International Journal of Theoretical Physics, Vol. 42, No. 7, July 2003 (© 2003)

Lattices of Quantum Automata

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

Received September 23, 2002

We defined and studied three different types of lattice-valued finite state quantum *automata (LQA) and four different kinds of LQA operations*, discussed their advantages, disadvantages, and various properties. There are four major results obtained in this paper. First, no one of the above mentioned LQA follows the law of lattice value conservation. Second, the theorem of classical automata theory, that each nondeterministic finite state automaton has an equivalent deterministic one, is not necessarily valid for finite state quantum automata. Third, we proved the existence of semilattices and also lattices formed by different types of LQA. Fourth, there are tight relations between properties of the original lattice l and those of the l-valued lattice formed by LQA.

KEY WORDS: quantum automata; lattice of quantum automata; operation of quantum automata.

1. INTRODUCTION

Studies in quantum computing are of interest not only as explorations for smart quantum algorithms and efficient implementations of quantum circuits, but also as basic researches in quantum computation theory (Deutsch, 1985). As a matter of fact, an appropriate logic interpretation of quantum mechanics has attracted attention of many scientists already in the early years of last century. From Heisenberg to Reichenbach (Reichenbach, 1998), scientists have recognized that the classic (two-valued) logic can no longer serve as the basis of a modem theory of quantum physics. It was von Neumann who has studied a new type of logic, the quantum logic, where lattices are taken as mathematical models for such logics.

[1] Open Lab of Intelligent Information Processing, Fudan University, Shanghai, People's Republic of China.
[2] Key Lab of Multimedia and Intelligent Software, Beijing Polytechnic University, Beijing, People's Republic of China.
[3] Key Lab of IIP, Institute of Computing Technology, Academia Sinica, Beijing, People's Republic of China.
[4] Key Lab of MADIS, Academia Sinica, Beijing, People's Republic of China.
[5] Institute of Mathematics, AMSS, Academia Sinica, Beijing, People's Republic of China.
[6] To whom correspondence should be addressed at Institute of Mathematics, AMSS, Academia Sinica, 100080, Beijing, People's Republic of China; e-mail: rqlu@math08.math.ac.cn.

0020-7748/03/0700-1435/0 © 2003 Plenum Publishing Corporation

Nowadays, with ever-increasing interest in quantum computing, different kinds of quantum logics have been put forward (Greechie, 1981; Rawling and Selesnick, 2000). They are based on different principles, for example based on probability (Moore and Crutchfield, 2000). Recently, Ying has studied lattice-valued quantum logic and lattice-valued finite-state quantum automata and obtained a series of results (Ying, 2000a,b). This paper is a further study on the latter with the aim of providing more insight in the mathematical basis of quantum computation. As it is well known, the theory of finite state automata is an essential part of the classical theory of computation. We are convinced that the lattice-valued finite-state quantum automata will also play an important role in developing a sound and powerful theory of quantum computation.

For the subsequent use in this paper, we first give some essential facts about lattices.

Following are two equivalent definitions of lattice:

Definition 1.1. (Cohn, 1981): A set L of elements forms a lattice, if

1. L is a partial ordered ($\subseteq$) set
2. For each pair of elements a, b of L, there is a unique least upper bound sup (a, b) and a unique largest lower bound inf (a, b), with the representation:

$$\sup(a, b) = a \bigcup b, \inf(a, b) = a \bigcap b.$$

Definition 1.2. (Hermes, 1955). A set L of elements forms a lattice if there are two binary operations $\bigcup$ and $\bigcap$ defined on L, such that for all elements a and b of L:

1. The two commutative rules:

$$a \bigcup b = b \bigcup a,$$
$$a \bigcap b = b \bigcap a.$$

2. The two associative rules:

$$\left(a \bigcap b\right) \bigcap c = a \bigcap \left(b \bigcap c\right)$$
$$\left(a \bigcup b\right) \bigcup c = a \bigcup \left(b \bigcup c\right)$$

3. and the two absorption rules:

$$a \bigcap \left(a \bigcup b\right) = a,$$
$$a \bigcup \left(a \bigcap b\right) = a$$

hold.

Proposition 1.3. (Cohn, 1981; Hermes, 1955). *Definitions 1.1 and 1.2 are equivalent.*

In this paper we will use the symbol l to denote a lattice in general. The capital letter L is only used to denote its element set.

Corollary 1.4. *For a lattice l, and finite set S of its elements has a unique least upper bound* sup (S) *and a unique largest lower bound* inf (S), *where*

$$\sup(S) = \bigcup\{x|x \in S\}, \qquad \inf(S) = \bigcap\{x|x \in S\}$$

Both sup (S) and inf (S) belong to L.

Definition 1.5. If l has a greatest element sup (l) and a least element inf (l), then l is called a bounded lattice. If any (finite or infinite) set S of the elements of a lattice l has a least upper bound sup (S) and a greatest lower bound inf (S), which belong to L, then l is called a complete lattice.

In this paper we concern only bounded lattices and will use the notation $l = (L, \subseteq, 0, 1)$ to describe a lattice, where L is the set of lattice elements, $\subseteq$ describes the partial order of lattice elements, 0 is the least element and 1 the greatest element. Due to Proposition 1.3, a lattice can be also represented in form $l = (L, \bigcap, \bigcup, 0.1)$.

Definition 1.6. If for any elements a, b, and c of a lattice l, the following two distributive rules hold:

$$a \bigcap \left(b \bigcup c\right) = \left(a \bigcap b\right) \bigcup \left(a \bigcap c\right),$$
$$a \bigcup \left(b \bigcap c\right) = \left(a \bigcup b\right) \bigcap \left(a \bigcup c\right),$$

then it is called a distributive lattice.

Proposition 1.7. *The two equations of Definition 1.6 are equivalent, i.e., any one of them can be inferred from another.*

Definition 1.8. If for any elements, a, b, and c of a lattice l, the following modular rule holds:

$$a \subseteq c \rightarrow a \bigcup \left(b \bigcap c\right) = \left(a \bigcup b\right) \bigcap c$$

then l is called a modular lattice.

Proposition 1.9. *For any elements a, b, and c of a lattice l, the following inclusion rules hold:*

$$\left(a\bigcap b\right)\bigcup\left(a\bigcap c\right)\subseteq a\bigcap\left(b\bigcup c\right)$$
$$a\bigcup\left(b\bigcap c\right)\subseteq\left(a\bigcup b\right)\bigcap\left(a\bigcup c\right)$$

Proposition 1.10. *From Propositions 1.7 and 1.9 follows, that the lattice l is a distributive lattice if one of the following inclusion rules holds:*

$$a\bigcap\left(b\bigcup c\right)\subseteq\left(a\bigcap b\right)\bigcup\left(a\bigcap c\right),$$
$$\left(a\bigcup b\right)\bigcap\left(a\bigcup c\right)\subseteq a\bigcup\left(b\bigcap c\right).$$

Proposition 1.11. *For any elements a, b, and c of a lattice l, the following inclusion rule holds:*

$$a\bigcup\left(b\bigcap c\right)\subseteq\left(a\bigcup b\right)\bigcap c$$

Proposition 1.12. *From Proposition 1.11 and Definition 1.8 follows, that the lattice l is a modular lattice if for any elements a, b, and c of l, the following inclusion rule holds:*

$$a\subseteq c\rightarrow\left(a\bigcup b\right)\bigcap c\subseteq a\bigcup\left(b\bigcap c\right)$$

Proposition 1.13. *A lattice l is modular if and only if for any elements a, b, and c of l,*

$$a\bigcup\left(b\bigcap\left(a\bigcup c\right)\right)=\left(a\bigcup b\right)\bigcap\left(a\bigcup c\right)$$

2. LATTICE-VALUED FINITE-STATE QUANTUM AUTOMATA

First we repeat the definition of a lattice-valued finite-state quantum automaton, LQA for short, defined by Ying (2000a) in a slightly different notation. At the same time we will introduce a new version of LQA and compare their acceptance characteristics.

Definition 2.1. Let $l = (L, \subseteq, 0, 1)$ be a lattice, Σ be a finite alphabet, called the input alphabet. An LQA $\mathscr{R}$ defined on (l, Σ) is a quadruple $\mathscr{R} = (Q, I, T, \triangle)$, where $I \subseteq Q$ is a set of initial states, $T \subseteq Q$ is a set of terminating states, $\triangle$ is a set of l-valued predicates defined on $Q \times \Sigma \times Q$: for each $q_1, q_2 \in Q$ and $x \in \Sigma$, $\delta(q_1, x, q_2) \in \triangle$ is an element of l. Note that in $\triangle$ only those $\delta(q_1, x, q_2)$, which are not equal to 0 (least element of l) are listed in $\triangle.\delta(q_1, x, q_2)$ is called the acceptance degree of x by $\mathscr{R}$ that the state q_1 is transformed to q_2 when the symbol x is input. The LQA are classified in type A LQA and type B LQA in Definition 2.4 according to the way the acceptance degree is calculated.

Definition 2.2. Let $\mathscr{R} = (Q, I, T\triangle)$ be a LQA defined on (l, Σ). For each i, j, where $\delta(q_i, x, q_j) \neq 0$, where 0 is the least element of the lattice l, the triplet $q_i\ x\ q_j$ is called a transition of $\mathscr{R}$. A finite connection of transitions $q_0\ x_1 q_1 x_2 \cdots q_{n-1} x_n q_n$ is called a transition sequence of $\mathscr{R}$ where all q belong to Q and all x belong to Σ. If besides q_0 belongs to I and q_n belongs to T, then we say the transition sequence is successful and is called a single path w of $\mathscr{R}$. We call the symbol sequence $s = x_1\ x_2 \cdots x_n$ the label of w, or an accepted string of the automaton $\mathscr{R}$. The acceptance degree of s by this single path is defined as: $\text{Accept}_w\ (\mathscr{R}, s) = \bigcap_{i=0}^{n-1} \delta(q_i, x_{i+1}, q_{i+1})$, where $\bigcap$ is the inf operation of the lattice l.

Since the automaton is in general non deterministic, the same label may be contained in more than one path. We have yet to define the acceptance degree in general.

Proposition 2.3. *Let $\mathscr{R} = (Q, I, T, \triangle)$ be a LQA defined on (l, Σ), $s = x_1 x_2 \cdots x_n$ be a string of Σ^*. Let $T(\mathscr{R}, s) = \{w \mid w$ is a path of $\mathscr{R}$, s is the label of $w\}$ denote the "distributed path" in $\mathscr{R}$ accepting s. Then $|T(\mathscr{R}, s)|$ is finite for every s.*

The proof is easy and will not be paraphrased here.

With help of this proposition we are able to give the following:

Definition 2.4. Let $\mathscr{R}$, s and $T(\mathscr{R}, s)$ be defined as above. The general acceptance degree of s by $\mathscr{R}$ is defined in two different ways:

1. $$\text{Accept}_\text{A}(\mathscr{R},\ s) = \bigcup_{w\in T(R,s)} \bigcap_{i=0}^{n-1} \delta(q_{w,i}, x_{i+1}, q_{w,i+1})$$

2. $$\text{Accept}_\text{B}(\mathscr{R},\ s) = \bigcap_{i=0}^{n-1} \bigcup_{w\in T(R,s)} \delta(q_{w,i}, x_{i+1}, q_{w,i+1})$$

where $q_{w,i}$, $i = 0, 1, 2, \ldots, n$, are the states traversed by the path w. LQA defined on the basis of Accept_A and Accept_B are called LQA of type A and type B. In word, in the case of type A, we first calculate the acceptance degree of s for each single path, then unite them together. In the case of type B, we take the first transition of all paths in $T(\mathscr{R}, s)$, unite their acceptance degree $\delta(q_{w,0}, x_1, q_{w,1})$ together by the sup operation $\bigcup$. And then we take the second, third, . . . transition of all paths and unite their acceptance degrees separately. At last, we perform the inf operation $\bigcap$ on all these united values and get the wanted general acceptance degree of s by $\mathscr{R}$ In any case, the language accepted by $\mathscr{R}$ is $\{(s, \text{Accept}(\mathscr{R}, s)) \mid s \in \Sigma^*\}$.

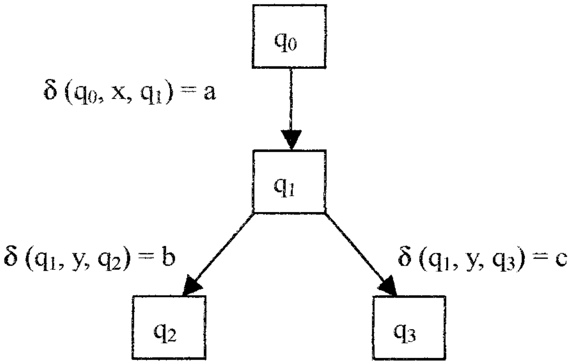

Fig. 1. ALQA with four states.

Example 2.5. Let $\mathscr{R} = (\{q_0, q_1, q_2, q_3\}, \{q_0\}, \{q_2, q_3\}, \{\delta(q_0, x, q_1) = a, \delta(q_1, y, q_2) = b, \delta(q_2, y, q_3) = c\})$. There are two paths for accepting the string *xy* in this automaton: $w_1 = q_0\, x\, q_1\, y\, q_2$ and $w_2 = q_0\, x\, q_1\, y\, q_3$. The single path acceptance degrees are Accept$w_1(\mathscr{R}, xy) = a \bigcap b$ and Accept$w_2(\mathscr{R}, xy) = a \bigcap c$ respectively. The general acceptance degree of type A is $\text{Accept}_\text{A}(\mathscr{R}, xy) = (a \bigcap b) \bigcup (a \bigcap c)$. On the other hand, the general acceptance degree of type B is $\text{Accept}_\text{B}(\mathscr{R}, xy) = a \bigcap (b \bigcup c)$. See Fig. 1 for an illustration.

According to Proposition 1.9, we have $\text{Accept}_\text{A}(\mathscr{R}, xy) = (a \bigcap b) \bigcup (a \bigcap c) \subseteq a \bigcap (b \bigcup c) = \text{Accept}_\text{B}(\mathscr{R}, xy)$. In fact, we have the more general

Proposition 2.6. *For any LQA $\mathscr{R}$ the general acceptance degrees (called recognizability in Ying (2000a)) calculated according to rules of type A and type B have the relationship:*

1. $Accept_A(\mathscr{R}, s) \subseteq Accept_B(\mathscr{R}, s)$
2. *The inclusion symbol $\subseteq$ can be replaced by the equation symbol $=$ if $\mathscr{R}$ is a deterministic LQA.*

That means, the type B LQA is more "generous" in nondeterministically accepting a string from Σ^*.

Proof: This proposition can be proved based on a mathematical induction with help of Proposition 1.9, can also follow directly from Lemma 2.14 given below. □

The remaining discussion of this section relates only to LQA of type B, unless it is otherwise stated.

Note that any of the four components Q, I, T, Δ, of $\mathscr{R}$ in Definition 2.1 can be empty. In this case the automaton $\mathscr{R}$ accepts no sentence from Σ^* no matter whether it is of types A or B. Any such $\mathscr{R}$ is called an empty automaton.

Based on the concept of acceptance degree, it is possible to define a partial order for the quantum automata.

Definition 2.7. Assume $\mathscr{R}_1$ and $\mathscr{R}_2$ are two LQA (both of type A or both of type B) defined on $(1, \Sigma)$. If

$$\text{For all } s \in \Sigma^*, \quad \text{Accept}(\mathscr{R}_1, s) \subseteq \text{Accept}(\mathscr{R}_2, s),$$

Then we say that $\mathscr{R}_1, \subseteq' \mathscr{R}_2$.

Proposition 2.8. *The inclusion relation given in Definition 2.7 defines a partial order.*

Definition 2.9. Two LQA $\mathscr{R}_1$ and $\mathscr{R}_2$ defined on the same (l, Σ) (both of types A or B) are called semantically equivalent if for any s of Σ^*,

$$\text{Accept}(\mathscr{R}_1, s) \subseteq \text{Accept}(\mathscr{R}_2, s), \qquad \text{Accept}(\mathscr{R}_2, s) \subseteq \text{Accept}(\mathscr{R}_1, s),$$

are both valid. In this case we use the notation:

$$\text{Accept}(\mathscr{R}_1, s) = \text{Accept}(\mathscr{R}_2, s), \quad \text{or} \quad \mathscr{R}_1 \approx \mathscr{R}_2$$

to represent their relationship, where the equation symbol $=$ means the identity of lattice elements in l.

Since we have established a partial order for the LQA, we are now interested in exploring the question: is it possible to build a lattice with the LQA themselves as lattice elements? In the following we will introduce and discuss some binary operations on LQA and see whether they can serve as the supremum and infimum operations of the wanted lattice. First we review two operations, which were defined in Ying (2000b).

Definition 2.10. Let $\mathscr{R}_1 = (Q_1, I_1, T_1, \Delta_1)$ and $\mathscr{R}_2 = (Q_2, I_2, T_2, \Delta_2)$ be two LQA defined on (l, Σ). The product $\mathscr{R}_1 \times \mathscr{R}_2$ of $\mathscr{R}_1$ and $\mathscr{R}_2$ is also a LQA defined on (l, Σ), called the product automaton, with

$\mathscr{R}_1 \times \mathscr{R}_2 = \mathscr{R}_3 = (Q_3, I_3, T_3, \Delta_3)$, where

$$Q_3 = Q_1 \times Q_2$$
$$I_3 = I_1 \times I_2$$
$$T_3 = T_1 \times T_2$$
$$\Delta_3 = \Big\{\delta(q, x, q') \mid q = (q_1, q_2); q' = (q_1', q_2'); q_1, q_1' \in Q_1; q_2, q_2' \in Q_2;$$
$$x, y \in \Sigma, \delta(q, x, q') = \delta_{R_1}(q_1, x, q_1') \bigcap \delta_{R_2}(q_2, x, q_2') \neq 0\Big\}$$

Definition 2.11. Let $\mathscr{R}_1 = (Q_1, I_1, T_1, \Delta_1)$ and $\mathscr{R}_2 = (Q_2, I_2, T_2, \Delta_2)$ be two LQA defined on (l, Σ), where $Q_1 \cap Q_2 = \oslash$. The sum $\mathscr{R}_1 + \mathscr{R}_2$ of $\mathscr{R}_1$ and $\mathscr{R}_2$ is also a LQA defined on (l, Σ), called the sum automaton, with

$$\mathscr{R}_1 + \mathscr{R}_2 = \mathscr{R}_3 = (Q_3, I_3, T_3, \Delta_3), \text{ where}$$
$$Q_3 = Q_1 \bigcup Q_2$$
$$I_3 = I_1 \bigcup I_2$$
$$T_3 = T_1 \bigcup T_2$$
$$\Delta_3 = \Delta_1 \bigcup \Delta_2$$

Theorem 2.12. (Ying, 2000b). *For any $s \in \Sigma^*$*

$$\text{Accept}_A(\mathscr{R}_1 \times \mathscr{R}_2, s) \subseteq \text{Accept}_A(\mathscr{R}_1, s) \bigcap \text{Accept}_A(\mathscr{R}_2, s)$$

Theorem 2.13. (Ying, 2000b). *For any $s \in \Sigma^*$,*

$$\text{Accept}_A(\mathscr{R}_1 + \mathscr{R}_2, s) = \text{Accept}_A(\mathscr{R}_1, s) \bigcup \text{Accept}_A(\mathscr{R}_2, s)$$

So, the LQA of type A form a upper semilattice with respect to acceptance degree and sum operation. But they do not form a lower semilattice.

We will prove similar results for LQA of type B. In order to do that, we need a lemma.

Lemma 2.14. *Assume all $a_{i,j}$ are elements of the same lattice, then*

$$\bigcup_{\substack{j_1, j_2, \ldots, j_n \text{ is an ordered} \\ n-\text{tuple of numbers from } (1,2,\ldots,m)}} \left(a_{1,j_1} \bigcap a_{2,j_2} \bigcap \cdots \bigcap a_{n,j_n}\right) \subseteq \bigcap_{i=1}^{n} \bigcup_{j=1}^{m} a_{i,j}$$

Proof: Since for an arbitary i,

$$a_{i,j_i} \subseteq \bigcup_{j=1}^{m} a_{i,j}$$

therefore, for any (fixed) combination of $(j_1, j_2, \ldots, j_n)$,

$$\left(a_{1,j_1} \bigcap a_{2,j_2} \bigcap \cdots \bigcap a_{n,j_n}\right) \subseteq \bigcap_{i=1}^{n} \bigcup_{j=1}^{m} a_{i,j}$$

Take the union over all possible combinations of $(j_1, j_2, \ldots, j_n)$ and then we get the lemma proved. □

Theorem 2.15. *For any* $s \in \Sigma^*$

$$\text{Accept}_{\text{B}}(\mathscr{R}_1 \times \mathscr{R}_2, s) \subseteq \text{Accept}_{\text{B}}(\mathscr{R}_1, s) \bigcap \text{Accept}_{\text{B}}(\mathscr{R}_2, s)$$

Proof: Let $s = x_1 x_2 \ldots x_n$

$$\begin{aligned}
&\text{Accept}_{\text{B}}(\mathscr{R}_1 \times \mathscr{R}_2, s)\\
&= \bigcap_{i=0}^{n-1} \bigcup_{\substack{w_1 \in T(R_1,s),\\ w_2 \in T(R_2,s)}} \left[\delta_{R_1}(q_{w_1,i}, x_{i+1}, q_{w_1,i+1}) \bigcap \delta_{R_2}(q_{w_2,i}, x_{i+1}, q_{w_2,i+1})\right]
\end{aligned}$$

with help of Lemma 2.14

$$\begin{aligned}
&\subseteq \bigcap_{i=0}^{n-1} \left\{\left[\bigcup_{w_1 \in T(R_1,s)} \delta_{R_1}(q_{w_1,i}, x_{i+1}, q_{w_1,i+1})\right] \bigcap \left[\bigcup_{w_2 \in T(R_2,s)} \delta_{R_2}(q_{w_2,i}, x_{i+1}, q_{w_2,i+1})\right]\right\}\\
&= \left[\bigcap_{i=0}^{n-1} \bigcup_{w_1 \in T(R_1,s)} \delta_{R_1}(q_{w_1,i}, x_{i+1}, q_{w_1,i+1})\right] \bigcap \left[\bigcap_{i=0}^{n-1} \bigcup_{w_2 \in T(R_2,s)} \delta_{R_2}(q_{w_2,i}, x_{i+1}, q_{w_2,i+1})\right]\\
&= \text{Accept}_{\text{B}}(\mathscr{R}_1, s) \bigcap \text{Accept}_{\text{B}}(\mathscr{R}_2, s).
\end{aligned}$$

□

Theorem 2.16. *For any* $s \in \Sigma^*$,

$$\text{Accept}_{\text{B}}(\mathscr{R}_1, s) \bigcup \text{Accept}_{\text{B}}(\mathscr{R}_2, s) \subseteq \text{Accept}_{\text{B}}(\mathscr{R}_1 + \mathscr{R}_2, s)$$

Proof: Let $s = x_1 x_2 \ldots x_n$. Use a similar idea of the proof procedure of Theorem 2.15 we have:

$$\begin{aligned}
&\text{Accept}_{\text{B}}(\mathscr{R}_1, s) \bigcup \text{Accept}_{\text{B}}(\mathscr{R}_2, s)\\
&= \left[\bigcap_{i=0}^{n-1} \bigcup_{w_1 \in T(R_1,s)} \delta_{R_1}(q_{w_1,i}, x_{i+1}, q_{w_1,i+1})\right] \bigcup \left[\bigcap_{i=0}^{n-1} \bigcup_{w_2 \in T(R_2,s)} \delta_{R_2}(q_{w_2,i}, x_{i+1}, q_{w_2,i+1})\right]\\
&\subseteq \bigcap_{\substack{i=0,1,\ldots,n-1\\ j=0,1,\ldots,n-1}} \left\{\left[\bigcup_{w_1 \in T(R_1,s)} \delta_{R_1}(q_{w_1,i}, x_{i+1}, q_{w_1,i+1})\right] \bigcup \left[\bigcup_{w_2 \in T(R_2,s)} \delta_{R_2}(q_{w_2,j}, x_{j+1}, q_{w_2,j+1})\right]\right\}\\
&\subseteq \bigcap_{i=0,1,\ldots,n-1} \left\{\left[\bigcup_{w_1 \in T(R_1,s)} \delta_{R_1}(q_{w_1,i}, x_{i+1}, q_{w_1,i+1})\right] \bigcup \left[\bigcup_{w_2 \in T(R_2,s)} \delta_{R_2}(q_{w_2,i}, x_{i+1}, q_{w_2,i+1})\right]\right\}\\
&= \text{Accept}_{\text{B}}(\mathscr{R}_1 + \mathscr{R}_2, s)
\end{aligned}$$

□

But, in order to justify their role in defining a lattice of LQA, we should be able to prove additionally the supremum and infimum properties of these two

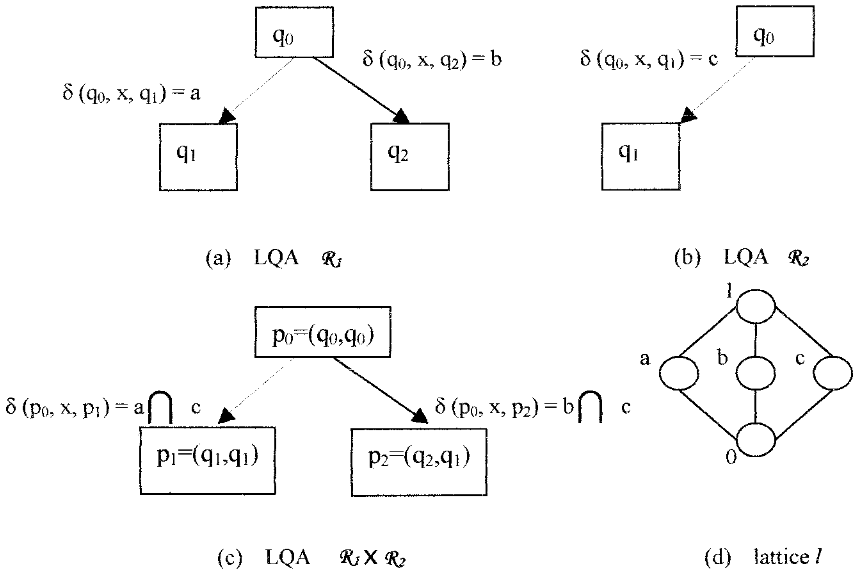

Fig. 2. The product automaton.

operations. Unfortunately, apart from the sum operation of type A, all other three operations (product operation of types A and B, sum operation of type B) do not have this property. For product operation of types A and B, we can convince us about this conclusion from the following example.

Example 2.17. Let $\mathscr{R}_1 = (\{q_0, q_1, q_2\}, \{q_0\}, \{q_1, q_2\}, \{\delta_1(q_0, x, q_1) = a, \delta_1(q_0, x, q_2) = b\})$ and $\mathscr{R}_2 = (\{q_0, q_1\}, \{q_0\}, \{q_1\}, \{\delta_1(q_0, x, q_1) = c\})$ be two LQA. Then according to definition $\mathscr{R}_1 \times \mathscr{R}_2 = (\{(q_0, q_0), (q_0, q_1), (q_1, q_0), (q_1, q_1), (q_2, q_0), (q_2, q_1)\}, \{(q_0, q_0)\}, \{(q_1, q_1), (q_2, q_1)\}, \{\delta \times ((q_0, q_0), x, (q_1, q_1)) = a \bigcap c, \delta \times ((q_0, q_0), x, (q_2, q_1)) = b \bigcap c\})$. If the lattice $l = (\{a, b, c, 1, 0\}, \{0 \subseteq a, b, c \subseteq 1\}, 0, 1)$ (see Fig. 2(d)), then it is Accept $(\mathscr{R}_1 \times \mathscr{R}_2, x) = (a \bigcap c) \bigcup (b \bigcap c) = 0 \subseteq c = (a \bigcup b) \bigcap c = \text{Accept}(\mathscr{R}_1, x) \bigcap \text{Accept}(\mathscr{R}_2, x)$ and $0 \neq c$. This shows that $\mathscr{R}_1 \times \mathscr{R}_2$ is not the largest lower bound of $\mathscr{R}_1$ and $\mathscr{R}_2$, in the sense of acceptance degree.

Note that the conclusion obtained by this example is not only valid for type A, but also for type B LQA, because the two types of LQA do not differ from each other with respect to this example.

More generally, it is possible that $\mathscr{R}_1$ and $\mathscr{R}_2$ have no states in common and thus the intersection automaton is hard to define. In this case we may find examples, which show that the intersection is not the largest lower bound of $\mathscr{R}_1$

and $\mathscr{R}_2$ in the sense of acceptance degree. This is an evidence that we should consider modifying the definition of quantum automaton.

In order to find a possible solution for this problem, let us have a closer look at the source of difficulty we have met. First, the old definition of sum and product is based on structure operation on the two operand automata. This definition has a syntactic character and is not suitable for defining automata intersection based on acceptance degree, which is essentially a semantic affair. Second, the old definition of sum and product is based on nondeterministic quantum automata, which have the difficulty of not following the distributive rule of lattice operation. Therefore, our new idea is first to try to transform the quantum automata into deterministic form and then try to find a suitable form of automata operation.

Thus, the first problem we now meet is the question of whether there exists always an equivalent deterministic LQA for each nondeterministic LQA? If the answer is yes, then how do we calculate it? It is well known that each classical nondeterministic automaton has an equivalent deterministic one. But this is not necessary true for LQA.

Definition 2.18. An LQA $\mathscr{R}$ is called proper nondeterministic if it is nondeterministic and there exists no corresponding deterministic automaton $\mathscr{R}'$ such that $\mathscr{R}$ and $\mathscr{R}'$ are semantically equivalent. That is:

$$\text{is}\,(\mathscr{R}', \text{proper nondeterministic}) \equiv \sim\exists\,\mathscr{R}', \text{is}(\mathscr{R}', \text{deterministic}), \forall s \in \Sigma^*,$$

$$\text{Accept}(\mathscr{R}, s) = \text{Accept}(\mathscr{R}', s)$$

Theorem 2.19. *There exist proper nondeterministic LQA.*

Proof: Consider the LQA $\mathscr{R} = (\{q_0, q_1, q_2, q_3\}, \{q_0\}, \{q_2, q_3\}, \{\delta(q_0, x, q_1) = a, \delta(q_1, y, q_2) = a, \delta(q_0, x, q_3) = b\})$, which is nondeterministic. See Fig. 3(a). The lattice used is $l = (\{a, b, 0, 1\}, \{a \subseteq 1, b \subseteq 1, 0 \subseteq a, 0 \subseteq b\}, 0, 1)$. See Fig. 3(b).

For both types A and B, the language accepted by $\mathscr{R}$ is $\{(xy, a), (x, b)\}$. If there were a deterministic LQA $\mathscr{R}'$ such that $\mathscr{R} \approx \mathscr{R}'$, then because $\mathscr{R}'$ is deterministic, it has only one initial state. Let's call it q_0'. Since $\mathscr{R}'$ must accept the sentence x to the degree b, there must be a terminating state q_1', and the transition $\delta(q_0', x, q_1') = b$. q_1' should not coincide with q_0' because otherwise there would be a loop and sentences of arbitrary length would be accepted. On the other hand, since $\mathscr{R}'$ must accept the sentence xy to the degree a, there must be another terminating state q_2', which is also a state separated from q_0' and q_1'. Assume $\delta(q_1', y, q_2') = g$, where the value g is to be determined. The following equation should be satisfied:

$$b \bigcap g = a \bigcap a = a$$

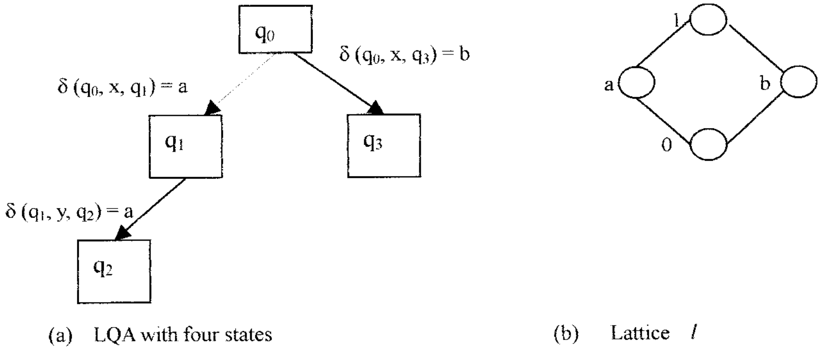

Fig. 3. Proper nondeterministic LQA.

But from Fig. 3(b) it is easy to see that this equation can never be fulfilled. This fact refutes the possibility of existence of an equivalent deterministic $\mathscr{R}'$. □

Note that the existence of proper nondeterministic LQA may be a characteristic property of quantum automata versus the traditional ones.

Thus we have to modify our definition of LQA and to define a new type, the type C of LQA with the hope that this new kind of LQA will help us to find an equivalent deterministic LQA for each nondeterministic one and thus to enable the construction of the lattice theoretic union and intersection operations.

Definition 2.20. A lattice valued quantum automaton of type C defined on (l, Σ) is a quadruple $\mathscr{R} = (Q, I, T, \{\delta(q, x, q', \text{out}) > 0 | x \in \Sigma; \text{out} \in \Sigma \cup \{\varepsilon\}; q, q' \in Q\})$ where Q is the set of states, $I \subseteq Q$ is the set of initial states, $T \subseteq Q$ is the set of terminating states. The notation "out" is either a symbol of Σ or the empty symbol ε. Each $\delta(q, x, q', \text{out})$ is an element of the lattice l, which means that portion of the degree of accepting the input symbol x, which is assigned to the case that the next input symbol $y \in \Sigma$ (if out $= y$) or the total acceptance degree of this transition (if out $=$ empty and q' is a derminating state), and transforming the state from q to q'. The lattice union of $\delta(q, x, q', y)$ for all possible y is then the general degree of accepting the input symbol x.

The new definition is a refinement of the old one. It divides the acceptance degree to different exits of the fan out of the state q'.

An additional advantage of this definition is the ability of defining partial acceptance.

Definition 2.21. Let $w = q_0\, x_1\, q_1 \ldots q_{n-1}\, x_n\, q_n$ be a path of the LQA $\mathscr{R}$ where q_0 is an initial state and q_n is a terminating state. Then each string $s = x_0\, x_1 \ldots x_i$, $1 \leq i \leq n$, is partially accepted by $\mathscr{R}$. We use the notation P-Accept $(\mathscr{R}, s)$ to

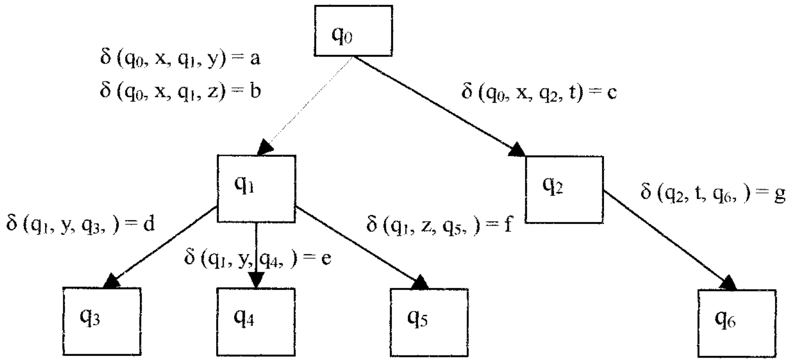

Fig. 4. LQA of type C.

denote partial acceptance. A partially accepted string is (wholly) accepted by a LQA if and only if $i = n$.

Example 2.22. Let $\Sigma = \{x, y, z, t\}$, $\mathscr{R} = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{q_0\}, \{q_3, q_4, q_5, q_6\}, \{\delta(q_0, x, q_1, y) = a, \delta(q_0, x, q_1, z) = b, \delta(q_1, y, q_3,) = d, \delta(q_1, y, q_4,) = e, \delta(q_1, z, q_5,) = f, \delta(q_0, y, q_2, t) = c, \delta(q_2, y, q_6,) = g\})$. See Fig. 4.

In this example, we have P-Accept$_c(\mathscr{R}, x) = a \bigcup b \bigcup c$, Accept$_c(\mathscr{R}, xy) = a \bigcap (d \bigcup e)$, Accept$_c(\mathscr{R}, xz) = b \bigcap f$, Accept$_c(\mathscr{R}, xt) = c \bigcap g$.

Here it is easy to see that if $a = b$ (i.e. the state transitions $(q_1 \rightarrow q_3, q_1 \rightarrow q_4$ and $q_1 \rightarrow q_5$ share the same "heritage" from the previous transition $q_0 \rightarrow q_1)$ then the resulting acceptance degrees coincide with those calculated with our old definition given in last section.

Now we will give an algorithm for transforming any nondeterministic LQA into an equivalent deterministic LQA. This algorithm is based on a classical one (e.g. see (Hopcroft and Ullman, 1979)) and modified according to the characteristics of our LQA definition.

Algorithm 2.23. *(Delete nondeterminacy)*

1. *Given an LQA* $\mathscr{R} = (Q, I, T, \triangle)$ *of type C.*
2. *Construct* $Q' = \wp(Q) = \{S\}$, *where* $\wp$ *means power set.*
3. *Let* $I' = \{I\}$
4. *Let* $T' = \{S | S \in Q', S \cap T \neq \emptyset\}$

5. *Let* $\varDelta' = \{\varDelta'_1 \cup \varDelta'_2 | x, y \in \Sigma\}$, *where*

$$\varDelta'_1 = \left\{\delta(\mathsf{S}, \mathsf{x}, \mathsf{S}', \mathsf{y}) = \bigcup_i \{\mathsf{a}_i | \exists\, \mathsf{q} \in \mathsf{S}, \mathsf{q}' \in \mathsf{S}', \delta(\mathsf{q}, \mathsf{x}, \mathsf{q}', \mathsf{y}) = \mathsf{a}_i \in \varDelta\}\right\}$$

$$\varDelta'_2 = \left\{\delta(\mathsf{S}, \mathsf{x}, \mathsf{S}',) = \bigcup_i \{\mathsf{a}_i | \exists\, \mathsf{q} \in \mathsf{S}, \mathsf{q}' \in \mathsf{S}', \mathsf{q}' \in \mathsf{T}, \delta(\mathsf{q}, \mathsf{x}, \mathsf{q}',) = \mathsf{a}_i \in \varDelta\}\right\}$$

Note that $\bigcup_i$ is a lattice union operation while $\cup$ is set union operation.

Proposition 2.24. *The LQA* $\mathscr{R}' = (Q', I', T', \varDelta')$ *is equivalent to* $\mathscr{R}$ *with respect to accepting strings from* Σ^*.

Proof: Use mathematical induction. □

In the sequel, we consider the process of producing a deterministic product LQA of type C by using the above algorithm.

Definition 2.25. Let $\mathscr{R}_1 = (Q_1, I_1 = \{q_{1,0}\}, T_1, \varDelta_1)$ and $\mathscr{R}_2 = (Q_2, I_2 = \{q_{2,0}\}, T_2, \varDelta_2)$ be two deterministic LQA of type C on (l, Σ). The product $\mathscr{R}_1 \times \mathscr{R}_2$ of $\mathscr{R}_1$ and $\mathscr{R}_2$ is also a deterministic LQA of type C defined on (l, Σ), called the product automation, with

$$\mathscr{R}_1 \times \mathscr{R}_2 = \mathscr{R}_3 = (\mathrm{Q}_3, \mathrm{I}_3, \mathrm{T}_3, \varDelta_3), \text{ where}$$

$$Q_3 = Q_1 \times Q_2$$

$$I_3 = I_1 \times I_2 (= (q_{1,0}, q_{2,0}))$$

$$I_3 = I_1 \times T_2$$

$$\varDelta_3 = \{\delta(q, x, q', y) | q = (q_1, q_2); q' = (q'_1, q'_2); q_1, q'_1 \in Q_1; q_2, q'_2 \in Q_2; x, y \in \Sigma, \delta(q, x, q', y) = \delta_{R_1}(q_1, x, q'_1, y) \bigcap \delta_{R_1}(q_2, x, q'_2, y) \neq 0\} \bigcup \{\delta(q, x, q',) \mid q = (q_1, q_2); q' = (q'_1, q'_2); q_1 \in Q_1; q'_1 \in T_1; q_2 \in Q_2; q'_2 \in T_2; x \in \Sigma, \delta(q, x, q',) = \delta_{R_1}(q_1, x, q'_1,) \bigcap \delta_{R_2}(q_2, x, q'_2,) \neq 0\}$$

Proposition 2.26. $\forall \mathbf{s} \in \Sigma^*$, $\mathrm{Accept_c}(\mathscr{R}_1 \times \mathscr{R}_2, s) = \mathrm{Accept_c}(\mathscr{R}_1, s) \bigcap \mathrm{Accept_c}(\mathscr{R}_2, s)$

Proof: Assume $s = x_1 x_2 \ldots x_n$, $w_1 = q_{1,0}\, x_1\, q_{1,1} \ldots q_{1,n-1}\, x_n\, q_{1,n}$ and $w_2 = q_{2,0}\, x_1\, q_{2,1} \ldots q_{2,n-1}\, x_n\, q_{2,n}$ are the corresponding paths in $\mathscr{R}_1$ and $\mathscr{R}_2$ respectively.

First we consider the case of $n = 1$. Since both $\mathscr{R}_1$ and $\mathscr{R}_2$ are deterministic, we have

$$\begin{aligned}&\text{Accept}_c(\mathscr{R}_1, s) \bigcap \text{Accept}_c(\mathscr{R}_2, s)\\&= \delta_{w_1}(q_{1,0}, x_1, q_{1,1,}) \bigcap \delta_{w_2}(q_{2,0}, x_1, q_{2,1,}) = \text{Accept}_c(\mathscr{R}_1 \times \mathscr{R}_2, s)\end{aligned}$$

where $q_{1,0} \in I_1;\ q_{1,1} \in T_1; q_{2,0} \in I_2; q_{2,1} \in T_2;$
For an arbitrary $n > 1$ we have:

$$\begin{aligned}&\text{Accept}_c(\mathscr{R}_1, s) \bigcap \text{Accept}_c(\mathscr{R}_2, s)\\&= \left[\bigcap_{i=0}^{n-2} \delta_{w_1}(q_{1,i}, x_{i+1}, q_{1,i+1}, x_{i+2})\right] \bigcap \delta_{w_1}(q_{1,n-1}, x_n, q_{1,n},) \bigcap\\&\quad \left[\bigcap_{i=0}^{n-2} \delta_{w_2}(q_{2,i}, x_{i+1}, q_{2,i+1}, x_{i+2})\right] \bigcap \delta_{w_2}(q_{2,n-1}, x_n, q_{2,n},)\\&= \left[\bigcap_{i=0}^{n-2} \left[\delta_{w_1}(q_{1,i}, x_{i+1}, q_{1,i+1}, x_{i+2}) \bigcap \delta_{w_2}(q_{2,i}, x_{i+1}, q_{2,i+1}, x_{i+2})\right]\right]\\&\quad \bigcap \left[\delta_{w_1}(q_{1,n-1}, x_n, q_{1,n},) \bigcap \delta_{w_2}(q_{2,n-1}, x_n, q_{2,n},)\right]\\&= \text{Accept}_c(\mathscr{R}_1 \times \mathscr{R}_2, s)\end{aligned}$$

□

Therefore, the LQA of type C form a lower semilattice with respect to acceptance degree and the product operation $\times$. But they do not form an upper lattice due to Theorem 2.16 Taking the notice after Theorem 2.13 in consideration, we see there is a symmetry between types A and C automata. In order to get a true lattice of LQA, we need to find another way of defining LQA lattice.

3. LAT (l, Σ, Θ): A LATTICE OF LQA

The reason that the LQA described in last section do not form a lattice is the grain size of their equivalent groups, which is too large. LQA accepting input strings to the same degree may have quite different structures. In this section, we will limit the variety of LQA structures a little bit and will see that this limitation helps to develop a true lattice of quantum automata. As the first step, we will modify our definition slightly such that the states the LQA may take form a fixed state space. We rephrase the definition of a LQA with this modification.

Definition 3.1. Let $l = (L, \subseteq 0, 1)$ be a lattice, Σ be a finite alphabet, called the input alphabet, Θ be a finite set of states, called the state space. An LQA $\mathscr{R}$ defined on (l, Σ, Θ) is a quadruple $\mathscr{R} = (Q, I, T, \triangle)$, where $Q \subseteq \Theta$ is a set of states, $I \subseteq Q$ is a set of initial states, $T \subseteq Q$ is a set of terminating states, $\triangle$ is a set of l valued predicates defined on $Q \times \Sigma \times Q$: for each $q_1, q_2 \in Q$ and $x \in \Sigma$, $\delta(q_1, x, q_2) \in \triangle$ is an element of l. Note that in $\triangle$ only those $\delta(q_1, x, q_2)$, which are not equal to 0 (least element of l) are listed in $\triangle \cdot \delta(q_1, x, q_2)$ is called the acceptance degree of x by $\mathscr{R}$ that the state q_1 is transferred to q_2 when the symbol x is inputted. The LQA are classified in type A LQA and type B LQA in Definition 2.4 according to the way the acceptance degree is calculated.

Apart from introducing the state space Θ, almost all other definitions about LQA in the last section are kept unchanged here. Except that we will follow another way of defining a lattice of quantum automata. Remember in last section, we started from defining the partial order $\subseteq'$ and then tried to use it to define the supremum and infimum operation for the wanted lattice of quantum automata. In this section, we will go the opposite way. We first define the two lattice operations $\cup$, $\cap$ for the wanted automata lattice and prove that they fulfil the rules given in Definition 1.2 Then we define a partial order $\subseteq''$ based on these two operations. Finally, we will prove that the new partial order is a suborder of the old one ($\subseteq'$). That means, if $\mathscr{R}_1 \subseteq'' \mathscr{R}_2$ then it is also $\mathscr{R}_1 \subseteq' \mathscr{R}_2$, but not vice versa. During the process of developing the theory, we will give proofs both for types A and B LQA.

Definition 3.2. Let $\mathscr{R}_1 = (Q_1, I_1, T_1, \triangle_1)$ and $\mathscr{R}_2 = (Q_2, I_2, T_2, \triangle_2)$ be two LQA on (l, Σ, Θ). The intersection $\mathscr{R}_1 \cap \mathscr{R}_2$ of $\mathscr{R}_1$ and $\mathscr{R}_2$ is also a LQA defined on (l, Σ, Θ), called the intersection automation, with

$$\mathscr{R}_1 \cap \mathscr{R}_2 = \mathscr{R}_3 = (Q_3, I_3, T_3, \triangle_3), \text{ where}$$

$$Q_3 = Q_1 \bigcap Q_2$$

$$I_3 = I_1 \bigcap I_2$$

$$T_3 = T_1 \bigcap T_2$$

$$\triangle_3 = \Big\{\delta(q, x, q') \mid q, q' \in Q_3; x \in \Sigma_1 \bigcap \Sigma_2, \delta(q, x, q') = \delta_{R_1}(q, x, q') \bigcap \delta_{R_2}(q, x, q') \neq 0\Big\}$$

Definition 3.3. Let $\mathscr{R}_1 = (Q_1, I_1, T_1, \triangle_1)$ and $\mathscr{R}_2 = (Q_2, I_2, T_2, \triangle_2)$ be two LQA on (l, Σ, Θ). The union $\mathscr{R}_1 \cup \mathscr{R}_2$ of $\mathscr{R}_1$ and $\mathscr{R}_2$ is also a LQA defined on (l, Σ, Θ), called the union automation, with

$$\mathscr{R}_1 \cup \mathscr{R}_2 = \mathscr{R}_3 = (Q_3, I_3, T_3, \triangle_3), \text{ where}$$

$$\begin{aligned} Q_3 &= Q_1 \bigcup Q_2 \\ I_3 &= I_1 \bigcup I_2 \\ T_3 &= T_1 \bigcup T_2 \\ \varDelta_3 &= \Big\{\delta(q, x, q')|q, q' \in Q_3; x \in \Sigma, \delta(q, x, q') \\ &= \delta_{R_1}(q, x, q') \bigcup \delta_{R_2}(q, x, q') \neq 0\Big\} \end{aligned}$$

Note that a union automation is in general not equal to a sum automation defined in last section, since in Definition 3.3, we have removed the limitation that $Q_1 \bigcap Q_2 = \oslash$. In addition, these two operations follow the rules of lattice operation defined in first section, as the two following theorems illustrate.

The reader is reminded once again that $\bigcup$ and $\bigcap$ are lattice operations for the lattice l, $\cup$ and $\cap$ are lattice operations for our automata lattice, while $\bigcup$ and $\bigcap$ are set operations.

Theorem 3.4. *Let $\mathscr{R}_1$, $\mathscr{R}_2$, and $\mathscr{R}_3$ be three LQA defined on (l, Σ, Θ), then:*

1. $\mathscr{R}_1 \cap \mathscr{R}_2 = \mathscr{R}_2 \cap \mathscr{R}_1$
2. $\mathscr{R}_1 \cup \mathscr{R}_2 = \mathscr{R}_2 \cup \mathscr{R}_1$
3. $(\mathscr{R}_1 \cap \mathscr{R}_2) \cap \mathscr{R}_3 = \mathscr{R}_1 \cap (\mathscr{R}_2 \cap \mathscr{R}_3)$
4. $(\mathscr{R}_1 \cup \mathscr{R}_2) \cup \mathscr{R}_3 = \mathscr{R}_1 \cup (\mathscr{R}_2 \cup \mathscr{R}_3)$

Proof: From the definition of intersection automaton and union automaton we know that the validity of the first two points of this theorem are true. We will prove only the third point. Let $\mathscr{R}_i = (Q_i, I_i, T_i, \varDelta_i)$, $i = 1, 2, 3$ by arbitrary LQA defined on (l, Σ, Θ), then

$$\begin{aligned} &(\mathscr{R}_1 \cap \mathscr{R}_2) \cap \mathscr{R}_3 \\ &= \Big(\Big(Q_1 \bigcap Q_2\Big) \bigcap Q_3, \Big(I_1 \bigcap I_2\Big) \bigcap I_3, \Big(T_1 \bigcap T_2\Big) \bigcap T_3, \Big\{\delta(q, x, q') \mid x \in \\ &\quad \Sigma, q, q' \in \Big(Q_1 \bigcap Q_2\Big) \bigcap Q_3, \delta(q, x, q') \\ &= \Big(\delta_{R_1}(q, x, q') \bigcap \Big(\delta_{R_2}(q, x, q')\Big) \bigcap \delta_{R_3}(q, x, q')\Big\}\Big) \\ &= \Big(Q_1 \bigcap \Big(Q_2 \bigcap Q_3\Big), I_1 \bigcap \Big(I_2 \bigcap I_3\Big), T_1 \bigcap \Big(T_2 \bigcap T_3\Big), \\ &\quad \times \Big\{\delta(q, x, q') \mid x \in \Sigma, q, q' \in Q_1 \bigcap \Big(Q_2 \bigcap Q_3\Big), \delta(q, x, q') \\ &= \delta_{R_1}(q, x, q') \bigcap \Big(\delta_{R_2}(q, x, q') \bigcap \delta_{R_3}(q, x, q')\Big)\Big\}\Big) \\ &= \mathscr{R}_1 \cap (\mathscr{R}_2 \cap \mathscr{R}_3) \end{aligned}$$

□

Theorem 3.5. *Let $\mathscr{R}_1$ and $\mathscr{R}_2$ be arbitrary LQA defined on* (l, Σ, Θ), *then:*

1. $(\mathscr{R}_1 \cap \mathscr{R}_2) \cup \mathscr{R}_2 = \mathscr{R}_2$
2. $(\mathscr{R}_1 \cup \mathscr{R}_2) \cap \mathscr{R}_2 = \mathscr{R}_2$

Proof: We prove only the first assertion. Let $\mathscr{R}_i = (Q_i, I_i, T_i, \varDelta_i)$, $i = 1, 2$ be arbitrary LQA defined on (l, Σ, Θ), then

$$(\mathscr{R}_1 \cap \mathscr{R}_2) \cup \mathscr{R}_2 = \Big(\Big(Q_1 \bigcap Q_2\Big) \bigcup Q_2, \Big(I_1 \bigcap I_2\Big) \bigcup I_2, \Big(T_1 \bigcap T_2\Big) \bigcup T_2,$$

$$\Big\{\delta(q, x, q') \mid x \in \Sigma, q, q' \in \Big(Q_1 \bigcap Q_2\Big) \bigcup Q_2, \delta(q, x, q')$$

$$= \Big(\delta_{R_1}(q, x, q') \bigcap \delta_{R_2}(q, x, q')\Big) \bigcup \delta_{R_2}(q, x, q')\Big\}\Big)$$

$$= \Big(Q_2, I_2, T_2, \Big\{\delta(q, x, q') \mid x \in \Sigma, q, q' \in Q_2, \delta(q, x, q') = \delta_{R_2}(q, x, q')\Big\}\Big)$$

$$= \mathscr{R}_2$$

Similarly, we can prove $(\mathscr{R}_1 \cup \mathscr{R}_2) \cap \mathscr{R}_2 = \mathscr{R}_2$ □

Note that we have used the same symbols for the intersection and union of sets and of automata, respectively. There should be no confusion.

Theorem 3.6. *The lattice-valued finite-state quantum automata defined on* (l, Σ, Θ) form *a lattice.*

Proof: By summarizing the results of Theorems 3.4 and 3.5 we can see that they have fulfilled all rules specified in Definition 1.2. Therefore, these automata form a lattice. We call it Lat (l, Σ, Θ). □

As usually, we can use the two operations $\cap$ and $\cup$ to define the partial order of lattice elements in Lat (l, Σ, Θ).

Definition 3.7. For arbitrary LQA $\mathscr{R}_1$ and $\mathscr{R}_2$, we define

$$\mathscr{R}_1 \subseteq'' \mathscr{R}_2 \text{ if and only if } \big(\mathscr{R}_1 \cap \mathscr{R}_2\big) = \mathscr{R}_1$$

Corollary 3.8. $\subseteq''$ *is a partial order*

Proof: We first have to prove that if $\mathscr{R}_1 \subseteq'' \mathscr{R}_2$ and $\mathscr{R}_2 \subseteq'' \mathscr{R}_3$ then $\mathscr{R}_1 \subseteq'' \mathscr{R}_3$. Assume that $(\mathscr{R}_1 \cap \mathscr{R}_2) = \mathscr{R}_1$ and $(\mathscr{R}_2 \cap \mathscr{R}_3) = \mathscr{R}_2$ then

$$\mathscr{R}_1 \cap \mathscr{R}_3 = \big(\mathscr{R}_1 \cap \mathscr{R}_2\big) \cap \mathscr{R}_3 = \mathscr{R}_1 \cap \big(\mathscr{R}_2 \cap \mathscr{R}_3\big) = \mathscr{R}_1 \cap \mathscr{R}_2 = \mathscr{R}_1$$

Thus the transitivity rule holds. The second thing is to prove that

$$\text{if } \mathscr{R}_1 \subseteq'' \mathscr{R}_2 \text{ and } \mathscr{R}_2 \subseteq'' \mathscr{R}_1 \text{ then } \mathscr{R}_1 = \mathscr{R}_2.$$

This means to prove that if $(\mathscr{R}_1 \cap \mathscr{R}_2) = \mathscr{R}_1$ and $(\mathscr{R}_2 \cap \mathscr{R}_1) = \mathscr{R}_2$ then $\mathscr{R}_1 = \mathscr{R}_2$, which is obvious. □

It is easy to prove (by using duality rule) that Definition 3.7 is equivalent to the following form:

$$\mathscr{R}_1 \subseteq'' \mathscr{R}_2 \text{ if and only if } (\mathscr{R}_1 \cup \mathscr{R}_2) = \mathscr{R}_2$$

We have used the concept of acceptance degree to define a partial order $\subseteq'$ between the lattice valued automata in last section. On the other hand, we have also defined the two lattice operations $\cap$ and $\cup$ for these automata. Now we will inspect the relation between the two partial orders $\subseteq'$ and $\subseteq''$. In order to do that, we need the following theorems for types A and B LQA, respectively.

Theorem 3.9. *For any $s \in \Sigma^*$*

$$\text{Accept}_\text{A}(\mathscr{R}_1 \cap \mathscr{R}_2, s) \subseteq \text{Accept}_\text{A}(\mathscr{R}_1, s) \bigcap \text{Accept}_\text{A}(\mathscr{R}_2, s)$$

Proof: Let $s = x_1 x_2 \cdots x_n$

$$\begin{aligned}
&\text{Accept}_\text{A}\left(\mathscr{R}_1 \cap \mathscr{R}_2, s\right)\\
&= \bigcup_{w \in T(R_1 \bigcap R_2, s)} \bigcap_{i=0}^{n-1} \left(\delta_{R_1}(q_{w,i}, x_{i+1}, q_{w,i+1}) \bigcap \delta_{R_2}(q_{w,i}, x_{i+1}, q_{w,i+1})\right)\\
&= \bigcup_{w \in T(R_1 \cap R_2, s)} \left[\left(\bigcap_{i=0}^{n-1} \delta_{R_1}(q_{w,i}, x_{i+1}, q_{w,i+1})\right) \bigcap \left(\bigcap_{i=0}^{n-1} \delta_{R_2}(q_{w,i}, x_{i+1}, q_{w,i+1})\right)\right]\\
&\subseteq \left[\bigcup_{w \in T(R_1 \cap R_2, s)} \left[\bigcap_{i=0}^{n-1} \delta_{R_1}(q_{w,i}, x_{i+1}, q_{w,i+1})\right]\right] \bigcap \left[\bigcup_{w \in T(R_1 \cap R_2, s)} \left[\bigcap_{i=0}^{n-1} \delta_{R_2}(q_{w,i}, x_{i+1}, q_{w,i+1})\right]\right]\\
&\subseteq \left[\bigcup_{w \in T(R_1, s)} \left[\bigcap_{i=0}^{n-1} \delta_{R_1}(q_{w,i}, x_{i+1}, q_{w,i+1})\right]\right] \bigcap \left[\bigcup_{w \in T(R_2, s)} \left[\bigcap_{i=0}^{n-1} \delta_{R_2}(q_{w,i}, x_{i+1}, q_{w,i+1})\right]\right]\\
&= \text{Accept}_\text{A}(\mathscr{R}_1, s) \bigcap \text{Accept}_\text{A}(\mathscr{R}_2, s)
\end{aligned}$$

□

Theorem 3.10. *For any $s \in \Sigma^*$*

$$\text{Accept}_\text{B}(\mathscr{R}_1 \cap \mathscr{R}_2, s) \subseteq \text{Accept}_\text{B}(\mathscr{R}_1, s) \bigcap \text{Accept}_\text{B}(\mathscr{R}_2, s)$$

Proof: Let $s = x_1 x_2 \cdots x_n$

$$
\begin{aligned}
&\text{Accept}_{\text{B}}(\mathscr{R}_1 \cap \mathscr{R}_2, s)\\
&= \bigcap_{i=0}^{n-1} \bigcup_{w \in T(R_1 \cap R_2, s)} \left[\delta_{R_1}(q_{w,i}, x_{i+1}, q_{w,i+1}) \bigcap \delta_{R_2}(q_{w,i}, x_{i+1}, q_{w,i+1})\right]\\
&\subseteq \bigcap_{i=0}^{n-1} \bigcup_{\substack{w_1 \in T(R_1,s)\\ w_2 \in T(R_2,s)}} \left[\delta_{R_1}(q_{w_1,i}, x_{i+1}, q_{w_1,i+1}) \bigcap \delta_{R_2}(q_{w_2,i}, x_{i+1}, q_{w_2,i+1})\right]\\
&\subseteq \bigcap_{i=0}^{n-1} \left\{\left[\bigcup_{w_1 \in T(R_1,s)} \delta_{R_1}(q_{w_1,i}, x_{i+1}, q_{w_1,i+1})\right] \bigcap \left[\bigcup_{w_2 \in T(R_2,s)} \delta_{R_2}(q_{w_2,i}, x_{i+1}, q_{w_2,i+1})\right]\right\}\\
&= \left[\bigcap_{i=0}^{n-1} \bigcup_{w_1 \in T(R_1,s)} \delta_{R_1}(q_{w_1,i}, x_{i+1}, q_{w_1,i+1})\right] \bigcap \left[\bigcap_{i=0}^{n-1} \bigcup_{w_2 \in T(R_2,s)} \delta_{R_2}(q_{w_2,i}, x_{i+1}, q_{w_2,i+1})\right]\\
&= \text{Accept}_{\text{B}}(\mathscr{R}_1, s) \bigcap \text{Accept}_{\text{B}}(\mathscr{R}_2, s).
\end{aligned}
$$

□

In the proof of both theorems we have made use of Lemma 2.14.

Proposition 3.11. $\subseteq'' \subseteq \subseteq'$

Proof: We prove $\mathscr{R}_1 \subseteq'' \mathscr{R}_2 \to \mathscr{R}_1 \subseteq' \mathscr{R}_2$

If $\mathscr{R}_1 \subseteq'' \mathscr{R}_2$ then $(\mathscr{R}_1 \cap \mathscr{R}_2) = \mathscr{R}_1$. With Theorems 3.9 and 3.10 we know that Accept $(\mathscr{R}_1, s) = \text{Accept}(\mathscr{R}_1 \cap \mathscr{R}_2, s) \subseteq \text{Accept}(\mathscr{R}_1, s) \bigcap \text{Accept}(\mathscr{R}_2, s) \subseteq' \text{Accept}(\mathscr{R}_2, s)$. Thus $\mathscr{R}_1 \subseteq' \mathscr{R}_2$. □

Theorem 3.12. Lat (l, Σ, Θ) *is a bounded lattice.*

Proof: We consider the LQA $\mathscr{R}_{\max} = (\Theta, \Theta, \Theta, \measuredangle_{\max} = \{\delta(q, x, q') = 1 | x \in \Sigma, q, q' \in \Theta)\})$, where 1 is the greatest element of the lattice l. For any LQA $\mathscr{R} = (Q, I, T, \measuredangle)$ of Lat (l, Σ, Θ) we have

1. $\mathscr{R}_{\max} \cup \mathscr{R} = (Q \bigcup \Theta,\ I \cup \Theta, T \bigcup \Theta,\ \{\delta(q, x, q') | x \in \Sigma; q, q' \in Q \bigcup \Theta; \delta(q, x, q') = \delta_{R_{\max}}(q, x, q') \bigcup \delta_{\mathscr{R}}(q, x, q')\}) = (\Theta, \Theta, \Theta, \measuredangle_{\max} = \{\delta(q, x, q') = 1 | x \in \Sigma, q, q' \in \Theta)\}) = \mathscr{R}_{\max}$
2. $\mathscr{R}_{\min} \bigcap \mathscr{R} = (Q \bigcap \Theta, I \bigcap \Theta, T \cap \Theta, \{\delta(q, x, q') | x \in \Sigma; q, q' \in Q \bigcap \Theta; \delta(q, x, q') = \delta_{R_{\max}}(q, x, q') \bigcap \delta_{\mathscr{R}}(q, x, q')\}) = (Q, I, T, \measuredangle) = \mathscr{R}$

Then we consider the LQA $\mathscr{R}_{\min} = (\oslash, \oslash, \oslash, \oslash)$. Similarly, it is easy to prove that the rules

1. $\mathscr{R}_{\min} \cup \mathscr{R} = \mathscr{R}$
2. $\mathscr{R}_{\min} \cap \mathscr{R} = \mathscr{R}_{\min}$

hold. This examination shows that $\mathscr{R}_{\max}$ and $\mathscr{R}_{\min}$ are the greatest and least element of Lat (l, Σ, Θ), respectively. Thus Lat (l, Σ, Θ) is a bounded lattice. □

Corollary 3.13. *For any LQA $\mathscr{R}$ of* Lat (l, Σ, Θ)*, we have the inclusion rule:*

$$\mathscr{R}_{\min} \subseteq'' \mathscr{R} \subseteq'' \mathscr{R}_{\max}$$

Proof: Use Definition 3.7.

Now we can write the bounded lattice Lat (l, Σ, Θ) in the following form:

$$\mathrm{Lat}(l, \Sigma, \Theta) = (\mathrm{LQA}(l, \Sigma, \Theta), \cap, \cup, \mathscr{R}_{\min}, \mathscr{R}_{\max})$$

or also in the form:

$$\mathrm{Lat}(l, \Sigma, \Theta) = (\mathrm{LQA}(l, \Sigma, \Theta), \subseteq'', \mathscr{R}_{\min}, \mathscr{R}_{\max})$$

It is meaningful to discuss the relationship between the basic lattice l and the lattice of LQA defined on (l, Σ, Θ). We have the following. □

Theorem 3.14. Lat (l, Σ, Θ) *is a complete lattice if and only if l itself is a complete lattice.*

Proof: Consider the intersection of an arbitrary number of LQA:

$$\begin{aligned} &\cap_{i=1}^{\infty} \mathscr{R}_i(Q_i, I_i, T_i, \measuredangle_i) \\ &= \mathscr{R}\left(\bigcap_{i=1}^{\infty} Q_i, \bigcap_{i=1}^{\infty} I_i, \bigcap_{i=1}^{\infty} T_i, \left\{\delta(q, x, q') | q, q' \in \bigcap_{i=1}^{\infty} Q_i, x \in \Sigma, \delta(q, x, q')\right.\right. \\ &= \left.\left.\bigcap_{i=1}^{\infty} \delta_{R_i}(q, x, q')\right\}\right) \\ &= \mathscr{R}(Q, I, T, \measuredangle_\infty) = \mathscr{R}_\infty \end{aligned}$$

Note that

1. Since for each i, it is $I_i \subseteq Q_i$, $T_i \subseteq Q_i$, therefore, we have $I \subseteq Q$, $T \subseteq Q$.
2. Since l is a complete lattice, $\delta(q, x, q')$ is also an element of l.

Therefore, $\mathscr{R}_\infty$ is an element of Lat (l, Σ, Θ). Lat (l, Σ, Θ) is a complete lattice.

On the other hand, if Lat (l, Σ, Θ) is a complete lattice, then $\mathscr{R}_\infty$ is an element of Lat (l, Σ, Θ). Thus, $\delta(q, x, q')$ must be also an element of l. This shows l is a complete lattice. □

Theorem 3.15. Lat (l, Σ, Θ) *is a distributive lattice if and only if l itself is a distributive lattice.*

Proof: Let $\mathscr{R}_i = (Q_i, I_i, T_i, \Delta_i)$, $i = 1, 2, 3$ be three LQA defined on (l, Σ, Θ). Assume that l is distributive, then:

$$\begin{aligned}
&\mathscr{R}_1 \cap (\mathscr{R}_2 \cup \mathscr{R}_3)\\
&= \Big(Q_1 \bigcap \Big(Q_2 \bigcup Q_3\Big), I_1 \bigcap \Big(I_2 \bigcup I_3\Big), T_1 \bigcap \Big(T_2 \bigcup T_3\Big),\\
&\quad \Big\{\delta(q, x, q') \mid x \in \Sigma; q, q' \in Q_1 \bigcap \Big(Q_2 \bigcup Q_3\Big); \delta(q, x, q')\\
&= \Big(\delta_{R_1}(q, x, q') \bigcap \Big[\delta_{R_2}(q, x, q') \bigcup \delta_{R_3}(q, x, q')\Big]\Big\}\Big)\\
&= \Big(\Big(Q_1 \bigcap Q_2\Big) \bigcup \Big(Q_1 \bigcap Q_3\Big); \Big(I_1 \bigcap I_2\Big) \bigcup \Big(I_1 \bigcap I_3\Big),\\
&\quad \Big(T_1 \bigcap T_2\Big) \bigcup \Big(T_1 \bigcap T_3\Big), \Big\{\delta(q, x, q') \mid x \in \Sigma; q, q' \in \Big(Q_1 \bigcap Q_2\Big) \bigcup\\
&\quad \Big(Q_1 \bigcap Q_3\Big); \delta(q, x, q')\\
&= \Big[\delta_{R_1}(q, x, q') \bigcap \delta_{R_2}(q, x, q')\Big] \bigcup \Big[\delta_{R_1}(q, x, q') \bigcap \delta_{R_3}(q, x, q')\Big]\Big\}\Big)\\
&= (\mathscr{R}_1 \cap \mathscr{R}_2) \cup (\mathscr{R}_1 \cap \mathscr{R}_3)
\end{aligned}$$

This chain of reasoning can be also done backwards. Therefore $\mathscr{R}_1 \cap (\mathscr{R}_2 \cup \mathscr{R}_3) = (\mathscr{R}_1 \cap \mathscr{R}_2) \cup (\mathscr{R}_1 \cap \mathscr{R}_3)$ if and only if $a \bigcap (b \cup c) = (a \bigcap b) \bigcup (a \bigcap c)$ for any elements a, b, and c of l.

Similarly we can prove that $\mathscr{R}_1 \cup (\mathscr{R}_2 \cap \mathscr{R}_3) = (\mathscr{R}_1 \cup \mathscr{R}_2) \cap (\mathscr{R}_1 \cup \mathscr{R}_3)$ if and only if $a \bigcup (b \bigcap c) = (a \bigcup b) \bigcap (a \bigcup c)$ for any elements a, b, and c of l.

Thus if Lat (l, Σ, Θ) is distributive if and only if l is distributive. □

Theorem 3.16. Lat (l, Σ, Θ) *is a modular lattice if and only if l itself is a modular lattice.*

Proof: Let $\mathscr{R}_i = (Q_i, I_i, T_i, \Delta_i)$, $i = 1, 2, 3$ be three LQA defined on (l, Σ, Θ). Assume that l is modular, then:

$$\begin{aligned}
&\mathscr{R}_1 \cup (\mathscr{R}_2 \cap (\mathscr{R}_1 \cup \mathscr{R}_3))\\
&= \Big(Q_1 \bigcup \Big(Q_2 \bigcap \Big(Q_1 \bigcup Q_3\Big)\Big), I_1 \bigcup \Big(I_2 \bigcap \Big(I_1 \bigcup I_3\Big)\Big), T_1 \bigcup \Big(T_2 \bigcap \Big(T_1 \bigcup T_3\Big)\Big),\\
&\quad \Big\{\delta(q, x, q') | x \in \Sigma; q, q' \in Q_1 \bigcup \Big(Q_2 \bigcap \Big(Q_1 \bigcup Q_3\Big)\Big); \delta(q, x, q')\\
&= \delta_{R_1}(q, x, q') \bigcup \Big[\delta_{R_2}(q, x, q') \bigcap \Big(\delta_{R_1}(q, x, q') \bigcup \delta_{R_3}(q, x, q')\Big)\Big]\Big\}\Big)
\end{aligned}$$

According to Proposition 1.13, it is then

$$= \Big(\Big(Q_1 \bigcup Q_2\Big) \bigcap \Big(Q_1 \bigcup Q_3\Big), \Big(I_1 \bigcup I_2\Big) \bigcap \Big(I_1 \bigcup I_3\Big), \Big(T_1 \bigcup T_2\Big) \bigcap \Big(T_1 \bigcup T_3\Big), \Big\{\delta(q, x, q') | x \in \Sigma; q, q' \in \Big(Q_1 \bigcup Q_2\Big) \bigcap \Big(Q_1 \bigcup Q_3\Big), \delta(q, x, q') = \Big[\delta_{R_1}(q, x, q') \bigcup \delta_{R_2}(q, x, q')\Big] \bigcap \Big[\delta_{R_1}(q, x, q') \bigcup \delta_{R_3}(q, x, q')\Big]\Big\}\Big)$$

$$= (\mathscr{R}_1 \cup \mathscr{R}_2) \cap (\mathscr{R}_1 \cup \mathscr{R}_3)$$

This chain of reasoning can be also done backwards. Therefore $\mathscr{R}_1 \cup (\mathscr{R}_2 \cap (\mathscr{R}_1 \cup \mathscr{R}_3)) = (\mathscr{R}_1 \cup \mathscr{R}_2) \cap (\mathscr{R}_1 \cup \mathscr{R}_3)$ if and only if $a \bigcup (b \bigcap (a \bigcup c)) = (a \bigcup b) \bigcap (a \bigcup c)$ for any elements a, b and c of l.

Thus if Lat (l, Σ, Θ) is modular if and only if l is modular. □

4. CONCLUSION

In this paper, we have reviewed the lattice-valued finite-state quantum automata introduced by Ying (2000a). We have redefined the acceptance degree (recognizability in Ying's paper) and got a new definition of quantum automata. We call the LQA defined in Ying (2000a) as LQA of type A and that in this paper as LQA of type B. Based on the two automata operations $\times$ and $+$ introduced in Ying's paper, we have shown that of the four automata operations (two for type A and two for type B), only one of them follows the law of conservation of acceptance degree. This shows a basic difference between classical automata and quantum ones.

In the effort of studying the structure of the set of LQA defined on the same lattice l and alphabet Σ, we get the conclusion that they do not form a lattice with respect to the acceptance degree. LQA of type A form an upper semilattice with the two operations $\times$ and $+$. LQA of type B don't. The source of trouble is the nondeterminacy. While trying to transform all nondeterministic LQA in equivalent deterministic form, we met the problem that it is not always possible to transform a nondeterministic LQA (no matter it is of type A or type B) into a deterministic one. We have proved this fact rigorously. This may be a second essential difference between classical and quantum automata.

We then defined another new type of LQA, the LQA of type C. We have proved that each (nondeterministic) LQA of type C can be transformed to an equivalent deterministic one. With this result we were able to show that the LQA of type C form a lower semilattice with respect to the acceptance degree. This is another symmetry between different definitions of LQA acceptance degree.

The introduction of type B LQA has the advantage that it considers the procedure of accepting a string a by a nondeterministic LQA not as a mix of multiple single path acceptance procedures, but as an integrated procedure, which proceeds concurrently in multiple directions. We call it distributed acceptance. This definition provides a unified way of processing deterministic and nondeterministic LQA. Its second advantage is to consider the acceptance of a string $s = x_1 x_2 \ldots x_n$ as an evolutional procedure. Intuitionally, the degree of accepting $x_1 x_2 \ldots x_j$ for any $2 \leq j \leq n$ is "less or equal" (in the sense of the partial order of the lattice) to that of accepting $x_1 x_2 \ldots x_{j-1}$. This evolutionary character of acceptance degree calculation is further improved by introducing LQA of type C, where the acceptance degree of a transition is divided in portions and assigned to different exit directions.

Finally, by reducing the size of equivalence group of LQA and concentrating on the structure properties of LQA we have succeeded in developing a true lattice Lat (l, Σ, Θ) of LQA, where the most interesting thing is to discover the relationship between the original lattice l and the lattice of LQA. We have proved that the validity of many properties of the lattice Lat (l, Σ, Θ), such as whether it is complete, distributive or modular, depend on the corresponding properties of the original lattice. This may be the third essential difference between classical automata and quantum ones.

Sine we have introduced lattices and semilattices for the quantum automata, the next work may be to explore further the structure of these lattices and the relations between the structure of original lattice l and that of the quantum lattice.

ACKNOWLEDGMENTS

This work was partially supported by CAS project of brain and mind science, Pre-973 project 2001CCA03000, 863 project, the innovation foundation of IOM, AMSS, and ICT projects, NSFC foundation (Grant No. 69733020). The authors thank Prof Mingsheng Ying sincerely. The work on lattice-valued quantum automata described in this paper is based on his inspiration and benefits also from his valuable comments and suggestions. We also thank Prof Hong Zhu and Prof Jigui Sun for their kindly help.

REFERENCES

Cohn, P. M. (1981). *Universal Algebra*, Reidel, Dordrecht.

Deutsch, D. (1985). Quantum theory, the Church Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London 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. Beltramettiand and B. C. van Fraassen, eds., Plenum, New York, pp. 375–380.

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, Reading, MA.

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

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

Reichenbach, H. (1998). *Philosophic Foundations of Quantum Mechanics*, Dover Publications, California.

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.