International Journal of Theoretical Physics, Vol. 39, No. 11, 2000

Automata Theory Based on Quantum Logic II

Mingsheng Ying[1]

Received May 1, 2000

We establish the pumping lemma in automata theory based on quantum logic under certain conditions on implication, and discuss the recognizability by the product and union of orthomodular lattice-valued (quantum) automata. In particular, we show that the equivalence between the recognizabilty by the product of automata and the conjunction of the recognizabilities by the factor automata is equivalent to the distributivity of meet over union in the truth-value set.

1. INTRODUCTION

The idea of establishing mathematics based on many-valued logics was first proposed by Rosser and Turquette [RT52] in 1952, but this idea has not attracted much attention in logical community. One reason for this may be that there is no suitable method to develop mathematics within the framework of many-valued logics. As is well known, classical logic is the underlying logic of classical mathematics and the former is used as the deduction tool in the latter. In other words, what is used in classical mathematics is the deduction (proof-theoretic) aspect of classical logic. However, the proof theory of many-valued logics is much more complicated than that of classical logic and it is not an easy task to conduct reasoning in the realm of the proof theory of many-valued logics, and, even worse, the axiomatizations of some many-valued logics are still to be found. Thus, our experience in studying classical mathematics may be not suited, or at least cannot directly apply, to develop mathematics based on many-valued logics. In the early 1990s, the author [Y91–93, Y93] established elementarily topology based on residuated lattice-valued logic by employing so-called a semantical analysis approach. Roughly speaking, the semantical analysis approach transforms our intended

[1] State Key Laboratory of Intelligent Technology and Systems, Department of Computer Science and Tsinghua University, Beijing 100084, China; e-mail: yingmsh@mail.tsinghua.edu.cn

conclusions in mathematics, which are usually expressed as implication formulas in our logical language, into inequalities in the truth-value set by truth valuation rules, and then we demonstrate these inequalities in an algebraic way and conclude that the conclusions are semantically valid.

Recently, the author used the semantical analysis approach to study automata theory based on quantum logic. Our purpose is to provide a new model of quantum computation [F82, F86, D85]. Computation based on quantum mechanics was first realized by Feynman [F82, F86], and then was elaborated and formalized by Deutsch [D85]. Since Deutsch's seminal work [D85], several theoretical models of quantum computation were introduced; for example, see [MC00]. These models may be roughly seen as the quantum probabilistic version of automata. The author [Y00], however, proposed to study quantum computation from a different direction, namely, to develop automata theory based on quantum logic. As early as 1936, Birkhoff and von Neumann [BvN36] realized that quantum mechanical systems are not governed by classical logical laws, and they introduced orthomodular lattice-valued logic as the logic of quantum mechanism. Quantum logic has received a great deal of interest in both the mathematical physics community and the logic community; see [DC86, RR91, RZ99] for examples. With this background in mind, one may naturally conceive that it is interesting to establish computing theory based on quantum logic. The author's paper [Y00] might be an initial step toward this direction. In that paper, we presented a basic framework of automata theory based on quantum logic. In particular, the orthomodular lattice-valued (quantum) predicate of recognizability was introduced. Then we clarified the relationship between classical recognizability and quantum recognizability; that is, we gave a simple conncetion between them, and showed that quantum recognizability does not in general degenerate into the classical one by an example. It is also shown that the recognizability of a quantum language is not less than the volume of its finite part. Furthermore, we introduced the inverse operation of a quantum automaton and showed that the language recognized by the inverse of a quantum automaton is the inverse of the language recognized by this quantum automaton, simply generalizing the corresponding result in classical automata theory,

This paper is a continuation of [Y00]. In this paper, we establish the pumping lemma in automata theory based on quantum logic under certain natural conditions on implication (see Proposition 1), introduce product and union of quantum automata, and discuss the relation between the language recognized by the product or union of quantum automata and the languages recognized by its components. The case of union is very simple, and we have a direct generalization of the result regarding the union of automata in the classical theory; that is, the language recognized by the union of quantum automata is the union of the languages recognized by these quantum automata (see Proposition 3). The situation is completely different for the product of quantum automata. In classical automata theory, the language recognized by the product of automata is the intersection of the languages recognized by the factors. However, this result holds if and only if the meet is distributive over the union in the truth-value set of the underlying logic (see Proposition 2). This is indeed a negative result in automata theory based on quantum logic. Because an orthomodular lattice possessing the distributivity is a Boolean algebra, the result concerning the product of automata in the classical theory is no longer true unless the underlying quantum logic degenerates to classical (Boolean) logic. This negative result may help us to clarify the boundary between classical computation and quantum computation.

2. PRELIMINARIES

For convenience, in this section we recall some notions and notations in quantum logic [DC86, RR91, RZ99, S98]. Also, we review several concepts from our previous work [Y00].

First, we consider truth-value sets of quantum logic. What we use as truth-value sets of quantum logic are complete orthomodular lattices. A complete orthomodular lattice is a 7-tuple $l = \langle L, \leq, \wedge, \vee, \perp, 1, 0\rangle$, where:

1. $\langle L, \leq, \wedge, \vee, 1, 0\rangle$ is a complete lattice; 1 and 0 are respectively the greatest element and the least element of L; $\leq$ is the partial ordering in L; and for any $M \subseteq L$, $\wedge M$ and $\vee M$ stand for the greatest lower bound and the least upper bound of M, respectively. Thus, $1 = \wedge\phi = \vee L$ and $0 = \vee\phi = \wedge L$. The binary $\wedge$ and $\vee$ are respectively called meet and union.

2. $\perp$ is a unary operation on L, called orthocomplement, and it satisfies the following conditions for all $a, b \in L$:
(a) $a \wedge a^{\perp} = 0$ and $a \vee a^{\perp} = 1$.
(b) $a^{\perp\perp} = a$.
(c) $a \leq b$ implies $b^{\perp} \leq a^{\perp}$.
(d) $a \wedge (a^{\perp} \vee (a \wedge b)) \leq b$.

Quantum logic is an orthomodular lattice-valued logic. Obviously, orthocomplement $\perp$, meet $\wedge$, and union $\vee$ in an orthomodular lattice may serve as the truth-value functions of connectives: negation, conjunction, and disjunction, respectively, in quantum logic; and 1 and 0 may act as the respective interpretations of truth and falsity. In addition, we are concerned with complete orthomodular lattices in this paper; so, arbitrary $\wedge$ and $\vee$ are well defined and they may be used to interpret the universal and existential quantifiers. Therefore, what is still missing in a complete orthomodular lattice employed as a truth-value set of quantum logic is a binary operation over it which is suited to be the truth-value function of implication. As is well known, all implication operators that one can reasonably envisage in an orthomodular lattice are more or less anomalous in the sense that they do not share this or that fundamental property of the implication in classical logic. Usually one chooses a fixed implication operator with a certain reason and then establishes various theorems in the quantum logic with this specific implication. For example, Román and Zuazua [RZ99] argued the reasonableness of the Sasaki arrow. In [Y00] we also adopted the Sasaki arrow as implication operator. For a detailed discussion on implications in quantum logic, see DC86, Section 2. In this paper, a different strategy is used: we suppose a truth-value set of quantum logic is a complete orthomodular lattice equipped with a binary operation $\rightarrow$. This operation $\rightarrow$ will be used as the truth-value function of implication in quantum logic, but we leave it completely unspecified and then observe what conditions it should satisfy to validate an intended property of quantum automata.

Second, we present the syntax of quantum logic. According to the strategy stated above, we now assume that $l = \langle L, \leq, \wedge, \vee, \perp, 1, 0\rangle$ is a complete orthomodular lattice and $\rightarrow$: $L \times L \rightarrow L$ is a binary operation on L. Then we work within the l-valued (quantum) logic. The language of l-valued logic has three primitive connectives: one unary connective $\neg$ (negation) and two binary connectives $\wedge$ (conjunction) and $\rightarrow$ (implication). Note that here $\rightarrow$ must be treated as a primitive connective in our logic because the implication operator in the truth-value set is presumed, but not derived from other operations. This is different from what we did in [Y00] where we adopted the Sasaki arrow as implication operator, and the Sasaki arrow may be defined in terms of orthocomplement and meet and union. The logical language also has a primitive quantifier $\forall$ (universal quantifier). In what follows we shall additionally employ the set-theoretic predicate symbol $\in$ (membership) as a primitive symbol. The syntax of l-valued logic is defined in a familiar way; we omit its details, but display several derived formulas in the l-valued logical language and set-theoretic language. These formulas will be needed in the sequel:

1. $\varphi \bigvee \psi \overset{\text{def}}{=} \neg(\neg\varphi \bigwedge \neg\psi)$
2. $\varphi \leftrightarrow \psi \overset{\text{def}}{=} (\varphi \rightarrow \psi) \bigwedge (\psi \rightarrow \varphi)$
3. $(\exists x)\varphi \overset{\text{def}}{=} \neg(\forall x) \neg\varphi$
4. $A \subseteq B \overset{\text{def}}{=} (\forall x)(x \in A \rightarrow x \in B)$
5. $A \equiv B \overset{\text{def}}{=} (A \subseteq B) \bigwedge (B \subseteq A)$

Third, we give the semantics of quantum logic. An l-valued interpretation is different from an interpretation in classical logic only in that each predicate symbol is associated with an l-valued relation, i.e., a mapping from the product of some copies of the discourse universe into L, where the number of copies is exactly the arity of the predicate symbol. As usual, an l-valued interpretation determines a truth valuation, i.e., a mapping from the set of (well-formed) formulas into the truth-value set. This truth valuation assigns an element of the truth-value set to each formula and this element is the truth value of the formula under the interpretation. For any formula φ, its truth value is in L and we write $\lceil\varphi\rceil$ for it. Then the truth valuation for logical and set-theoretic formulas is given by the following valuation rules:

1. $\lceil\neg\varphi\rceil = \lceil\varphi\rceil^{\perp}$.
2. $\lceil\varphi \wedge \psi\rceil = \lceil\varphi\rceil \wedge \lceil\psi\rceil$.
3. $\lceil\varphi \rightarrow \psi\rceil = \lceil\varphi\rceil \rightarrow \lceil\psi\rceil$.
4. If U is the discourse universe, then $\lceil(\forall x)\varphi\rceil = \bigwedge_{u\in U}\lceil\varphi\{u/x\}\rceil$.
5. $\lceil x \in A\rceil = A(x)$, where A on the left-hand side is a set constant (unary predicate symbol) and its interpretation is A on the right-hand side, which is an l-valued subset of the discourse universe, i.e., a mapping from the discourse universe into L.

It is worthy to note that in the second and third truth valuation rules $\wedge$ and $\rightarrow$ on the left-hand side are two connectives in quantum logic, whereas $\wedge$ and $\rightarrow$ on the right-hand side are two operations on the truth-value set L. Let Γ be a set of formulas and φ a formula. Then φ is called a semantic consequence of Γ in l-valued logic, written $\Gamma \overset{l}{\models} \varphi$, if we have $\wedge_{\psi\in\Gamma}\lceil\psi\rceil \leq \lceil\varphi\rceil$ for all l-valued interpretations.

We conclude this section by recalling the concept of quantum automaton. Let Σ be a finite alphabet. Then an l-valued (quantum) automaton over Σ is a quadruple $\Re = \langle Q, I, T, \delta\rangle$, where:

1. Q is a finite set of states.
2. $I \subseteq Q$ is the set of initial states.
3. $T \subseteq Q$ is the set of terminal states.
4. δ is an l-valued subset of $Q \times \Sigma \times Q$, i.e., a mapping from $Q \times \Sigma \times Q$ into L and it is called the l-valued (quantum) transition relation of $\Re$. Intuitively, for any $p, q \in Q$ and $\sigma \in \Sigma$, $\delta(p, \sigma, q)$ indicates the truth value of the proposition that input σ causes state p to become q.

An l-valued automaton over Σ determines an l-valued (unary) predicate $\mathrm{rec}_{\Re}$ on $\Sigma^* = \cup_{k=0}^{\infty}\Sigma^k$, and it is defined as follows: for all $k \geq 0$, $\sigma_1, \ldots, \sigma_k \in \Sigma$,

$$\mathrm{rec}_{\Re}(\sigma_1 \cdots \sigma_k)$$

$$\overset{\mathrm{def}}{=} (\exists q_0 \in I, q_1, \ldots, q_{k-1} \in Q, q_k \in T)\ \mathrm{path}_{\Re}(q_0\sigma_1 q_1 \cdots q_{k-1}\sigma_k q_k)$$

where

$$\mathrm{path}_{\mathfrak{R}}(q_0\sigma_1 q_1 \cdots q_{k-1}\sigma_k q_k) \stackrel{\mathrm{def}}{=} \bigwedge_{i=0}^{k-1}[(q_i, \sigma_{i+1}, q_{i+1}) \in \delta]$$

Intuitively, $\mathrm{rec}_{\mathfrak{R}}(\sigma_1 \cdots \sigma_k)$ stands for the proposition that the word $\sigma_1 \cdots \sigma_k$ is recognized by the quantum automaton $\mathfrak{R}$, and its truth value is

$$\lceil \mathrm{rec}_{\mathfrak{R}}(\sigma_1 \cdots \sigma_k) \rceil = \bigvee_{q_0 \in I, q_1, \ldots, q_{k-1} \in Q, q_k \in T} \bigwedge_{i=0}^{k-1} \delta(q_i, \sigma_{i+1}, q_{i+1})$$

We call an l-valued subset of Σ^* an l-valued (quantum) language over Σ. Thus, $\mathrm{rec}_{\mathfrak{R}}$ may be interpreted as an l-valued language over Σ, i.e., $\mathrm{rec}_{\mathfrak{R}}$ may be seen as a mapping from Σ^* into L and for each $s \in \Sigma^*$ it assigns $\lceil \mathrm{rec}_{\mathfrak{R}}(s) \rceil$ to s. We may further define the l-valued (unary) predicate Rec_{Σ} on l-valued languages over Σ: for any $A \in L^{\Sigma^*}$,

$$\mathrm{Rec}_{\Sigma}(A) \stackrel{\mathrm{def}}{=} (\exists \mathfrak{R} \in \mathbf{A}(\Sigma, l))(A \equiv \mathrm{rec}_{\mathfrak{R}})$$

where $\mathbf{A}(\Sigma, l)$ is the class of l-valued automata over Σ. Intuitively, $\mathrm{Rec}_{\Sigma}(A)$ is the proposition that the l-valued language A is recognizable. It is easy to see that its truth value is

$$\lceil \mathrm{Rec}_{\Sigma}(A) \rceil = \bigvee_{\mathfrak{R} \in \mathbf{A}(\Sigma, l)} \bigwedge_{s \in \Sigma^*} (A(s) \leftrightarrow \lceil \mathrm{rec}_{\mathfrak{R}}(s) \rceil)$$

where $a \leftrightarrow b \stackrel{\mathrm{def}}{=} (a \to b) \wedge (b \to a)$ for any $a, b \in L$.

3. THE PUMPING LEMMA

In classical automata theory, the pumping lemma is a powerful tool to show that certain languages are not regular. The purpose of this section is to establish a pumping lemma in automata theory based on quantum logic. As we shall see shortly, this pumping lemma depends on certain properties of implication in the underlying logic.

Proposition 1 (Pumping Lemma). Let the implication operator $\to$ satisfy the following conditions for all $a, b, c \in L$ and for any $\{a_i: i \in I\}, \{b_i: i \in I\} \subseteq L$:

1. $a \leq b$ implies $a \to b = 1$
2. $b \leq c$ implies $a \to b \leq a \to c$
3. $(a \to b) \wedge (b \to c) \leq a \to c$
4. $\wedge_{i \in I} (a_i \to b_i) \leq \wedge_{i \in I} a_i \to \wedge_{i \in I} b_i$
5. $\wedge_{i \in I} (a_i \to b_i) \leq \vee_{i \in I} a_i \to \vee_{i \in I} b_i$

Then for any l-valued language $A \in L^{\Sigma^*}$ over Σ,

$$\overset{l}{\models} \mathrm{Rec}_\Sigma(A) \to (\exists n \geq 0)\,(\forall s \in \Sigma^*)\,[s \in A \wedge |s| \geq n \to$$
$$(\exists u, v, w, \in \Sigma^*)\,(s = uvw \wedge |uv| \leq n \wedge |v| \geq 1$$
$$\wedge\,(\forall i \geq 0)\,(uv^iw \in A))]$$

where for any word $t = \sigma_1 \cdots \sigma_k \in \Sigma^*$, $|t|$ stands for the length n of t.

Proof. With the condition 1, it suffices to show that

$$\lceil \mathrm{Rec}_\Sigma(A) \rceil \leq \lceil (\exists n \geq 0)\,(\forall s \in \Sigma^*)\,[s \in A \wedge |s| \geq n \to (\exists u, v, w \in \Sigma^*)$$
$$(s = uvw \wedge |uv| \leq n \wedge |v| \geq 1 \wedge (\forall i \geq 0)\,(uv^iw \in A))] \rceil$$
$$= \bigvee_{n\geq 0} \bigwedge_{s\in\Sigma^*,|s|\geq n} (A(s) \to \bigvee_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq n,|v|\geq 1} \bigwedge_{i\geq 0} A(uv^iw))$$

Noting that

$$\lceil \mathrm{Rec}_\Sigma(A) \rceil = \bigvee_{\mathfrak{R}\in\mathbf{A}(\Sigma,l)} \lceil A \equiv \mathrm{rec}_{\mathfrak{R}} \rceil$$

we only need to prove that for any $\mathfrak{R} \in \mathbf{A}(\Sigma, l)$,

$$\lceil A \equiv \mathrm{rec}_{\mathfrak{R}} \rceil \leq \bigvee_{n\geq 0} \bigwedge_{s\in\Sigma^*,|s|\geq n} (A(s)$$
$$\to \bigvee_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq n,|v|\geq 1} \bigwedge_{i\geq 0} A(uv^iw))$$

Now, we demonstrate the above inequality. Let $\mathfrak{R} \in \mathbf{A}(\Sigma, l)$ and let Q be the set of states of $\mathfrak{R}$. First, it holds that for any $s \in \Sigma^*$ with $|s| \geq |Q|$,

$$\mathrm{rec}_{\mathfrak{R}}(s) \leq \bigvee_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq|Q|,|v|\geq 1} \bigwedge_{i\geq 0} \mathrm{rec}_{\mathfrak{R}}(uv^iw) \qquad (1)$$

In fact, suppose that $s = \sigma_1 \cdots \sigma_k$. Then

$$\mathrm{rec}_{\mathfrak{R}}(s) = \bigvee_{q_0\in I,q_1,\ldots,q_{k-1}\in Q,q_k\in T} \bigwedge_{i=0}^{k-1} \delta(q_i, \sigma_{i+1}, q_{i+1}) \qquad (2)$$

Therefore, it amounts to showing that for any $q_0 \in I, q_1, \ldots, q_{k-1} \in Q$ and $q_k \in T$,

$$\bigwedge_{i=0}^{k-1} \delta(q_i, \sigma_{i+1}, q_{i+1}) \leq \bigvee_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq|Q|,|v|\geq 1} \bigwedge_{i\geq 0} \mathrm{rec}_{\mathfrak{R}}(uv^iw) \quad (3)$$

Since $k = |s| \geq |Q|$, there are two identical states among $q_0, q_1, \ldots, q_{|Q|}$; in other words, there are $m \geq 0$ and $n > 0$ such that $m + n \leq |Q|$ and $q_m = q_{m+n}$. We set $u_0 = \sigma_1 \cdots \sigma_m$, $v_0 = \sigma_{m+1} \cdots \sigma_{m+n}$, and $w_0 = \sigma_{m+n+1} \cdots \sigma_k$. Then $s = u_0v_0w_0$, $|u_0v_0| = m + n \leq |Q|$, $|v| = n \geq 1$, and

$$\bigvee_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq|Q|,|v|\geq 1} \bigwedge_{i\geq 0} \mathrm{rec}_{\mathfrak{R}}(uv^iw) \geq \bigwedge_{i\geq 0} \mathrm{rec}_{\mathfrak{R}}(u_0v_0^iw_0) \qquad (4)$$

From the definition of $\mathrm{rec}_{\mathfrak{R}}$, it is easy to see that for all $i \geq 0$,

$$\mathrm{rec}_{\mathfrak{R}}(u_0v_0^iw_0)$$
$$\geq \lceil \mathrm{path}_{\mathfrak{R}}(q_0\sigma_1q_1 \cdots \sigma_mq_m(\sigma_{m+1}q_{m+1} \cdots \sigma_{m+n}q_{m+n})^i$$

$$\sigma_{m+n+1} q_{m+n+1} \cdots \sigma_k q_k)\rceil$$
$$= \bigwedge_{i=0}^{m+n-1} \delta(q_j, \sigma_{j+1}, q_{j+1}) \wedge \bigwedge_{l=1}^{i-1} [\delta(q_{m+n}, \sigma_{m+1}, q_{m+1})$$
$$\wedge \bigwedge_{j=m+1}^{m+n-1} \delta(q_j, \sigma_{j+1}, q_{j+1})] \wedge \bigwedge_{j=m+n}^{k-1} \delta(q_j, \sigma_{j+1}, q_{j+1})$$
$$= \bigwedge_{j=0}^{k-1} \delta(q_j, \sigma_{j+1}, q_{j+1}) \tag{5}$$

because $q_{m+n} = q_m$ and $\delta(q_{m+n}, \sigma_{m+1}, q_{m+1}) = \delta(q_m, \sigma_{m+1}, q_{m+1})$. Thus, by combining (4) and (5), we obtain (3) which, together with (2), yields (1).

Second, with the conditions 4 and 5 we have

$$\bigvee_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq|Q|,|v|\geq 1} \bigwedge_{i\geq 0} \mathrm{rec}_{\Re}(uv^i w)$$
$$\to \bigvee_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq|Q|,|v|\geq 1} \bigwedge_{i\geq 0} A(uv^i w)$$
$$\geq \bigwedge_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq|Q|,|v|\geq 1} (\bigwedge_{i\geq 0} \mathrm{rec}_{\Re}(uv^i w) \to \bigwedge_{i\geq 0} A(uv^i w))$$
$$\geq \bigwedge_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq|Q|,|v|\geq 1} \bigwedge_{i\geq 0} (\mathrm{rec}_{\Re}(uv^i w) \to A(uv^i w))$$
$$\geq \bigwedge_{t\in\Sigma^*} (\mathrm{rec}_{\Re}(t) \to A(t)) = \lceil \mathrm{rec}_{\Re} \subseteq A\rceil \tag{6}$$

Finally, using (1) and (6) and Conditions 2 and 3, we obtain

$$\lceil A \equiv \mathrm{rec}_{\Re}\rceil = \lceil A \subseteq \mathrm{rec}_{\Re}\rceil \wedge \lceil \mathrm{rec}_{\Re} \subseteq A\rceil$$
$$= \bigwedge_{s\in\Sigma^*}(A(s) \to \mathrm{rec}_{\Re}(s)) \wedge \lceil \mathrm{rec}_{\Re} \subseteq A\rceil$$
$$= \bigwedge_{s\in\Sigma^*}((A(s) \to \mathrm{rec}_{\Re}(s)) \wedge \lceil \mathrm{rec}_{\Re} \subseteq A\rceil)$$
$$\leq \bigwedge_{s\in\Sigma^*,|s|\geq|Q|} ((A(s) \to \mathrm{rec}_{\Re}(s)) \wedge \lceil \mathrm{rec}_{\Re} \subseteq A\rceil)$$
$$\leq \bigwedge_{s\in\Sigma^*,|s|\geq|Q|} (\{A(s) \to \bigvee_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq|Q|,|v|\geq 1}$$
$$\bigwedge_{i\geq 0} \mathrm{rec}_{\Re}(uv^i w)\} \wedge \lceil \mathrm{rec}_{\Re} \subseteq A\rceil)$$
$$\leq \bigwedge_{s\in\Sigma^*,|s|\geq|Q|} (\{A(s) \to \bigvee_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq|Q|,|v|\geq 1}$$
$$\bigwedge_{i\geq 0} \mathrm{rec}_{\Re}(uv^i w)\}$$
$$\wedge \{\bigvee_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq|Q|,|v|\geq 1} \bigwedge_{i\geq 0} \mathrm{rec}_{\Re}(uv^i w) \to$$
$$\bigvee_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq|Q|,|v|\geq 1} \bigwedge_{i\geq 0} A(uv^i w)\})$$
$$\leq \bigwedge_{s\in\Sigma^*,|s|\geq|Q|} (A(s) \to \bigvee_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq|Q|,|v|\geq 1}$$
$$\bigwedge_{i\geq 0} A(uv^i w))$$
$$\leq \bigvee_{n\geq 0} \bigwedge_{s\in\Sigma^*,|s|\geq n} (A(s) \to \bigvee_{u,v,w\in\Sigma^*,s=uvw,|uv|\leq n,|,v|\geq 1}$$
$$\bigwedge_{i\geq 0} A(uv^i w))$$

This completes the proof. ■

4. OPERATIONS ON QUANTUM AUTOMATA

In automata theory, product, union, and inverse of automata are three basic operations. The inverse of the automaton was discussed in [Y00]. This section considers the relationship between the language recognized by the product or union of automata and the languages recognized by its components in quantum logic. First, we consider the product of quantum automata. Let $\Re = (Q_A, I_A, T_A, \delta_A)$, $\wp = (Q_B, I_B, T_B, \delta_B) \in \mathbf{A}(\Sigma, l)$ be two l-valued automata over Σ. Then their product $\Im = \Re \times \wp$ is $(Q_C, I_C, T_C, \delta_C)$ and it is defined as follows:

1. $Q_C = Q_A \times Q_B$.
2. $I_C = I_A \times I_B$.
3. $T_C = T_A \times I_B$.
4. δ_C: $Q_C \times \Sigma \times Q_C \to L$ and for any $\sigma \in \Sigma$, p_a, $q_a \in Q_A$, and p_b, $q_b \in Q_B$,

$$\delta_C((p_a, p_b), \sigma, (q_a, q_b)) = \delta_A(p_a, \sigma, q_a) \bigwedge \delta_B(p_b, \sigma, q_b)$$

Proposition 2. Suppose that the implication operator $\to$ on L satisfies that $a \leftrightarrow b \stackrel{\text{def}}{=} (a \to b) \bigwedge (b \to a) = 1$ if and only if $a = b$ for any $a, b \in L$. Then the following two statements are equivalent:

(i) The meet $\wedge$ is (finitely) distributive over the union $\vee$ in l, i.e., for all $a, b, c \in L$, $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$.

(ii) For any $\Re$, $\wp \in \mathbf{A}(\Sigma, l)$ and for any $s \in \Sigma^*$

$$\stackrel{l}{\models} \mathrm{rec}_{\Re\times\wp}(s) \leftrightarrow \mathrm{rec}_{\Re}(s) \bigwedge \mathrm{rec}_{\wp}(s)$$

Proof. We first prove (i) implies (ii). Let $s = \sigma_1 \cdots \sigma_k$. Then

$$\begin{aligned}\lceil \mathrm{rec}_{\Re}(s) \bigwedge \mathrm{rec}_{\wp}(s) \rceil &= \lceil \mathrm{rec}_{\Re}(s) \rceil \bigwedge \lceil \mathrm{rec}_{\wp}(s) \rceil \\ &= \{\bigvee_{q_{a0}\in I_A, q_{a1},\ldots,q_{a(k-1)}\in Q_A, q_{ak}\in T_A} \bigwedge_{i=0}^{k-1} \delta_A(q_{ai}, \sigma_{i+1}, q_{a(i+1)})\} \\ &\quad \bigwedge \{\bigvee_{q_{b0}\in I_B, q_{b1},\ldots,q_{b(k-1)}\in Q_B, q_{bk}\in T_B} \bigwedge_{i=0}^{k-1} \delta_B(q_{bi}, \sigma_{i+1}, q_{b(i+1)})\}\end{aligned}$$

Since Q_A and Q_B are finite, I_A, $T_A \subseteq Q_A$ and I_B, $T_B \subseteq Q_B$, we can use (i) and obtain:

$$\begin{aligned}&\lceil \mathrm{rec}_{\Re}(s) \bigwedge \mathrm{rec}_{\wp}(s) \rceil \\ &\quad = \bigvee_{q_{a0}\in I_A, q_{a1},\ldots,q_{a(k-1)}\in Q_A, q_{ak}\in T_A} \\ &\quad\quad \bigvee_{q_{b0}\in I_B, q_{b1},\ldots,q_{b(k-1)}\in Q_B, q_{bk}\in T_B} [\bigwedge_{i=0}^{k-1} \delta_A(q_{ai}, \sigma_{i+1}, q_{a(i+1)}) \\ &\quad\quad \bigwedge\bigwedge_{i=0}^{k-1} \delta_B(q_{bi}, \sigma_{i+1}, q_{b(i+1)})]\end{aligned}$$

$$= \bigvee_{(q_{a0},q_{b0})\in I_A\times I_B,\ (q_{a1},q_{b1}),\dots,(q_{a(k-1)},q_{b(k-1)})\in Q_A\times Q_B,(q_{ak},q_{bk})\in T_A\times T_B}$$

$$[\bigwedge_{i=0}^{k-1} (\delta_A(q_{ai}, \sigma_{i+1}, q_{a(i+1)}) \wedge \delta_B(q_{bi}, \sigma_{i+1}, q_{b(i+1)}))]$$

$$\bigvee_{(q_{a0},q_{b0})\in I_A\times I_B,(q_{a1},q_{b1}),\dots,(q_{a(k-1)},q_{b(k-1)})\in Q_A\times Q_B,(q_{ak},q_{bk})\in T_A\times T_B}$$

$$\bigwedge_{i=0}^{k-1} \delta_{A\times B}((q_{ai}, q_{bi}), \sigma_{i+1}, (q_{a(i+1)}, q_{b(i+1)}))$$

$$= \lceil \mathrm{rec}_{\Re\times\wp}(s)\rceil$$

Second, we prove (ii) implies (i). For any $a, b, c \in L$, we choose some $\sigma_0 \in \Sigma$ and set $\Re = (\{p\}, \{p\}, \{p\}, \delta_A)$, where $\delta_A(p, \sigma, p) = a$ if $\sigma = \sigma_0$ and 0 otherwise, and $\wp = (\{q, r, s\}, \{q\}, \{r, s\}, \delta_B)$, where $\delta_B(x, \sigma, y) = b$ if $x = q, y = r$, and $\sigma = \sigma_0$; c if $x = q, y = s$, and $\sigma = \sigma_0$; and 0 otherwise. Then $\Re, \wp \in \mathbf{A}(\Sigma, l)$, and it is easy to show that $\Re \times \wp = (\{(p, q), (p, r), (p, s)\}, \{(p, q)\}, \{(p, r), (p, s)\}, \delta_{A\times B})$, where $\delta_{A\times B}((p, x), \sigma, (p, y)) = a \wedge b$ if $x = q, y = r$, and $\sigma = \sigma_0$; $a \wedge c$ if $x = q, y = s$ and $\sigma = \sigma_0$; and 0 otherwise. Furthermore, by a routine calculation we have

$$\lceil \mathrm{rec}_{\Re}(\sigma_0)\rceil = a$$
$$\lceil \mathrm{rec}_{\wp}(\sigma_0)\rceil = b \vee c$$
$$\lceil \mathrm{rec}_{\Re\times\wp}(\sigma_0)\rceil = (a \wedge b) \vee (a \wedge c)$$

Therefore, with (ii) we finally obtain

$$a \wedge (b \vee c) = \lceil \mathrm{rec}_{\Re}(\sigma_0)\rceil \wedge \lceil \mathrm{rec}_{\wp}(\sigma_0)\rceil$$
$$= \lceil \mathrm{rec}_{\Re\times\wp}(\sigma_0)\rceil = (a \wedge b) \vee (a \wedge c) \quad \blacksquare$$

We now consider the union of quantum automata. Let $\Re = (Q_A, I_A, T_A, \delta_A)$ and $\wp = (Q_B, I_B, T_B, \delta_B) \in \mathbf{A}(\Sigma, l)$ be two l-valued automata over Σ. We assume that $Q_A \cap Q_B = \phi$. Then the (disjoint) union of $\Re$ and $\wp$ is $\Im = \Re \cup \wp = (Q_C, I_C, T_C, \delta_C)$, where:

1. $Q_C = Q_A \cup Q_B$.
2. $I_C = I_A \cup I_B$.
3. $T_C = T_A \cup I_B$.
4. δ_C: $Q_C \times \Sigma \times Q_C \to L$ is defined as follows: for any $\sigma \in \Sigma$ and for any $p, q \in Q_C = Q_A \cup Q_B$, $\delta_C(p, \sigma, q) = \delta_A(p, \sigma, q)$ if $p, q \in Q_A$, $\delta_B(p, \sigma, q)$ if $p, q \in Q_B$: and 0 otherwise.

Proposition 3. If the implication operator $\to$ satisfies that $a \leftrightarrow a = 1$ for any $a \in L$, then for any $\Re, \wp \in \mathbf{A}(\Sigma, l)$ and for any $s \in \Sigma^*$

$$\overset{l}{\models} \mathrm{rec}_{\Re \cup \wp}(s) \leftrightarrow \mathrm{rec}_{\Re}(s) \bigvee \mathrm{rec}_{\wp}(s)$$

Proof. Let $s = \sigma_1 \ldots \sigma_k$. Then

$$\begin{aligned}\lceil \mathrm{rec}_{\Re\cup\wp}(s)\rceil &= \bigvee_{q_0 \in I_A \cup I_B, q_1,\ldots,q_{k-1} \in Q_A \cup Q_B, q_k \in T_A \cup T_B} \bigwedge_{i=0}^{k-1} \delta_{A\cup B}(q_i, \sigma_{i+1}, q_{i+1}) \\ &= [\bigvee[_{q_0 \in I_A, q_1,\ldots,q_{k-1} \in Q_A, q_k \in T_A} \bigwedge_{i=0}^{k-1} \delta_{A\cup B}(q_i, \sigma_{i+1}, q_{i+1})] \\ &\quad \bigvee [_{q_0 \in I_B, q_1,\ldots,q_{k-1} \in Q_B, q_k \in T_B} \bigwedge_{i=0}^{k-1} \delta_{A\cup B}(q_i, \sigma_{i+1}, q_{i+1})] \\ &\quad \bigvee\{\bigwedge_{i=0}^{k-1} \delta_{A\cup B}(q_i, \sigma_{i+1}, q_{i+1}) : q_0 \in I_A \cup I_B, q_1, \ldots, q_{k-1} \\ &\quad \in Q_A \cup Q_B, q_k \in T_A \cup T_B, \text{ and there are } i, j \text{ such that} \\ &\quad 0 \leq i, j \leq k \text{ and } q_i \in Q_A \text{ and } q_j \in Q_B\}\end{aligned}$$

From the definition of $\Re\cup\mathfrak{p}$, we know that for any $q_0 \in I_A, q_1, \ldots, q_{k-1} \in Q_A$, $q_k \in T_A$, we have

$$\bigwedge_{i=0}^{k-1} \delta_{A\cup B}(q_i, \sigma_{i+1}, q_{i+1}) = \bigwedge_{i=0}^{k-1} \delta_A(q_i, \sigma_{i+1}, q_{i+1})$$

and for any $q_0 \in I_B, q_1, \ldots, q_{k-1} \in Q_B, q_k \in T_B$, we have

$$\bigwedge_{i=0}^{k-1} \delta_{A\cup B}(q_i, \sigma_{i+1}, q_{i+1}) = \bigwedge_{i=0}^{k-1} \delta_B(q_i, \sigma_{i+1}, q_{i+1})$$

If $q_0 \in I_A \cup I_{\mathrm{B}}, q_1, \ldots, q_{k-1} \in Q_A \cup Q_B, q_k \in T_A \cup T_B$, and there are i, j such that $0 \leq i, j \leq k$ and $q_i \in Q_A$ and $q_j \in Q_B$, then we can find some $m \in \{0, 1, \ldots, k-1\}$ such that $q_m \in Q_A$ and $q_{m+1} \in Q_B$, or $q_m \in Q_B$ and $q_{m+1} \in Q_A$. Then $\delta_{A\cup B}(q_m, \delta_{m+1}, q_{m+1}) = 0$, and

$$\bigwedge_{i=0}^{k-1} \delta_{A\cup B}(q_i, \sigma_{i+1}, q_{i+1}) = 0$$

Therefore, it follows that

$$\begin{aligned}\lceil \mathrm{rec}_{\Re\cup\mathfrak{p}}(s)\rceil &= [\bigvee_{q_0 \in I_A, q_1,\ldots,q_{k-1} \in Q_A, q_k \in T_A} \bigwedge_{i=0}^{k-1} \delta_A(q_i, \sigma_{i+1}, q_{i+1})] \\ &\quad \bigvee [\bigvee_{q_0 \in I_B, q_1,\ldots,q_{k-1} \in Q_B, q_k \in T_B} \bigwedge_{i=0}^{k-1} \delta_B(q_i, \sigma_{i+1}, q_{i+1})] \\ &= \lceil \mathrm{rec}_{\Re}(s)\rceil \bigvee \lceil \mathrm{rec}_{\mathfrak{p}}(s)\rceil \\ &= \lceil \mathrm{rec}_{\Re}(s) \bigvee \mathrm{rec}_{\mathfrak{p}}(s)\rceil \quad \blacksquare\end{aligned}$$

5. CONCLUSION

In this paper, we continue our study of automata theory based on quantum logic initiated in [Y00]. The pumping lemma is generalized into the setting of orthomodular lattice-valued (quantum) automata, the operations of product and union of quantum automata are introduced, and the relation between the language recognized by these operations of quantum automata and the languages recognized by the operands is discussed. The most interesting result obtained in this paper is that the language recognized by the product of automata is the intersection of the languages recognized by the factors if and only if the truth-value lattice of the underlying logic is distribtutive. This is a negative conclusion in automata theory based on quantum logic, and it means that the classical result concerning the product of automata does not extend to quantum automata. This negative result stimulates us to consider the problem of a logical revisit to mathematics. Various classical mathematical results have been established based upon classical logic. Mathematicians usually use logic implicitly in their reasoning, and they do not seriously care which logical laws they have employed. But from a logician's point of view, it is very interesting to determine how strong a logic we need to validate a given mathematical theorem, and which logic guarantees this theorem and which does not among the large population of nonclassical logics. This suggests new research topics. Proposition 2 is a simple example of this kind of research.

ACKNOWLEDGMENTS

This work was supported by the National Foundation for Distinguished Young Scholars (Grant No. 69725004), the National Key Project for Basic Research (Grant No. 1998030509), the National Foundation of Natural Sciences (Grant No. 69823001), and the National Research and Development Project of High Technology (Grant No. 863-306-05-05-3B) of China, and it was carried out when the author was at Institut für Informatik, Ludwig-Maximilians-Universität München, as an Alexander von Humboldt Research Fellow.

REFERENCES

G. Birkhoff and J. von Neumann (1936). *Ann. Math.* **37** 823–843.

M. L. Dalla Chiara, In D. Gabbay and E. Guenthner (eds.), *Handbook of Philosophical Logic, Vol. III: Alternatives to Classical Logic*, Reidel, Dordrecht (1986) pp. 427–469.

D. Deutsch (1985). *Proc. R. Soc. Lond. A* **400** 97–11.

S. Eilenberg, *Automata, Languages, and Machines, Vol. A*, Academic Press, New York (1974).

R. P. Feynman (1982). *Int. J. Theor. Phys.* **21** 467–488.

R. P. Feynman (1986). *Found. Phys.* **16** 507–531.

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

C. Moore and J. P. Crutchfield, Quantum automata and quantum grammars, *Theor. Computer Sci.*, to appear.

L. Román and B. Rumbos (1991). *Found. Phys.* **21** 727–734.

L. Román and R. E. Zuazua (1999). *Int. J. Theor. Phys.* **38** 793–797.
J. B. Rosser and A. R. Turquette, *Many-Valued Logics*, North-Holland, Amsterdam (1952).
K. Svozil, *Quantum Logic*, Springer-Verlag, Berlin (1998).
M. S. Ying (1991). *Fuzzy Sets Syst.* **39** 303–321; (1992). **47** 221–232; (1993). **55** 193–207.
M. S. Ying (1993). *Fuzzy Sets Syst.* **56** 337–373.
M. S. Ying (2000). *Int. J. Theor. Phys.* **39** 981–991.