2i\varphi_1} + s_{12}t_{2g}e^{2i\varphi_2} + \cdots + s_{1,(n-1)}t_{(n-1),g}e^{2i\varphi_{n-1}} &= 0 \\ &\cdots \\ s_{11}t_{1g}e^{(n-1)i\varphi_1} + s_{12}t_{2g}e^{(n-1)i\varphi_2} + \cdots + s_{1,(n-1)}t_{(n-1),g}e^{(n-1)i\varphi_{n-1}} &= 0 \\ s_{11}t_{1g}e^{ni\varphi_1} + s_{12}t_{2g}e^{ni\varphi_2} + \cdots + s_{1,(n-1)}t_{(n-1),g}e^{ni\varphi_{n-1}} &= r' \neq 0 \end{aligned} \tag{27}$$

The equation system (27) looks very much like equation system (24). But they are different. The former has only $n-1$ additive elements on the left side of each of its equations, while the latter has n such elements. For fixed coefficients $e^{ji\varphi_k}$, equation system (27) has $n-1$ variables in form of $s_{1j}t_{jg}$, whose values are to be determined, while equation system (24) has n variables. This difference is important, which leads to completely different results.

Considering the first $n-1$ equations of (27), we differentiate two cases.

Case 1. The rank of its coefficient matrix is $n-1$, then there is only a trivial solution: $\forall j$, $s_{1j}t_{jg} = 0$. This contradicts the nth equation of (27).

Case 2. The rank of its coefficient matrix is $q < n - 1$, then consider a new equation system, which consists of the last $n - 1$ equations of (27). Based on a similar idea of reasoning as in the proof of Theorem 4.3, the coefficient matrix of this equation system is q, and the rank of the extended coefficient matrix (with the constants on the right sides additionally) is $q + 1$. Thus, this equation system is inconsistent and has no solutions.

This proves that an $n - 1$-dimensional finite state quantum automaton satisfying the requirements (26) cannot exist. □

Corollary 5.1. *For each $n > 0$, the n-dimensional finite state quantum languages form a proper subset of the $n + 1$-dimensional finite state quantum languages.*

The hierarchy of quantum language inclusion is shown in Figure 3.

6. QUANTUM AUTOMATA WITH COMPLEX ACCEPTANCE DEGREES

Many physical quantities are defined or represented in complex values. For example, in electrodynamics, the phase of an electrical current is represented with a complex value. In our case, we have to note that the Hilbert space itself is a complex linear space, on which scalar products can be defined. Therefore, it should be meaningful to define complex valued acceptance degrees for quantum automata on Hilbert Space.

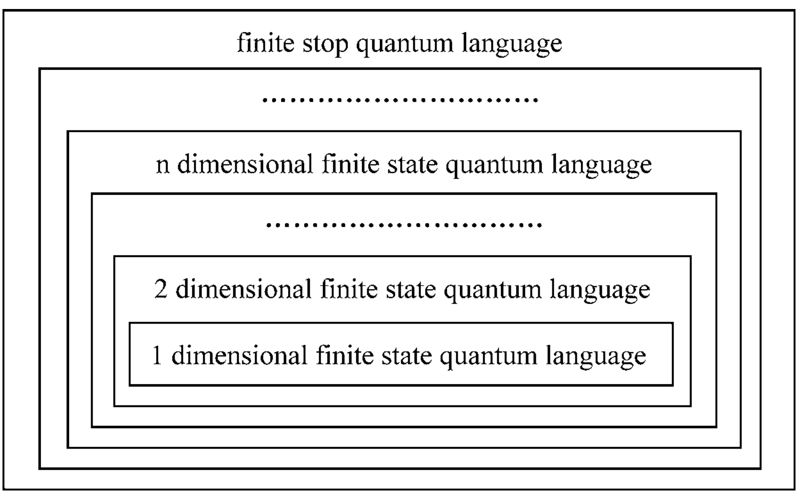

Fig. 3. The hierarchy of quantum language inclusion.

Definition 6.1. A complex valued finite state quantum automaton defined on an n-dimensional Hilbert space, called n-dimensional cfsq, or simply cfsq, is represented with a quintuple $\mathcal{R} = (H, S_{\text{init}}, P_{\text{accept}}, \Sigma, \Delta)$, with the following:

1. An n-dimensional Hilbert space H.
2. An initial vector $S_{\text{init}} \in H$, where $|S_{\text{init}}|^2 = 1$.
3. An operator P_{accept}, which is a complex matrix of order $n \times m$, $1 \leq m \leq n$, projecting elements of H into a sub-space of H.
   For any vector a of H,
$$P_{\text{accept}} : a \to a P_{\text{accept}}$$
   Besides, each column p of P_{accept} is subject to the limitation $|p|^2 \leq 1$;
4. A finite input alphabet Σ.
5. For each symbol $x \in \Sigma$, there corresponds an n-dimensional unitary matrix $U_x \in \Delta$.
6. Let $U_w = U_{x_1} U_{x_2} \ldots U_{x_n}$, where $w = x_1 x_2 \ldots x_n$, $x_j \in \Sigma$, $1 \leq j \leq n$
   The acceptance degree of w by $\mathcal{R}$ is defined as:
$$\text{Accept}(\mathcal{R}, w) = S_{\text{init}} U_w P_{\text{accept}},$$
   which is an m-dimensional complex vector.
7. The quantum language recognized by $\mathcal{R}$ is defined as the set of pairs $(w, f_R(w))$, where $w \in \Sigma^*$, $f_{\mathcal{R}}(w) = \text{Accept}(\mathcal{R}, w)$.

It follows from this definition that almost every thing of the definition of finite state quantum automata is kept unchanged except the acceptance degree, which can take the value of a complex vector.

Accordingly, we give the definition of complex valued finite stop quantum automata on Hilbert spaces.

Definition 6.2. A complex valued finite stop quantum automaton defined on an n-dimensional Hilbert space, called n-dimensional cftq, or simply cftq, is represented as an eight-tuple $\mathcal{R} = (H, S_{\text{init}}, S_{\text{term}}, \Sigma, Q, I, T, \Delta)$, with the following:

1. An n-dimensional Hilbert space H.
2. An initial vector $S_{\text{init}} \in H$, where $|S_{\text{init}}|^2 = 1$.
3. An operator S_{term}, which is a complex matrix of order $n \times m$, $1 \leq m \leq n$, projecting elements of H into an m-dimensional sub-space of H.
   For any vector a of H,
$$S_{\text{term}} : a \to a S_{\text{term}}$$
   Besides, each column p of S_{term} is subject to the limitation $|p|^2 \leq 1$.
4. A finite input alphabet Σ.

5. A finite set Q of stops, with $I \subset Q$ the set of initial stops, $I = \{q_0\}$, $T \subset Q$ the set of terminal stops.
6. A set Δ of transition functions,

$$\Delta = \{\delta(q_1, x, q_2) = U(q_1, x, q_2) | q_1, q_2 \in Q, x \in, U \text{ is an } n\text{-dimensional unitary matrix}\},$$

where $\delta(q_1, x, q_2)$ is a partial mapping from $Q \times \Sigma \times Q$ to the set of n-dimensional unitary matrices. A triple (q_1, x, q_2) is called a transition.

Thus, we have the following definition of the quantum language:

1. A string $s = x_1, x_2, \ldots, x_n, x_j \in \Sigma, 1 \leq j \leq n$, is said to be accepted by a cftq $\mathcal{R}$ if there is a chain of transitions $w = (q_0, x_1, q_1), (q_1, x_2, q_2), \ldots, (q_{n-1}, x_n, q_n)$, where $q_0 \in I, q_n \in T$.
2. The acceptance degree of string s by $\mathcal{R}$ is defined as:
   Accept $(\mathcal{R}, s) = \{S_{\text{init}} U_w S_{\text{term}} | w$ is a chain of transitions (henceforth called a path accepting s) of $\mathcal{R}$ and
$$U_w = U(q_0, x_1, q_1) U(q_1, x_2, q_2) \cdots U(q_{n-1}, x_n, q_n)\}.$$
3. The quantum language recognized by $\mathcal{R}$ is defined as the set of pairs $\{(s, \text{ Accept}\,(\mathcal{R}, s)) | s \in \Sigma^+, s$ accepted by $\mathcal{R}\}$.

Definition 6.3. Let $\mathcal{R}$ be a cftq. $\mathcal{R}$ is called deterministic if for each stop q_1 and input symbol $x \in \Sigma$, there is at most one stop q_2 and one unitary matrix $U(q_1, x, q_2)$, such that the mapping

$$\delta(q_1, x, q_2) \to U(q_1, x, q_2)$$

is well defined.

It is easy to see that all theorems (from Theorem 2.1 to Corollary 5.1) given earlier remain valid for complex valued finite state and finite stop automata after a slight modification of the proof procedures. Here we limit the projection operator $\mathrm{P}_{\text{accept}}$ respectively S_{term} to be an $n \times 1$ complex vector, i.e. we limit the projection sub-space to be one dimensional, since once the theorem is proved for one-dimensional sub-space, then its validity is obvious for any m-dimensional sub-space with $1 \leq m \leq n$.

Definition 6.4. A deterministic cftq is called invariant if for each input symbol $x \in \Sigma$, there is at most one unitary matrix U, such that for any stops q_1 and q_2, whenever the transition function $\delta(q_1, x, q_2)$ is defined, its value is always $U(q_1, x, q_2)$. Otherwise, it is called variant (one input symbol may correspond to several unitary matrices). In the sequel, we denote the former with icftq, and the latter with vcftq.

It is obvious that the set of all deterministic cftq forms a subset of the set of all cftq. But the following result is not obvious.

Theorem 6.1. *The set of all icftq forms a proper subset of the set of all vcftq.*

Proof: We have only to construct a vcftq $\mathcal{R}$ whose language $\mathcal{L}(\mathcal{R})$ cannot be accepted by any icftq. Let

$$S_{\text{init}} = \frac{1}{\sqrt{n}}(1, 1, \ldots, 1)$$

$$S_{\text{term}} = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

In case of $n = 1$, every unitary matrix is a complex number of the form $e^{i\varphi}$. Let $S_{\text{init}} = S_{\text{term}} = 1$ for $\mathcal{R}$. The other components of $\mathcal{R}$ are as follows: $\Sigma = \{x\}$, $Q = \{q_0, q_1, q_2\}$, $I = \{q_0\}$, $T = \{q_1, q_2\}$, $\Delta = \{\delta(q_0, x, q_1) = e^{i\varphi_1}, \delta(q_1, x, q_2) = e^{i\varphi_2}, \delta(q_2, x, q_2) = e^{i\varphi_3}\}$. Therefore,

$$\text{Accept}(\mathcal{R}, x) = e^{i\varphi_1}$$

$$\text{Accept}(\mathcal{R}, xx) = e^{i(\varphi_1+\varphi_2)}$$

$$\text{Accept}(\mathcal{R}, xxx) = e^{i(\varphi_1+\varphi_2+\varphi_3)}$$

If there is an equivalent icftq $\mathcal{R}'$, then there must be a complex number $e^{i\varphi}$, such that the following equations hold:

$$\text{Accept}(\mathcal{R}, x) = S_{\text{init}} e^{i\varphi} S_{\text{term}} = e^{i\varphi_1}$$

$$\text{Accept}(\mathcal{R}, xx) = S_{\text{init}} e^{2i\varphi} S_{\text{term}} = e^{i(\varphi_1+\varphi_2)}$$

$$\text{Accept}(\mathcal{R}, xxx) = S_{\text{init}} e^{3i\varphi} S_{\text{term}} e^{i(\varphi_1+\varphi_2+\varphi_3)}$$

According to definition, $S_{\text{init}} = e^{i\varphi_4}$ for some φ_4, $S_{\text{term}} = re^{i\varphi_5}$ for some φ_5 and $0 \leq r \leq 1$. The case $r < 1$ is impossible because of the values of the right sides. That means, the equations are equal to the following:

$$\varphi_4 + \varphi + \varphi_5 = \varphi_1$$

$$\varphi_4 + 2\varphi + \varphi_5 = \varphi_1 + \varphi_2$$

$$\varphi_4 + 3\varphi + \varphi_5 = \varphi_1 + \varphi_2 + \varphi_3$$

They are valid in the sense of $\bmod(2\pi)$. We can remove this limitation by making the $\varphi_1, \varphi_2, \varphi_3$ small enough. By subtracting the first equation from the second one, we get $\varphi = \varphi_2$. By subtracting the second equation from the third, we get $\varphi = \varphi_3$. This offers a contradiction if we let $\varphi_2 \neq \varphi_3$.

That means in case of $n = 1$ it is impossible that an equivalent icftq $\mathcal{R}'$ exits.

Now we consider the cases $n > 1$. First we construct a vcftq $\mathcal{R}$ as follows:

$$S_{\text{init}} = \frac{1}{\sqrt{n}}(1, 1, \ldots, 1)$$

$$S_{\text{term}} = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

$$\Sigma = \{x\}$$

$$Q = \{q_0, q_1, q_2\}, \quad I = \{q_0\}, \quad T = \{q_1, q_2\}$$

$$\Delta = \{\delta(q_0, x, q_1) = U_1, \delta(q_1, x, q_2) = U_1^{-1}, \delta(q_2, x, q_1) = U_1\}$$

In case of $n = 2$:

$$S_{\text{init}} = \frac{1}{\sqrt{2}}(1, 1)$$

$$S_{\text{term}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Let the unitary matrix U_1 be:

$$U_1 = \begin{bmatrix} e^{i\varphi} & 0 \\ 0 & e^{-i\varphi} \end{bmatrix}$$

where $\varphi = \arccos 1/2$, thus

$$\text{Accept}(\mathcal{R}, x) = S_{\text{init}} U_1 S_{\text{term}} = \frac{1}{2}\,(1, 1) \begin{bmatrix} e^{i\varphi} & 0 \\ 0 & e^{-i\varphi} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{2}$$

$$\text{Accept}(\mathcal{R}, xx) = 1$$

For $n > 2$, let the unitary matrix U_1 be the following diagonal matrix:

$$U_1 = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & e^{2\pi i/(n-1)} & & & \\ & & & e^{4\pi i/(n-1)} & & \\ & & & & \cdots & \\ & & & & & e^{2(n-2)\pi i/(n-1)} \end{bmatrix}$$

Obviously, $\mathcal{R}$ does not accept any input string other than x^f (f is an arbitrary positive integer). For x^f, $\mathcal{R}$ accepts it with the acceptance degree:

$$\begin{aligned}\text{Accept}(\mathcal{R}, x^f) &= S_{\text{init}} U_1 S_{\text{term}} = 1/n + 1/n(1 + e^{2\pi i/(n-1)} + e^{4\pi i/(n-1)} \\ &\quad + \cdots + e^{2(n-2)\pi i/(n-1)}) = 1/n \quad \text{if } f \text{ is odd} \\ \text{Accept}(\mathcal{R}, x^f) &= S_{\text{init}} U_I S_{\text{term}} = 1 \quad \text{if } f \text{ is even}\end{aligned}$$

where U_I is the n-dimensional identity matrix.

This automaton is illustrated in Figure 4. Summarizing the discussion made earlier, we are now prepared to treat all cases $n \geq 2$ in a unique way.

If there were an icftq $\mathcal{R}'$, which behaves in the same way as $\mathcal{R}$, then there must be an n-dimensional unitary matrix U_2, such that

$$\begin{aligned}\text{Accept}(\mathcal{R}', x^f) &= S_{\text{init}} U_2^f S_{\text{term}} = 1/n \text{ if } f \text{ is odd} \\ \text{Accept}(\mathcal{R}', x^f) &= S_{\text{init}} U_2^f S_{\text{term}} = 1 \quad \text{if } f \text{ is even}\end{aligned}$$

Transform U_2 in diagonal form:

$$U_2 = V \Theta V^{-1}$$

where V is a unitary matrix and Θ is a diagonal unitary matrix:

$$\Theta = \begin{bmatrix} e^{i\varphi_1} & & & & \\ & e^{i\varphi_2} & & & \\ & & \cdots & & \\ & & & e^{i\varphi_{n-1}} & \\ & & & & e^{i\varphi_n} \end{bmatrix}$$

Since

$$U_2^f = V \Theta^f V^{-1}$$

Let

$$sv = S_{\text{init}} V = (s_1, s_2, \ldots, s_n)$$

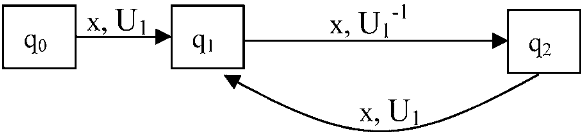

Fig. 4. A variant complex valued finite stop quantum automaton.

$$vt = V^{-1}S_{\text{term}} = \begin{bmatrix} t_1 \\ t_2 \\ \cdots \\ t_{n-1} \\ t_n \end{bmatrix}$$

we have

$$\text{Accept}(\mathcal{R}', x^f) = S_{\text{init}}U_2^f S_{\text{term}} = \sum_{j=1}^{n} s_j t_j e^{if\varphi_j} = 1/n, \quad \text{if } f \text{ is odd} \tag{28}$$

$$\text{Accept}(\mathcal{R}', x^f) = S_{\text{init}}U_2^f S_{\text{term}} = \sum_{j=1}^{n} s_j t_j e^{if\varphi_j} = 1, \quad \text{if } f \text{ is even} \tag{29}$$

Consider the $s_j t_j$ as variables, and the $e^{if\varphi_j}$ as coefficients, $j = 1, 2, \ldots, n$, $f = 1, 2, 3, \ldots$, then (28) and (29) are two systems with infinitely many linear equations each.

For all h, let $e^{i\varphi_h} = a_h$, then the coefficient matrix of (28) is the following infinite matrix:

$$C = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ a_1^3 & a_2^3 & \cdots & a_n^3 \\ a_1^5 & a_2^5 & \cdots & a_n^5 \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix}$$

Construct a sub-matrix

$$C_1(n) = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ a_1^3 & a_2^3 & \cdots & a_n^3 \\ \cdots & \cdots & \cdots & \cdots \\ a_1^{2n-1} & a_2^{2n-1} & \cdots & a_n^{2n-1} \end{pmatrix}$$

let

$$st = (s_1t_1, s_2t_2, \ldots, s_nt_n)$$

be a vector. Consider the equation system

$$\left\{ \sum_{j=1}^{n} s_j t_j e^{if\varphi_j} = 1/n, \qquad f = 1, 3, 5, \ldots, 2n-1 \right\} \tag{30}$$

and the equation system

$$\left\{\sum_{j=1}^{n} s_j t_j e^{if\varphi_j} = 1, \qquad f = 2, 4, 6, \ldots, 2n\right\} \tag{31}$$

Calculate the coefficient determinate of the equation system (30):

$$\det C_1(n) = \det\begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ a_1^3 & a_2^3 & \cdots & a_n^3 \\ \cdots & \cdots & \cdots & \cdots \\ a_1^{2n-1} & a_2^{2n-1} & \cdots & a_n^{2n-1} \end{pmatrix}\left(\prod_{j=1}^{n} a_j\right)\det C_1'(n)$$

where

$$C_1'(n) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ a_1^2 & a_2^2 & \cdots & a_n^2 \\ \cdots & \cdots & \cdots & \cdots \\ a_1^{2n-2} & a_2^{2n-2} & \cdots & a_n^{2n-2} \end{pmatrix}$$

Let $b_i = a_i^2$, then C_1' is a Van de Monde determinate with respect to b_i.

Therefore,

$$\det C_1'(n) = \prod_{n\geq u>v\geq 1}\left(a_u^2 - a_v^2\right)$$

Thus,

$$\det C_1(n) = \prod_{j=1}^{n} a_j \prod_{n\geq u>v\geq 1}\left(a_u^2 - a_v^2\right)$$

Therefore, $\det C_1(n) = 0$ if and only if there is at least one pair (a_u^2, a_v^2) with $a_u^2 = a_v^2$ (we know already that for all $a_u, a_u \neq 0$). That means, the rank of $C_1(n)$ is smaller than n, if and only if there exists a pair (a_u^2, a_v^2) with $a_u^2 = a_v^2$. It follows then: the rank of the infinite matrix C is smaller than n, if and only if there is a pair (a_u^2, a_v^2) with $a_u^2 = a_v^2$.

Assume there exist several such pairs (a_u^2, a_v^2) with $a_u^2 = a_v^2$. Assume there are altogether m mutual different a_u^2. Without loss of generality, assume these are the first m elements: $a_1^2, a_2^2, \ldots, a_m^2$. Consider the following system of m equations:

$$\left\{\sum_{j=1}^{m} c_j e^{if\varphi_j} = 1/n, \qquad f = 1, 3, 5, \ldots, 2m-1\right\} \tag{32}$$

where the numbers c_j are the results of merging equal $e^{if\varphi_j}$ items in the equations. Some of them, but not all of them, may be zero, because none of the right sides is equal to zero. Without loss of generality we still denote the number of non-zero c_j (of each equation) with m. Note that there must be equally many non-zero c'_j in all equations. In this way, we get a reduced system of linear equations, where we consider the $e^{if\varphi_j}$ as coefficients and the c_j as variables. Since now all φ_j are mutually different, the Van de Monde determinate formed by these coefficients has a non-zero value. Consider the c_j as solutions of this equation system, we have the result:

$$c_j = \frac{1}{n} \frac{\det \begin{bmatrix} e^{i\varphi_1} & \dots & 1 & \dots & e^{i\varphi_m} \\ e^{3i\varphi_1} & \dots & 1 & \dots & e^{3i\varphi_m} \\ \dots & \dots & \dots & \dots & \dots \\ e^{(2m-3)i\varphi_1} & \dots & 1 & \dots & e^{(2m-3)i\varphi_m} \\ e^{(2m-1)i\varphi_1} & \dots & 1 & \dots & e^{(2m-1)i\varphi_m} \end{bmatrix}}{\det \begin{bmatrix} e^{i\varphi_1} & \dots & e^{i\varphi_j} & \dots & e^{i\varphi_m} \\ e^{3i\varphi_1} & \dots & e^{3i\varphi_j} & \dots & e^{3i\varphi_m} \\ \dots & \dots & \dots & \dots & \dots \\ e^{(2m-3)i\varphi_1} & \dots & e^{(2m-3)i\varphi_j} & \dots & e^{(2m-3)i\varphi_m} \\ e^{(2m-1)i\varphi_1} & \dots & e^{(2m-1)i\varphi_j} & \dots & e^{(2m-1)i\varphi_m} \end{bmatrix}} = \frac{1}{n} \frac{\det C'_1(m, j)}{e^{i\varphi_j} \det C'_1(m)}$$

where $C'_1(m, j)$ is the result of replacing all elements of the jth column of $C'_1(m)$ with 1.

Now we calculate the coefficient determinate of the equation system (31). Its coefficient matrix is as follows:

$$C_2(n) = \begin{pmatrix} a_1^2 & a_2^2 & \cdots & a_n^2 \\ a_1^4 & a_2^4 & \cdots & a_n^4 \\ \dots & \dots & \dots & \dots \\ a_1^{2n} & a_2^{2n} & \cdots & a_n^{2n} \end{pmatrix}$$

$$\mathrm{Det} C_2(n) = \mathrm{Det} \begin{pmatrix} a_1^2 & a_2^2 & \cdots & a_n^2 \\ a_1^4 & a_2^4 & \cdots & a_n^4 \\ \dots & \dots & \dots & \dots \\ a_1^{2n} & a_2^{2n} & \cdots & a_n^{2n} \end{pmatrix} = \left(\prod_{j=1}^{n} a_j^2 \right) \mathrm{Det}\, C'_1(n)$$

Thus, we can repeat the discussion made earlier and get the following equation system:

$$\left\{ \sum_{j=1}^{m} c'_j e^{if\varphi_j} = 1, \qquad f = 2, 4, 6, \dots, 2m \right\} \tag{33}$$

Note that the c'_j in equation system (33) are just the same as the c_j in equation system (32). But here they are calculated with different rules, namely:

$$c'_j = \frac{\det\begin{bmatrix} e^{2i\varphi_1} & \dots\ 1\ \dots & e^{2i\varphi_m} \\ e^{4i\varphi_1} & \dots\ 1\ \dots & e^{4i\varphi_m} \\ \dots & \dots\dots\dots & \dots \\ e^{(2m-2)i\varphi_1} & \dots\ 1\ \dots & e^{(2m-2)i\varphi_m} \\ e^{2mi\varphi_1} & \dots\ 1\ \dots & e^{2mi\varphi_m} \end{bmatrix}}{\det\begin{bmatrix} e^{2i\varphi_1} & \dots & e^{2i\varphi_j} & \dots & e^{2i\varphi_m} \\ e^{4i\varphi_1} & \dots & e^{4i\varphi_j} & \dots & e^{4i\varphi_m} \\ \dots & \dots & \dots & \dots & \dots \\ e^{(2m-2)i\varphi_1} & \dots & e^{(2m-2)i\varphi_j} & \dots & e^{(2m-2)i\varphi_m} \\ e^{2mi\varphi_1} & \dots & e^{2mi\varphi_j} & \dots & e^{2mi\varphi_m} \end{bmatrix}} = \frac{\det C'_1(m, j)}{e^{2i\varphi_j}\det C'_1(m)} = \frac{nc_j}{e^{i\varphi_j}}$$

This shows that the c'_j are not equal to the c_j, which is a contradiction. Thus, we have proved that there is no icftq $\mathcal{R}'$ equivalent to the given vcftq $\mathcal{R}$. □

Theorem 6.2. *For any $n > 0$, the n-dimensional icftq languages form a proper subset of the $n+1$-dimensional icftq languages.*

Proof: It is enough to consider the case where S_{term} is a one-dimensional complex vector.

First we study the case of $n = 1$. We build a two-dimensional icftq $\mathcal{R}_2$ as follows. Let

$$\mathcal{R}_2 = (H_2, S_{\text{init}}, S_{\text{term}}, \Sigma, Q, I, T, \Delta),$$

where

$$S_{\text{init}} = \frac{1}{\sqrt{2}}(1, 1)$$

$$S_{\text{term}} = \frac{1}{\sqrt{2}}\begin{bmatrix} \frac{1-2i}{4i} \\ \frac{1+2i}{-4i} \end{bmatrix}$$

$$\Sigma = \{x\}$$

$$Q = \{q_0, q_1, q_2\}, \quad I = \{q_0\}, \quad T = \{q_1, q_2\}$$

$$\Delta = \{U_x\}, \text{ where } U_x = \begin{bmatrix} e^{i\frac{\pi}{2}} & 0 \\ 0 & e^{-i\frac{\pi}{2}} \end{bmatrix}$$

The initial conditions are fulfilled, because

$$|S_{\text{init}}|^2 = 1$$

$$|S_{\text{term}}|^2 = \frac{5}{16} < 1$$

$$|S_{\text{init}} S_{\text{term}}| = \frac{\sqrt{5}}{4} < 1$$

Now we calculate the acceptance degrees.

$$\text{Accept}(\mathcal{R}_2, x) = S_{\text{init}} U_x S_{\text{term}} = \frac{1}{4}$$

$$\text{Accept}(\mathcal{R}_2, xx) = S_{\text{init}} U_x U_x S_{\text{term}} = \frac{1}{2}$$

Assume there were a one-dimensional icftq $\mathcal{R}_1'$, which has the same functions as $\mathcal{R}_2$ does. That means:

$$\text{Accept}(\mathcal{R}_1', x) = \frac{1}{4}$$

$$\text{Accept}(\mathcal{R}_1', xx) = \frac{1}{2}$$

So

$$|\text{Accept}(\mathcal{R}_1', x)| = \frac{1}{4}$$

$$|\text{Accept}(\mathcal{R}_1', xx)| = \frac{1}{2}$$

Since each one-dimensional unitary matrix is a complex number in form of $e^{i\varphi}$, and since S_{init} and S_{term} must be also complex numbers in form of $e^{i\varphi_1}$ and $r\, e^{i\varphi_2}$, respectively, where $0 \leq r \leq 1$. Thus,

$$|\text{Accept}(\mathcal{R}_1', x)| = |e^{i\varphi_1} e^{i\varphi} \text{r} e^{i\varphi_2}| = r$$

$$|\text{Accept}(\mathcal{R}_1', xx)| = |e^{i\varphi_1} e^{2i\varphi} \text{r} e^{i\varphi_2}| = r$$

It is impossible that r equals to 1/2 and 1/4 at the same time. That means, it is impossible that an icftq $\mathcal{R}_1'$ equivalent to $\mathcal{R}_2$ exists.

Now we consider cases of $n > 1$. Build an $n+1$-dimensional icftq $\mathcal{R}_{n+1}$ as follows:

$$\mathcal{R}_{n+1} = (H_{n+1}, S_{\text{init}}, S_{\text{term}}, \Sigma, Q, I, T, \Delta),$$

where

$$\Sigma = \{x\}$$

$$Q = \{q_0, q_1, q_2, \ldots, q_n, q_{n+1}\}, \quad I = \{q_0\}, \quad T = \{q_1, q_2, \ldots, q_n, q_{n+1}\},$$

$$\Delta = \{U_x\},$$

where

$$U_x = \begin{bmatrix} e^{i\varphi_1} & & & & \\ & e^{i\varphi_2} & & & \\ & & \cdots & & \\ & & & e^{i\varphi_n} & \\ & & & & e^{i\varphi_{n+1}} \end{bmatrix}, \quad \text{where for } j \neq k,\ e^{i\varphi_j} \neq e^{i\varphi_k}$$

Consider the following system consisting of $n+1$ linear equations:

$$\sum_{j=1}^{n+1} s_j t_j e^{ki\varphi_j} = a, \quad a > 0,\ 1 \leq k \leq n$$

$$\sum_{j=1}^{n+1} s_j t_j e^{(n+1)i\varphi_j} = 2a \tag{34}$$

We calculate the coefficient determinate of system (34) based on the Van de Monde rule:

$$\begin{aligned} \mathrm{Det}C(n+1,1,\varphi) &= \det \begin{bmatrix} e^{i\varphi_1} & e^{i\varphi_2} & \cdots & e^{i\varphi_{n+1}} \\ e^{2i\varphi_1} & e^{2i\varphi_2} & \cdots & e^{2i\varphi_{n+1}} \\ \cdots & \cdots & \cdots & \cdots \\ e^{(n+1)i\varphi_1} & e^{(n+1)i\varphi_2} & \cdots & e^{(n+1)i\varphi_{n+1}} \end{bmatrix} \\ &= \prod_{j=1}^{n+1} e^{i\varphi_j} \det C(n+1,0,\varphi) \\ &= \prod_{j=1}^{n+1} e^{i\varphi_j} \prod_{1 \leq g < h \leq n} (e^{i\varphi_h} - e^{i\varphi_g}) \end{aligned}$$

This is not equal to zero because we have assumed that the $e^{i\varphi_j}$ are mutually different. Therefore, the linear system (34) is uniquely solvable. By solving the equation system, we get a solution for all $c_j = s_j t_j$.

Note that it is possible that the values of $|c_j|$ are not small enough, such that the initial conditions $|S_{\text{init}}|^2 = 1, |S_{\text{term}}|^2 \leq 1$ do not necessary hold. But we can let the constant a small enough, such that the condition $|c_1|^2 + |c_2|^2 + \cdots + |c_n|^2 = b$ is fulfilled for some $b \leq 1$. The constant b does not equal to zero because otherwise all c_j would be zero and the constant a on the right sides of the equation system would be also zero, which contradicts the assumption. Now let $S_{\text{init}} = [s_1, s_2, \ldots, s_n]$, $S_{\text{term}} = [t_1, t_2, \ldots, t_n]$,

For all j, let $s_j = \frac{c_j}{\sqrt{b}}$, $t_j = \sqrt{b}$, then

$$\begin{aligned} |S_{\text{init}} S_{\text{term}}|^2 &= |s_1 t_1|^2 + |s_2 t_2|^2 + \cdots + |s_n t_n|^2 = |c_1|^2 + |c_2|^2 + \cdots + |c_n|^2 \\ &= b \leq 1, \quad |S_{\text{init}}|^2 = 1, \qquad |S_{\text{term}}|^2 = b^2 \leq 1 \end{aligned}$$

Thus, all three conditions are fulfilled.

If there were an n-dimensional icftq, say $\mathcal{R}_{n,}$, which is equivalent to $\mathcal{R}_{n+1}$, then we must have: $\mathcal{R}_{n,} = (H_n, \hat{S}_{\text{init}}, \hat{S}_{\text{term}}, \hat{\Sigma}, \hat{Q}, \hat{I}, \hat{T}, \hat{\Delta})$ where $\hat{\Sigma} = \Sigma$, $\hat{\Delta} = \{\hat{U}_x\}$ such that

$$\begin{aligned} \hat{S}_{\text{init}} \hat{U}_x^n \hat{S}_{\text{term}} &= a, \quad 1 \le k \le n \\ \hat{S}_{\text{init}} \hat{U}_x^{n+1} \hat{S}_{\text{term}} &= 2a, \end{aligned} \tag{35}$$

where $\hat{S}_{\text{init}}$ is an n-dimensional complex vector and $\hat{S}_{\text{term}}$ is a transposed n-dimensional complex vector. Both of them should satisfy the initial conditions. $\hat{U}_x$ is a unitary matrix of order n. Let

$$\hat{U}_x = V \Theta V^{-1},$$

where V is unitary, Θ is unitary and diagonal:

$$\Theta = \begin{bmatrix} e^{i\phi_1} & & & \\ & e^{i\phi_2} & & \\ & & \cdots & \\ & & & e^{i\phi_n} \end{bmatrix} \tag{36}$$

Let $\tilde{S}_{\text{init}} = \hat{S}_{\text{init}} V = (s_1, s_2, \ldots, s_n)$,

$$\tilde{S}_{\text{term}} = V^{-1} \hat{S}_{\text{term}} = (t_1, t_2, \ldots, t_n)'$$

We get the following linear system of equations:

$$\begin{aligned} \sum_{j=1}^{n} s_j t_j e^{ki\phi_j} &= a, \quad 1 \le k \le n \\ \sum_{j=1}^{n} s_j t_j e^{(n+1)i\phi_j} &= 2a \end{aligned} \tag{37}$$

Consider the first n equations of (37). Their coefficient determinate is:

$$\begin{aligned} \det C(n, 1, \varphi) &= \det \begin{bmatrix} e^{i\phi_1} & \cdots & e^{i\phi_n} \\ \cdots & \cdots & \cdots \\ e^{ni\phi_1} & \cdots & e^{ni\phi_n} \end{bmatrix} \\ &= \prod_{j=1}^{n} e^{i\phi_j} \prod_{1 \le g < h \le n} (e^{i\phi_h} - e^{i\phi_g}) \end{aligned}$$

Since we are not sure whether there are some $e^{i\phi_j}$ mutual equal. We must assume a number m of mutually unequal $e^{i\phi_j}$ with $m \le n$. Merge the equal $e^{i\phi_j}$ together we get a reduced form of system (37):

$$\sum_{j=1}^{m} c'_j e^{ki\phi_j} = a, \qquad 1 \le k \le n$$

$$\sum_{j=1}^{m} c'_j e^{(n+1)i\phi_j} = 2a \tag{38}$$

where c'_j are the new coefficients produced by merging the terms $e^{i\phi_h}$.

$m = 1$ is impossible, because this would lead to

$$\begin{aligned} |c'_j| &= a \quad \text{by considering the first } n \text{ equations, and} \\ |c'_j| &= 2a \text{ by considering the } n+1\text{th equation,} \end{aligned}$$

This is a contradiction because $a \neq 0$.

For $m > 1$ we consider the system consisting of the last m ones of the first n equations of the equation system (38):

$$\sum_{j=1}^{m} c'_j e^{ki\phi_j} = a, \quad n - m + 1 \le k \le n \tag{39}$$

The coefficient determinate of the system (39) is:

$$\det C(m, n-m+1, \phi) = \prod_{j=1}^{m} e^{(n-m+1)i\phi_j} \det\, C(m, 0, \phi)$$

$$= \prod_{j=1}^{m} e^{(n-m+1)i\phi_j} \det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{i\phi_1} & e^{i\phi_2} & \cdots & e^{i\phi_m} \\ \cdots & \cdots & \cdots & \cdots \\ e^{(m-1)i\phi_1} & e^{(m-1)i\phi_2} & \cdots & e^{(m-1)i\phi_m} \end{bmatrix}$$

which is not equal to zero. Therefore, we can represent the solutions of this equation system as follows:

$$c'_j = \frac{\det C(m, n-m+1, j, a, \phi)}{\det C(m, n-m+1, \phi)}$$

where the matrix $C(m, n-m+1, j, a, \phi)$ is obtained by replacing all elements of the jth column of the matrix $C(m, n-m+1, \phi)$ with a.

Now consider a new system consisting of the last $m-1$ equations of the first n equations of (38) and the last equation of (38).

$$\sum_{j=1}^{m} c'_j e^{ki\phi_j} = a, \quad n - m + 2 \le k \le n$$

$$\sum_{j=1}^{m} c'_j e^{(n+1)i\phi_j} = 2a \tag{40}$$

The coefficient determinate of (40) is

$$\det C(m, n-m+2, \phi) = \prod_{j=1}^{m} e^{(n-m+2)i\varphi_j} \det C(m, 0, \phi)$$

Consider the c'_j as variables and solve the equation system (40). Denote the solutions with c''_j, we have

$$c''_j = \frac{\det C(m, n-m+2, j, (a, 2a), \phi)}{\det C(m, n-m+2, \phi)} \tag{41}$$

where $C(m, n-m+2, j, (a, 2a), \phi)$ is that matrix we obtain when we replace all elements of the jth column of the matrix $C(m, n-m+2, \phi)$ with a in the first $m-1$ rows and $2a$ in the last row. Then

$$c''_j = \frac{\det C(m, 0, j, (a, 2a), \phi)}{e^{(n-m+2)i\phi_j} \det C(m, 0, \phi)} \tag{42}$$

Consider the following difference:

$$
\begin{aligned}
& e^{(n-m+1)i\phi_j} \det C(m, 0, \phi)[c''_j - c'_j] \\
&= \frac{1}{e^{i\phi_j}} (\det C(m, 0, j, a, \phi) + \det C(m, 0, j, (0, a), \phi)) - \det C(m, 0, j, a, \phi) \\
&= \frac{a}{e^{i\phi_j}} (\det C(m, 0, j, 1, \phi) + \det C(m, 0, j, (0, 1), \phi)) - \det C(m, 0, j, a, \phi) \\
&= \frac{a}{e^{i\phi_j}} \left(\prod_{\substack{1\le g<h\le m,\\ \phi_j=0}} (e^{i\phi_h} - e^{i\phi_g}) + (-1)^{j+m} \prod_{\substack{1\le g<h\le m,\\ g\ne j, h\ne j}} (e^{i\phi_h} - e^{i\phi_g}) \right) \\
&\quad - a \prod_{\substack{1\le g<h\le m,\\ \phi_j=0}} (e^{i\phi_h} - e^{i\phi_g}) \\
&= \frac{a}{e^{i\phi_j}} \prod_{\substack{1\le g<h\le m,\\ g\ne j, h\ne j}} (e^{i\phi_h} - e^{i\phi_g}) \left[\prod_{j<h\le m} (e^{i\phi_h} - 1) \prod_{1\le g<j} (1 - e^{i\phi_g}) \right. \\
&\quad \left. + (-1)^{j+m} - e^{i\phi_j} \prod_{j<h\le m} (e^{i\phi_h} - 1) \prod_{1\le g<j} (1 - e^{i\phi_g}) \right]
\end{aligned}
$$

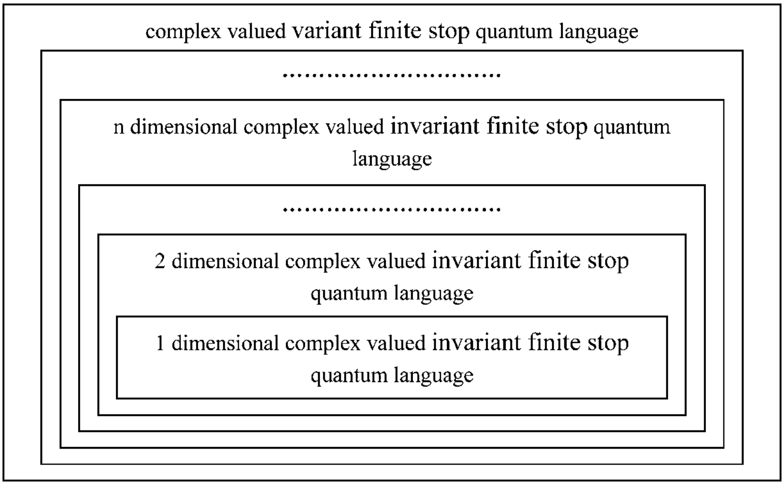

Fig. 5. The hierarchy of complex valued quantum language inclusion.

We will prove that the right side is not equal to zero. If it were zero, then we would have the equation:

$$\prod_{j<h\leq m}(e^{i\phi_h}-1)\prod_{1\leq g\leq j}(1-e^{i\phi_g})=(-1)^{j+m+1} \tag{43}$$

Denote the left side of (43) with $E(j)$, then

$$\frac{E(2)}{E(1)}=\frac{(1-e^{i\phi_2})}{(e^{i\phi_1}-1)}=-1$$

That would mean:

$$e^{i\phi_1}=e^{i\phi_2}$$

which contradicts our assumption of m mutually different $e^{i\phi_j}$. □

The hierarchy of quantum language inclusion discussed earlier is shown in Figure 5.

7. RELATED WORKS

It has been always an interesting problem of studying the relationships between languages accepted by different kinds of automata. The research on relations between formal languages dates back at least to Turing (1936), when he published his fundamental paper on Turing machines. It was an early result that the languages accepted by Turing machines, called recursively enumerable languages, with one or multiple tapes, are identical. Further, Shannon proved that any Turing machine with m input symbols and n internal states can be simulated by another one with only two input symbols and $8mn$ states, or one with two states and $4mn + m$ input symbols (Shannon, 1956). Chomsky (1956, 1959) has studied the hierarchy of formal grammars. Since it was shown that there exists a correspondence between grammars and automata at different levels of this hierarchy, e.g. (Chomsky, 1959; Chomsky and Miller, 1958; Kuroda, 1964), the hierarchy of formal grammars presents also a hierarchy of formal languages accepted by different automata. Among these researches, the research on regular languages accepted by finite state automata plays an important role.

The Chomsky hierarchy is not the only hierarchy of formal languages. The Lindenmayer system, L system for short, is a kind of parallel rewriting system which is a result of applying the theory of formal languages to the developmental biology (Lindenmayer and Prusinkiewicz, 1990). Each L system is a quadruple $G = \langle V, P, g, \omega\rangle$, where V is a finite symbol table. $\omega \in V^+$ is called an axiom. g is an environment symbol, which does not belong to the symbol table. P is a set of production rules. The parallel rewriting of a string S in a L system means replacing all symbols a in S with β at the same time by applying some rule $\alpha a \gamma \rightarrow \beta$ under the context condition (α, γ).

If we consider the symbols in V as cells, then the axiom is the origin of the biological object, the productions are the rules of cell division, and the process of rewriting is the process of cell division. The repeated execution of parallel rewriting means the development of the biological object. A (sub-)language denotes the development of biological specie. A language family denotes the development of different species, which have the same origin.

Many sub families of L systems have been proposed (Rozenberg and Salomaa, 1980). To list a few examples, they include IL systems (context-dependent L systems), 0L systems (context independent L systems), D0L systems (deterministic 0L systems), PD0L systems (propagating D0L systems), etc. The formal languages produced by these devices form a hierarchy, which does not fit into the Chomsky hierarchy. Moreover, Lu and Zhang (2002) have proposed generalized L systems, where the generation of L languages proceeds not only in a way of full synchronization, but can also in an asynchronous way. It was proved in (Lu and Zhang, 2002) that the generalized L systems are more powerful than traditional L systems. For example, the PD0L languages form a proper subset of GPD0L (generalized PD0L) languages. Besides, each GPD0L system is characterized by a finite set F of positive integers $\{f_1, f_2, \ldots, f_n\}$, which denotes the different numbers of time steps needed for language generation, called fission periods. It was proved in (Lu and Zhang, 2002) that for $F_1 \subset F_2$ the languages produced by system L_1 form a proper subset of those produced by system L_2. It was also proved that the sufficient and necessary condition for GPD0L $[m_1, m_2, \ldots, m_j]$ and GPD0L $[n_1, n_2, \ldots, n_k]$ to be equivalent is (1) $j = k$, and (2) there exist common divisor h of $(m_1, m_2, \ldots, m_j)$ and common divisor g of $(n_1, n_2, \ldots, n_k)$, such that $\frac{m_i}{h} = \frac{n_i}{g}, i = 1, \ldots, j$. This shows that they form a complete hierarchy of languages (Lu and Zhang, 2002).

It is a natural idea to study the relationship between languages accepted by quantum automata, which are explored from different points of view. Based on the idea of von Neumann type quantum logic, Ying has studied languages accepted by lattice valued finite state quantum automata and obtained a series of results (Ying, 2000a, 2000b). Lu and Zheng have studied properties of lattices built by quantum automata (Lu and Zheng, 2003), where it was proved that languages accepted by deterministic and non-deterministic quantum automata are generally not equivalent.

Quantum automata based on Hilbert space and probability have been studied extensively in the literature, e.g. (Kondacs and Watrous, 1997; Moore and Crutchfield, 2000). For such automata, language hierarchies have been studied, e.g. by Andris Ambainis *et al.* (1999), where a definition on finite state quantum automata, following that given in (Kondacs and Watrous, 1997) and other than that given in (Moore and Crutchfield, 2000) was presented. The following theorem was proved: there is a hierarchy of regular languages such that each language in the hierarchy can be accepted by one-way finite state quantum automata with a probability smaller than the corresponding probability for its preceding language in the same hierarchy. Thus, the hierarchy is characterized by the gradual increase of values of probabilities. We have, on the one hand, proved a classification of all quantum languages (not only a particular hierarchy of languages) accepted by finite state quantum automata in the sense of (Moore and Crutchfield, 2000), and, on the other hand, defined a new type of finite state quantum automata, the finite stop quantum automata. This definition is different from both given in (Kondacs and Watrous, 1997; Moore and Crutchfield, 2000). It is worth pointing out that in this paper not only a hierarchy of languages, but also a hierarchical classification of all finite stop quantum automata with respect to this definition is given. This was just the motivation of our research.

8. CONCLUSIONS

In this paper, first we proposed a definition of finite stop quantum automata based on Hilbert space and compared it with the finite state quantum automata proposed by Moore and Crutchfield (2000). We proved that the languages accepted by finite state quantum automata form a proper subset of the languages accepted by finite stop quantum automata. Furthermore, we proved the facts that the language class accepted by n-dimensional finite state quantum automata is a proper subclass of the language class accepted by $n + 1$-dimensional finite state quantum automata. In addition, we have introduced complex valued acceptance degrees for quantum automata. In this way, two new types of quantum automata have been defined and the relations between the languages accepted by them have been discussed. It was proved that the language class accepted by invariant finite stop quantum automata is a proper subclass of the language class accepted by variant finite stop quantum automata. Furthermore, we have also proved that the language class accepted by n-dimensional invariant finite stop quantum automata is a proper subclass of the language class accepted by $n + 1$-dimensional invariant finite stop quantum automata. In this way we have established two infinite hierarchies of quantum automata: the finite state ones and the invariant finite stop ones. The third infinite hierarchy, namely the hierarchy of variant finite stop quantum automata, has not yet been proven and thus remains an open problem.

ACKNOWLEDGMENTS

The authors wish to express their sincere thanks to Prof. Mingsheng Ying, who has pointed out an inappropriate statement in the original manuscript and made helpful suggestions to its improvement. This work and the previous work Lu and Zheng (2004) are partially supported by CAS project of brain and mind science, Pre-973 project 2001CCA03000, 863 project 2001AA113130, 973 Program 2001CB312004, the innovation foundation of IOM, AMSS and ICT projects, NSFC major program 60496324.

REFERENCES

Ambainis, A., Bonner, R., Freivalds, R., and Kikusts, A. (1999). A hierarchy of languages accepted by quantum finite automata. In *QCL 99, International Workshop on Quantum Computation and Learning*, Riga.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory* **2**(3), 113–124.

Chomsky, N. (1959). On certain formal properties of grammars. *Information and Control* **2**(2), 137–167.

Chomsky, N. and Miller, G. A. (1958). Finite state languages. *Information and Control* **1**(2), 91–112.

Cohen, D. W. (1989). *An Introduction to Hilbert Space and Quantum Logic*, Springer, New York.

Cohn, P. M. (1981). *F.R.S. Universal Algebra*, D. Reidel Publishing Company.

Deutsch, D. (1985). Quantum theory, the church Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London, Series A* **400**, 97–117.

Greechie, R. (1981). A non-standard quantum logic with a strong set of states. In *Current Issues in Quantum Logic*, Vol. 8, E. G. Beltrametti and B. C. van Fraassen, eds., Plenum Press, New York, pp. 375–380.

Gudder, S. (1999). Basic properties of quantum automata. *International Journal of Theoretical Physics* **38**, 2261–2282.

Hermes, H. (1955). *Einfuehrung in die Verbandstheorie*, Springer-Verlag, Berlin.

Hopcroft, J. E. and Ullman, J. D. (1979). *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley.

Kondacs, A. and Watrous, J. (1997). On the power of quantum finite state automata. In *Proceedings of 38th FOCS*, pp. 66–75.

Kuroda, S. Y. (1964). Classes of languages and linear bounded automata. *Information and Control* **7**(2), 207–223.

Lindenmayer, A. and Prusinkiewicz, P. (1990). *The Algorithmic Beauty of Plants*, Springer-Verlag, Berlin.

Lu, R. and Zhang, W. (2002). Generalized L systems. *Science in China, Series F* **45**(3), 220–231.

Lu, R. and Zheng, H. (2003). Lattices of quantum automata. *International Journal of Theoretical Physics* **42**(7), 1425–1449.

Lu, R. and Zheng, H. (2004). Pumping Lemma of quantum automata. *International Journal of Theoretical Physics* **43**(5), 1191–1217.

Mackey, G. W. (1957). Quantum mechanics and Hilbert space. *American Mathematics* **64**, 45–57.

Moore, C. and Crutchfield, J. P. (2000). Quantum automata and quantum grammars. *Theoretical Computer Science* **237**(1–2), 275–306.

Qiu, D. (2003). Automata and grammars theory based on quantum logic. *Journal of Software* **14**, 23–27.

Rawling, J. P. and Selesnick, S. A. (2000). Orthologic and quantum logic: Models and computational elements. *JACM* **47**(4), 721–751.

Reichenbach, H. (1948). *Philosophic Foundations of Quantum Mechanics*, California.

Rozenberg, G. and Salomaa, A. (1980). *The Mathematical Theory of L Systems*. Academic Press, New York.

Shannon, C. E. (1956). A universal Turing machine with two internal states. In *Automata Studies*, Princeton University Press, Princeton, NJ, pp. 129–153.

Turing, A. M. (1936). On computable numbers with an application to the Entscheidungs problem. *Proceedings of the London Mathematical Society* **2**(42), 230–265 (correction on **2**(43), 544–546).

Ying, M. S. (2000a). Automata theory based on quantum logic. I. *International Journal of Theoretical Physics* **39**(4), 985–995.

Ying, M. S. (2000b). Automata theory based on quantum logic. II. *International Journal of Theoretical Physics* **39**(11), 2545–2557.