International Journal of Theoretical Physics, Vol. 43, No. 5, May 2004 (© 2004)

Pumping Lemma for Quantum Automata

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

Received

Not all lattice-valued quantum automata possess the pumping property in its strict form. However the pumping lemma can be generalized, and all lattice-valued quantum automata possess the generalized pumping property.

KEY WORDS: pumping lemma; generalized pumping lemma; lattice-valued quantum automata.

1. INTRODUCTION

The pumping lemma is a classical and important concept in formal language theories. It plays a crucial role in recognizing the family of regular languages and also of context free languages. When generalizing traditional finite state automata to quantum automata, it is natural to ask whether something like the pumping lemma still exists in the new theory and what kind of role it plays. Many authors have discussed this problem. With respect to finite state quantum automata on Hilbert Spaces, Moore and Crutchfield (2000) have proved a weak form of pumping lemma, which states that, in some sense, the acceptance degree of the pumped input string uv^iw can approximate that of the original string uvw to any degree of exactness. On the other hand, Ying has proved a pumping lemma of lattice-valued finite state quantum automata (lfqa for short) under the assumption of some quantum logic rules (Ying, 2000a,b). Qiu also published a similar lemma

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

while generalizing the lfqa concept (Qiu, 2003). All these works are based on a formulation of the pumping lemma that is quite similar to that in traditional finite state automata theory (Hopcroft and Ullman, 1979). Since the acceptance of an input string by a lfqa has to get rid from the two-valued Boolean framework (a dichotomy of accepted or rejected) and based on lattice values (using lattice value to represent the degree of acceptance), the pumping lemma in quantum case should also stay away from the (0, 1) switch and follow the lattice-valued approach.

In this paper, we will introduce a new concept of pumping property for lfqa and prove a series of results with respect to this concept. The main result of this paper is the conclusion that the traditional pumping lemma can be generalized such that the generalized pumping lemma holds for lattice-valued finite state quantum automata.

2. BASIC DEFINITIONS OF LATTICE-VALUED FINITE STATE QUANTUM AUTOMATA

First we repeat the definition of a lfqa defined by Ying (Ying, 2000a) in a slightly different notation. At the same time we introduce a new version of lfqa and compare their acceptance characteristics.

Definition 2.1. (Ying, 2000a). Let $l = (L, \leq, 0, 1)$ be a lattice, Σ be a finite input alphabet. A lfqa R defined on (l, Σ) is a quadruple $R = (Q, I, T, \Delta)$, where $I \subseteq Q$ is a set of initial states, $T \subseteq Q$ is a set of terminating states, Δ is a set of l valued functions defined on $Q \times \Sigma \times Q$: for each $q_1, q_2 \in Q$ and $x \in \Sigma$, $\delta(q_1, x, q_2) \in \Delta$ is an element of l. Note that only those $\delta(q_1, x, q_2)$, which are not equal to 0 (least element of l) are listed in Δ. $\delta(q_1, x, q_2)$ is called the acceptance degree of x that the state q_1 is transformed to q_2 when the symbol x is inputted. Intuitionally, corresponding to each $\delta(q_1, x, q_2)$, a pair $(x, \delta(q_1, x, q_2))$ is attached to the arc from q_1 to q_2.

Definition 2.2. (Lu and Zheng, 2003). The lfqa are classified in type A lfqa and type B lfqa according to the way the acceptance degree of a whole input string is calculated.

Let $R = (Q, I, T, \Delta)$ be a lfqa defined on (l, Σ). For each i, j, where $\delta(q_i, x, q_j) \neq 0$, the pair $(x, \delta(q_i, x, q_j))$ is called a transition for the segment $q_i(x, \delta(q_i, x, q_j))q_j$ of R. A finite connection of segments $q_0(x_1, \delta(q_0, x_1, q_1))q_1$ $(x_2, \delta(q_1, x_2, q_2)) \cdots q_{n-1}(x_n, \delta(q_{n-1}, x_n, q_n))q_n$ is called a segment sequence of R, where all q_i belong to Q and all x_i belong to Σ. A simplified segment sequence is a segment sequence where the acceptance degree $\delta(q_i, x, q_j)$ of each transition does not appear explicitly. We get the label of a simplified segment sequence if all its states are dropped. In the above example, $w = q_0x_1q_1x_2q_2 \cdots q_{n-1}x_nq_n$ is a simplified segment sequence, $x_1x_2 \cdots x_n$ is its label. If q_0 belongs to I and q_n

belongs to T, then we say the (simplified) segment sequence is a (simplified) path of $\mathcal{R}$. Correspondingly, we call the symbol sequence $s = x_1x_2 \cdots x_n$ an accepted string of the automaton $\mathcal{R}$. Sometimes we will drop the word "simplified" in the remaining part of this paper, if no ambiguity will be raised.

The acceptance degree of s by this single path is defined as: $\text{Accept}_w(\mathcal{R}, s) = \cap_{i=0}^{n-1} \delta(q_i, x_{i+1}, q_{i+1})$, where $\cap$ is the meet operation of the lattice l.

Since the automaton $\mathcal{R}$ is in general nondeterministic, we have to calculate the acceptance degree of an input string s by integrating its acceptance degrees along all paths of $\mathcal{R}$. Let $T(\mathcal{R}, s) = \{w | w$ is a path of $\mathcal{R}, s$ is the accepted input string along this path $w\}$. It is easy to prove that $|T(\mathcal{R}, s)|$ is finite for every s. Since now the states q depend on the paths w, we write $q_{w,i}$ instead of q_i.

For type A lfqa, the acceptance degree of s by $\mathcal{R}$ is defined as

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

For type B lfqa, the acceptance degree of s by $\mathcal{R}$ is defined as

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

where $q_{w,i}, i = 0, 1, 2, \ldots, n$, are the states traversed by the path w. $\cup$ and $\cap$ are the two lattice operations join and meet. In a word, in the case of type A, we first calculate the acceptance degree of s for each single path, and then unite them together. In the case of type B, we take the first segment of all paths in $T(\mathcal{R}, s)$, unite their acceptance degrees $\delta(q_{w,0}, x_1, q_{w,1})$ together by the operation $\cup$. And then we take the second, third, ... transition of all paths and unite their acceptance degrees together separately by the same operation. At last, we perform the operation $\cap$ on all these united values and get the wanted general acceptance degree of s by $\mathcal{R}$. In any case, the language accepted by $\mathcal{R}$ is $\{(s, \text{Accept}(\mathcal{R}, s)) | s \in \Sigma^*\}$.

Example 2.1. Let $\mathcal{R} = (\{q_0, q_1, q_2, q_3\}, \{q_0\}, \{q_2, q_3\}, \{\delta(q_0, x, q_1) = a, \delta(q_1, y, q_2) = b, \delta(q_1, y, q_3) = c\})$. There are two paths for accepting the string xy in this automaton: $w_1 = q_0xq_1yq_2$ and $w_2 = q_0xq_1yq_3$. The single path acceptance degrees are $\text{Accept}_{w_1}(\mathcal{R}, xy) = a \cap b$ and $\text{Accept}_{w_2}(\mathcal{R}, xy) = a \cap c$ respectively. The general acceptance degree for type A is $\text{Accept}_\text{A}(\mathcal{R}, xy) = (a \cap b) \cup (a \cap c)$. On the other hand, The general acceptance degree for type B is $\text{Accept}_\text{B}(\mathcal{R}, xy) = a \cap (b \cup c)$. See Fig. 1 for an illustration.

We all know that for any elements a, b, and c of a lattice l, the following inclusion rule holds: $(a \cap b) \cup (a \cap c) \leq a \cap (b \cup c)$. Thus we have Accept_A

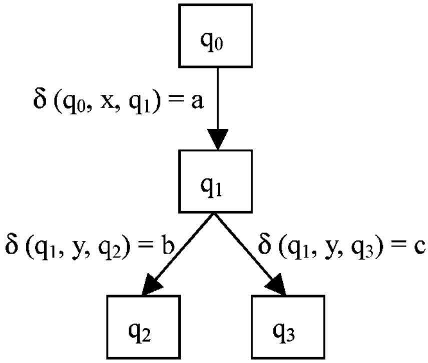

Fig. 1. Type A and type B lqfa.

$(\mathcal{R}, xy) = (a \cap b) \cup (a \cap c) \leq a \cap (b \cup c) = \text{Accept}_{\text{B}}(\mathcal{R}, xy)$. In fact, we have the more general.

Proposition 2.1. *For any lfqa $\mathcal{R}$, the general acceptance degrees, called recognizability in Ying (2000a), calculated according to rules of type A and type B have the relationship:*

1. $\text{Accept}_{\text{A}}(\mathcal{R}, s) \leq \text{Accept}_{\text{B}}(\mathcal{R}, s)$
2. *The inclusion symbol $\leq$ can be replaced by the equation symbol $=$ if $\mathcal{R}$ is a deterministic lfqa.*
3. *Let P be a set of paths accepting the string s in the lfqa $\mathcal{R}$, p be any path accepting the same string s. Then it is always*

$$\text{Accept}_{\text{A},P}(\mathcal{R}, s) \leq \text{Accept}_{\text{A},P\cup\{p\}}(\mathcal{R}, s) \tag{3}$$

$$\text{Accept}_{\text{B},P}(\mathcal{R}, s) \leq \text{Accept}_{\text{B},P\cup\{p\}}(\mathcal{R}, s) \tag{4}$$

where Accept$_{\text{A},P}(\mathcal{R}, s)$ means the acceptance degree of s along all paths belonging to P in case of type A. This explains also other notations in these expressions.

Proof: We prove only 3. Parts 1 and 2 are proved in Lu and Zheng (2003). Let $\mathcal{R}$ be a lfqa. For any $s = x_1x_2 \ldots x_m$, $s \in \Sigma^*$, assume k paths for accepting the string s in this automaton. Let P be the set of paths. $P = \{w_1, w_2 \ldots, w_k\}$. Now assuming p is another path for accepting the string s and $p \notin P$.

The paths for accepting the string s are as follows:

$$w_i = q_0(x_1, a_{1i})\, q_{1i}(x_2, a_{2i})\, q_{2i} \ldots q_{m-1,i}(x_m, a_{mi})\, q_{mi},$$

$$a_{1i}, a_{2i} \ldots, a_{mi} \in L,\ 1 \leq i \leq k$$

$$p = q_0(x_1, b_1)\, q_1(x_2, b_2)\, q_2 \ldots q_{m-1}(x_m, b_m)\, q_f,$$

$$b_1, b_2 \ldots, b_m \in L, \quad q_{mi}, q_f \in T$$

For type A lfqa,

$$\text{Accept}_{\text{A},P}(\mathcal{R},s) = \text{Accept}_{w_1}(R,s) \cup \text{Accept}_{w_2}(R,s) \cup \ldots \cup \text{Accept}_{w_k}(R,s)$$

$$\text{Accept}_{\text{A},P\cup\{p\}}(R,s) = \text{Accept}_{w_1}(R,s) \cup \ldots \cup \text{Accept}_{w_k}(R,s) \cup \text{Accept}_{p}(R,s)$$

It is obvious that $\text{Accept}_{\text{A},P}(\mathcal{R},s) \le \text{Accept}_{\text{A},P\cup\{p\}}(\mathcal{R},s)$ holds.

For type B lfqa, we first prove a general inclusion relation:

$$\text{If for each } i, b_i \le a_i, \text{ then for each } m, b_1 \cap b_2 \cap \ldots \cap b_m \le a_1 \cap a_2 \cap \ldots \cap a_m \tag{6}$$

The proof of this relation is easily done by mathematical induction if we consider the fact $b_1 \cap b_2 = b_1 \cap a_1 \cap b_2 \cap a_2 = b_1 \cap b_2 \cap a_1 \cap a_2 \le a_1 \cap a_2$, where we have made use of the rules of associativity and commutativity. Note that

$$\text{Accept}_{\text{B},P}(\mathcal{R},s) = (a_{11} \cup a_{12} \cup \ldots \cup a_{1k}) \cap (a_{21} \cup a_{22} \cup \ldots \cup a_{2k}) \cap \ldots$$
$$\cap (a_{m1} \cup a_{m2} \cup \ldots \cup a_{mk})$$

$$\text{Accept}_{\text{B},P\cup\{p\}}(\mathcal{R},s) = (a_{11} \cup a_{12} \cup \ldots \cup a_{1k} \cup b_1) \cap (a_{21} \cup a_{22} \cup \ldots$$
$$\cup a_{2k} \cup b_2) \cap \ldots \cap (a_{m1} \cup a_{m2} \cup \ldots \cup a_{mk} \cup b_m)$$

So we have $\text{Accept}_{\text{B},P}(\mathcal{R},s) \le \text{Accept}_{\text{B},P\cup\{p\}}(\mathcal{R},s)$ by considering (6). □

3. THE PUMPING PROPERTIES

Definition 3.1. (Pumping Property of Lattice-Valued Quantum Automata). A lattice-valued finite state quantum automaton $\mathcal{R}$ is said to have pumping property, if there exists a positive integer n, which depends only on $\mathcal{R}$ such that for each input string $s \in \Sigma^*$, with $|s| > n$ and $\text{Accept}(\mathcal{R},s) = a$, where a belongs to the lattice l, it is always possible to decompose s in $s = uvw$, such that $|uv| \le n$, $|v| \ge 1$, and for each $i \ge 1$, $\text{Accept}(\mathcal{R}, uv^i w) = a$.

Theorem 3.1. *Each deterministic finite state quantum automaton has pumping property.*

Proof: The idea is similar to that in the case of a usual finite state automaton. Let n be the number of the states of the quantum automaton. For any accepted string s whose length is larger than n, the path it traverses contains at least two equal states q_1 and q_2. Thus, the segment between q_1 and q_2 forms a loop, which can be repeated arbitrary times. That means s can be divided in $s = uvw$, where v corresponds to the loop, such that for each positive i, $uv^i w$ is also accepted by the quantum automaton. Because of its deterministic character, for each i there is only one path $p(i)$, which accepts $uv^i w$, and the acceptance degree is equal to the join of all lattice values attached to the segments of $p(i)$. Thus, to go a loop more is just to repeat the join of the same lattice values once more. It won't change the total acceptance degree. Therefore, we see the classical pumping lemma is also valid in the sense of deterministic lfqa. □

From the proof given above we get a more strict form of pumping lemma.

Definition 3.2. A lattice-valued finite state quantum automaton R is said to have strict pumping property, if the positive integer mentioned in Theorem 3.1 is at most equal to the number of states of the automaton R.

Corollary 3.1. *Each deterministic finite state quantum automaton has the strict pumping property.*

For the usual finite state automata, the strict pumping lemma always holds no matter whether it is deterministic or not. But this is not true in case of non-deterministic lattice-valued finite state quantum automata, because there may be more than one path that accepts the same string uv^iw. Let $p(i, j)$ denote the j-th path accepting uv^iw. Then the existence of $p(i, j)$ does not guarantee the existence of $p(i + 1, j)$, and vice versa. This fact can be shown by the following theorem.

Theorem 3.2. *Neither type A, nor type B lattice-valued finite state quantum automata have strict pumping property.*

Proof: Consider the lfqa R depicted in Fig. 2: where q_0 is the initial state and q_f is the final state. There are in total six states.

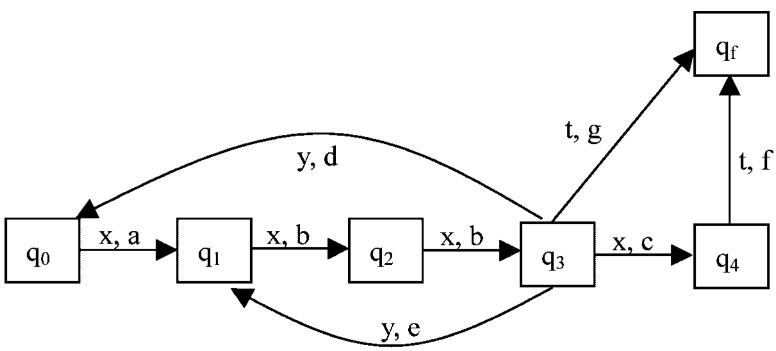

Fig. 2. A lfqa not having strict pumping property.

We consider the string $s = x^3yx^3t$, whose length equals to 8, larger than the number of states. The string s is accepted by R on two paths w_1 and w_2, where

$$\begin{aligned} w_1 &= q_0(x,a)\,q_1(x,b)\,q_2(x,b)\,q_3(y,d)\,q_0(x,a)\,q_1(x,b) \\ &\quad q_2(x,b)\,q_3(t,g)q_f \\ \text{Accept}_{w_1}(R,s) &= a \cap b \cap d \cap g \\ w_2 &= q_0(x,a)\,q_1(x,b)\,q_2(x,b)\,q_3(y,e)\,q_1(x,b)\,q_2(x,b) \\ &\quad q_3(x,c)\,q_4(t,f)\,q_f \\ \text{Accept}_{w_2}(R,s) &= a \cap b \cap c \cap e \cap f \end{aligned}$$

It is easy to see that there is no other path, which accepts s. Therefore, when this automaton is interpreted as type A lfqa, we have

$$\text{Accept}_A(R,s) = (a \cap b \cap d \cap g) \cup (a \cap b \cap c \cap e \cap f) \tag{7}$$

On the other hand, when it is interpreted as type B lfqa, we have:

$$\begin{aligned} \text{Accept}_\text{B}(R,s) &= a \cap b \cap (d \cup e) \wedge (a \cup b) \wedge (b \cup c) \cap (g \cup f) \\ &= a \cap b \cap (d \cup e) \cap (g \cup f) \end{aligned} \tag{8}$$

We will prove that it is impossible to represent $s = x^3yx^3t$ in form of uvw, such that $s(i) = uv^iw$ is accepted for arbitrary i with the same acceptance degree as s is accepted.

Assume this conclusion were not true. That means the desired decomposition of s in uvw were possible and $\text{Accept}_\text{A}(R, s(i)) = \text{Accept}_\text{A}(R, s)$, $\text{Accept}_\text{B}(R, s(i)) = \text{Accept}_\text{B}(R, s)$ for all $i > 0$. Then v must correspond to some loop of the automaton.

We will check all possible loops in R. From Fig. 2 we see that R contains only two loops. We denote them with Loop (above): $q_0q_1q_2q_3q_0$ and Loop (below): $q_1q_2q_3q_1$, respectively. Each accepted input string traverses the two loops Loop (above) and/or Loop (below) finite times. The general form of each accepted string is:

$$x^3 \prod_{i=1}^{k} [(yx^3)^{n_i}(yx^2)^{m_i}]x^jt \tag{9}$$

where $j = 0$ or 1, and n_i, m_i are non-negative integers, $k = 0, 1, 2, 3, \ldots$

The first half, $(yx^3)^{n_i}$, of each bracketed term, represents n_i times traversing Loop (above) while the second half, $(yx^2)^{m_i}$, represents m_i times traversing Loop (below) . In case that all m_i equal to 0 and $j = 0$, or when $m_k = 1$, and for all

$i < k, m_i = 0, j = 1$, the formula (3) becomes:

$$x^3 \prod_{i=1}^{k} (yx^3)^{n_i} t = x^3 (yx^3)^k t \quad k = 0, 1, 2, 3, \ldots \tag{10}$$

The lfqa of Fig. 2 contains two loops with the following seven possible representations:

$$xxxy, xxyx, xyxx, yxxx, xxy, xyx, yxx.$$

The possible decompositions of s in uvw conforming to these loop representations are the following seven, where in each case, the substring v is contained in a pair of parentheses:

$$\begin{aligned} s &= (x^3 y)x^3 t = x(x^2 yx)x^2 t = x^2(xyx^2)xt = x^3(yx^3)t = x(x^2 y)x^3 t \\ &= x^2(xyx)x^2 t = x^3(yx^2)xt \end{aligned} \tag{11}$$

We use the notation $s(i, j)$ to denote the power i pumping of the j-th loop representation of s in (11). For example, $s(2, 5) = x(xxy)^2 x^3 t$. We noticed that these seven loop representations can be classified in two equivalent groups. The first four loop representations form a group, since.

$$s(i, 1) = (x^3 y)^i x^3 t = (x^3 y)(x^3 y)^{i-1} x^3 t = x^3 (yx^3)^i t = s(i, 4)$$
$$s(i, 2) = x(x^2 yx)^i x^2 t = x(x^2 yx)(x^2 yx)^{i-1} x^2 t = x^3 (yx^3)^i t = s(i, 4)$$
$$s(i, 3) = x^2 (xyx^2)^i xt = x^2 (xyx^2)(xyx^2)^{i-1} xt = x^3 (yx^3)^i t = s(i, 4)$$

Let us have a closer look at the loop representation $s(i, 4)$ and check which paths it will traverse when it is accepted by the automaton. There are two paths accepting $s(i, 4)$. The first path $w_1(i, 4)$ can be illustrated as:

$$\begin{aligned} w_1(i, 4) = q_0(x, a)\, q_1(x, b)\, q_2(x, b)\, [q_3(y, d) \\ q_0(x, a)\, q_1(x, b)\, q_2(x, b)]^i q_3(t, g) q_f \end{aligned}$$

It traverses Loop (above) i-times, but makes no use of Loop (below). The second path $w_2(i, 4)$ makes use of Loop (below) and can be illustrated as

$$\begin{aligned} w_2(i, 4) = q_0(x, a)\, q_1(x, b)\, q_2(x, b)\, [q_3(y, d)\, q_0(x, a)\, q_1(x, b)\, q_2(x, b)]^{i-1} \\ [q_3(y, e)\, q_1(x, b)\, q_2(x, b)] q_3(x, c)\, q_4(t, f) q_f \end{aligned}$$

Note that $w_2(i, 4)$ traverses Loop (below) only once and only after all Loop (above) traverses are made, because traversing Loop (below) twice would produce a path containing the substring yx^2y, which does not appear in $s(i, 4)$. On the other hand, the same contradiction would happen if we perform a Loop (above) after a Loop (below).

It is also easy to check that these are the only two paths accepting $s(i,4)$. Therefore, for all $i > 1$,

$$\text{Accept}_{w_1}(\mathcal{R}, s(i,4)) = a \cap b \cap d \cap g$$

$$\text{Accept}_{w_2}(\mathcal{R}, s(i,4)) = a \cap b \cap c \cap d \cap e \cap f$$

In summary, for all $i > 1$,

$$\text{Accept}_{\text{A}}(\mathcal{R}, s(i,4)) = (a \cap b \cap d \cap g) \cup (a \cap b \cap c \cap d \cap e \cap f) \tag{12}$$

$$\begin{aligned}\text{Accept}_{\text{B}}(\mathcal{R}, s(i,4)) &= a \cap b \cap d \cap (d \cup e) \cap (a \cup b) \cap (b \cup c) \cap (g \cup f) \\ &= a \cap b \cap d \cap (g \cup f)\end{aligned} \tag{13}$$

We have to prove that for $i > 1$,

$$\text{Accept}_{\text{A}}(\mathcal{R}, s(i,4)) \neq \text{Accept}_{\text{A}}(\mathcal{R}, s)$$

$$\text{Accept}_{\text{B}}(\mathcal{R}, s(i,4)) \neq \text{Accept}_{\text{B}}(\mathcal{R}, s)$$

To this end, we construct the lattice l, on which the lfqa $\mathcal{R}$ is based, in the way as it is shown in Fig. 3:
Thus, for $i > 1$,

$$\text{Accept}_{\text{A}}(\mathcal{R}, s(i,4)) = k \neq a = \text{Accept}_{\text{A}}(\mathcal{R}, s), \tag{14}$$

$$\text{Accept}_{\text{B}}(\mathcal{R}, s(i,4)) = k \neq a = \text{Accept}_{\text{B}}(\mathcal{R}, s) \tag{15}$$

Above we have proved the theorem for $s(i,1), s(i,2), s(i,3), s(i,4)$. The remaining cases are $s(i,5)$, $s(i,6)$ and $s(i,7)$. They also form an equivalent group, since:

$$s(i,5) = x(x^2y)^i x^3 t = x(x^2y)(x^2y)^{i-1}x^3 t = x^3(yx^2)^i xt = s(i,7)$$

$$s(i,6) = x^2(xyx)^i x^2 t = x^2(xyx)(xyx)^{i-1}x^2 t = x^3(yx^2)^i xt = s(i,7)$$

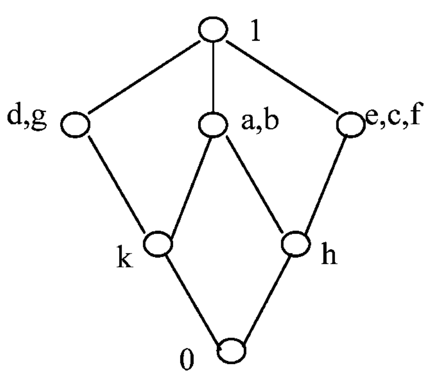

Fig. 3. The lattice l destroying the strict pumping property.

Therefore we have only to consider the case $s(i, 7)$. We will check which paths it will traverse when it is accepted by the automaton. There are two paths $w_1(i, 7)$ and $w_2(i, 7)$ which accept $s(i, 7)$. $w_1(i, 7)$ avoids Loop (above) and traverses only Loop (below):

$$w_1(i, 7) = q_0(x, a)\, q_1(x, b)\, q_2(x, b)\, [q_3(y, e)\, q_1(x, b)\, q_2(x, b)]^i\, q_3(x, c)\, q_4(t, f) q_f$$

It traverses Loop (below) i-times, but makes no use of loop (above). The second path $w_2(i, 7)$ makes use of both Loop (above) and Loop (below) and can be illustrated as

$$w_2(i, 7) = q_0(x, a)\, q_1(x, b)\, q_2(x, b)\, [q_3(y, e)\, q_1(x, b)\, q_2(x, b)]^{i-1}\, q_3(y, d)\, q_0(x, a)\, q_1(x, b)\, q_2(x, b)\, q_3(t, g)\, q_f$$

Note that $w_2(i, 7)$ traverses Loop (above) only once and only after all Loop (below) traverses are made, because traversing Loop (above) twice would produce a path containing the substring yx^3y, which does not appear in $s(i, 7)$. On the other hand, the same contradiction would happen if we perform a Loop (above) before a Loop (below).

It is also easy to check that these are the only two paths accepting $s(i, 7)$. Therefore, if $i > 1$,

$$\text{Accept}_{w_1}(R, s(i, 7)) = a \cap b \cap c \cap e \cap f$$
$$\text{Accept}_{w_2}(R, s(i, 7)) = a \cap b \cap d \cap e \cap g$$

In summary, for $i > 1$,

$$\text{Accept}_\text{A}(R, s(i, 7)) = (a \cap b \cap c \cap e \cap f) \cup (a \cap b \cap d \cap e \cap g) \quad (16)$$

$$\begin{aligned}\text{Accept}_\text{B}(R, s(i, 7)) &= (a \cap b \cap b) \cap (e \cap b \cap b)^{i-1} \cap (d \cup e) \cap (a \cup b)\\ &\quad \cap (b \cup b) \cap (b \cup c) \cap (g \cup f)\\ &= a \cap b \cap e \cap (d \cup e)\\ &\quad \cap (a \cup b) \cap (b \cup c) \cap (g \cup f)\\ &= a \cap b \cap e \cap (g \cup f) \end{aligned} \quad (17)$$

We have to prove that for $i > 1$,

$$\text{Accept}_\text{A}(R, s(i, 7)) \neq \text{Accept}_\text{A}(R, s),$$
$$\text{Accept}_\text{B}(R, s(i, 7)) \neq \text{Accept}_\text{B}(R, s),$$

Consider Fig. 3 once again. We get for $i > 1$,

$$\text{Accept}_\text{A}(R, s(i, 7)) = h \neq a = \text{Accept}_\text{A}(R, s) \quad (18)$$

$$\text{Accept}_{\text{B}}(R, s(i, 7)) = h \neq a = \text{Accept}_{\text{B}}(R, s) \tag{19}$$

We have now examined all possible cases and seen as result that for all $s(i, j)$ with $i > 1$ and $1 \leq j \leq 7$, $\text{Accept}_{\text{A}}(R, s(i, j)) \neq \text{Accept}_{\text{A}}(R, s)$, and $\text{Accept}_{\text{B}}(R, s(i, j)) \neq \text{Accept}_{\text{B}}(R, s)$. The breaking of pumping property for lfqa R in Fig. 2 is thus proved. □

But this example only shows that the integer n, where n is the number of states of the automaton, is not big enough for the pumping lemma to hold. We have not yet proved the proper pumping property for arbitrary lfqa. In order to show that, we have to extend our definition about pumping property a little bit. Namely, we need not only a definition for lfqa having pumping property. We need also a definition for an input string s having pumping property.

Definition 3.3. (Pumping property of an input string of a lattice-valued quantum automaton). An input string s of a lattice-valued finite state quantum automaton R is said to have pumping property, if s is accepted by R such that $\text{Accept}(R, s) = a$, where a belongs to the lattice l, and if it is possible to decompose s in $s = uvw$, such that $|v| \geq 1$, and for each $i \geq 1$, $\text{Accept}(R, uv^i w) = a$.

Proposition 3.2. *Any input string s accepted by the automaton R shown in Fig. 2, where the length of s is larger than $n = 12$, has the pumping property.*

Proof: For the lfqa R in Fig. 2, the general form of each accepted string is:

$$x^3 \prod_{i=1}^{k} [(yx^3)^{n_i} (yx^2)^{m_i}] x^j t$$

When $|s| > 12$, the forms of accepted string are as follows:

(1) $s = x^3[yx^3]^i x^j t$, when $j = 0, i \geq 3$; when $j = 1, i \geq 2$.
(2) $s = x^3[yx^2]^i x^j t$, when $j = 0, i \geq 3$; when $j = 1, i \geq 3$.
(3) $s = x^3 \prod_{i=1}^{k} [(yx^3)^{n_i} (yx^2)^{m_i}] x^j t$, and the accepted strings not only include the substring $[yx^3]$ but also the substring $[yx^2]$. That is to say both Loop (above) and Loop (below) must be traversed. More exactly, two copies of $[yx^3]$ or two copies of $[yx^2]$ should be in the accepted strings at least.

For the above three forms of accepted strings, we can iterate the reasoning steps one by one according to Theorem 3.2 and conclude that the accepted strings have the pumping property. □

Since we cannot yet prove the pumping property in general, we generalize its definition in the following way.

Definition 3.4. (Super (Sub)-Pumping Property). A lattice-valued finite state quantum automaton R is said to have super-pumping property, if there exists a positive integer n, which depends only on R such that for each input string $s \in \Sigma^*$, with $|s| > n$ and Accept$(R, s) = a$, where a belongs to the lattice l, it is always possible to decompose s in $s = uvw$, such that $|uv| \le n$, $|v| \ge 1$, and

$$\text{for each } i \ge 1, \text{Accept}(R, uv^i w) \ge a. \tag{20}$$

R is said to have sub-pumping property, if the inequality (21) holds instead of (20):

$$\text{for each } i \ge 1, \text{Accept}(R, uv^i w) \le a. \tag{21}$$

Definition 3.5. (Periodic Pumping Property). A lattice-valued finite state quantum automaton R is said to have periodic pumping property, if there exists a positive integer n, which depends only on R such that for each input string $s \in \Sigma^*$, with $|s| > n$, it is always possible to decompose s in $s = uvw$, such that $|v| \ge 1$, and there is a number $m > 0$ such that,

$$\text{for each } i \ge m, \text{there is a } j > i, \text{Accept}(R, uv^j w) = \text{Accept}(R, uv^m w) = a.$$

R is said to have periodic super-pumping property, if

$$\text{for each } i \ge m, \text{there is a } j > i, \text{Accept}(R, uv^j w) \ge \text{Accept}(R, uv^m w) = a.$$

R is said to have periodic sub-pumping property, if

$$\text{for each } i \ge m, \text{there is a } j > i, \text{Accept}(R, uv^j w) \le \text{Accept}(R, uv^m w) = a.$$

R is said to have periodic monotonic super-pumping property, if

$$\begin{aligned}\text{for each } i \ge m \text{ and Accept}(R, uv^i w) &\ge \text{Accept}(R, uv^m w)\\ &= \text{a, there is a } j > i, \text{Accept}(R, uv^j w) \ge \text{Accept}(R, uv^i w).\end{aligned}$$

R is said to have periodic monotonic sub-pumping property, if

$$\begin{aligned}\text{for each } i \ge m \text{ and Accept}(R, uv^i w) &\le \text{Accept}(R, uv^m w)\\ &= a, \text{ there is a } j > i, \text{Accept}(R, uv^j w) \le \text{Accept}(R, uv^i w).\end{aligned}$$

Both periodic monotonic super-pumping property and periodic monotonic sub-pumping property are called periodic monotonic property, too.

Definition 3.6. We define a loop of a lfqa R as a segment sequence, which starts from some state q of R and ends also in q. A loop is called Jordanian, if each state of it other than the starting state (ending state) q appears only once when the loop is traversed. Thus, $q_1x_1q_1x_2q_1$ is not a Jordanian loop, whereas $q_1x_1q_2x_2q_1$ is a Jordanian loop.

Proposition 3.3.

1. *All Jordanian loops of the same path form a total order.*
2. *There are only finitely many different Jordanian loops in a lfqa.*

It is easy to prove that on the same path p there are no two different Jordanian loops with the same starting state, since otherwise we would have two Jordanian loops $q_0x_1q_1\ldots x_kq_k\ldots q_{m-1}x_mq_m$ *and* $q_0x_1q_1\ldots x_kq_k$*, where* $0 < k < m$ *and* $q_0 = q_k = q_m$*, which contradicts the definition of a Jordanian loop. In this way, we can define a total order of all Jordanian loops on each path.*

On the other hand, it is also easy to prove that for any lfqa R*, there are only finitely many Jordanian loops on its paths. To be convinced of this, we just note that each Jordanian loop consists of finitely many nonrepeating segments. The total number of segments in a lfqa is finite. Therefore the possibility of their nonrepeating combinations is also finite, from which it follows the number of different Jordanian loops is also finite.*

Theorem 3.3. *All lfqa of type A have periodic monotonic super-pumping property.*

Proof: Let R be a type A lfqa with n states. Let $s = x_1x_2\ldots x_m$ be an input string accepted by R, where $m > n$. We do the following steps to prove the theorem:

Step 1. Find a loop on the initial accepting path.

Let $p = q_0x_1q_1x_2q_2\ldots q_{m-1}x_mq_m$ be one of the paths of R accepting s with $\text{Accept}_p(R, s) = a$. Then there must be at least two states q_j and q_k on p with $j < k$, such that $q_j = q_k$. This shows that $Q = q_jx_{j+1}q_{j+1}\ldots q_{k-1}x_kq_k$ forms a loop of the path p. Among all possible loops of p, we choose its first Jordanian loop. We have shown above that this is always possible.

Step 2. Pump the initial input string to get a new and enough large input string.

Let $v = x_{j+1}\ldots x_k$, $u = x_1x_2\ldots x_j$, $w = x_{k+1}\ldots x_m$, then all uv^iw with $i \geq 1$ are accepted by R on the paths $p(i) = q_0x_1q_1\ldots(q_jx_{j+1}q_{j+1}\ldots q_{k-1}x_k)^iq_k\ldots q_{m-1}x_mq_m$ with the same acceptance degree as $s = uvw$ is accepted by R on p. In particular, the path $p(m) = q_0x_1q_1\ldots(q_jx_{j+1}q_{j+1}\ldots q_{k-1}x_k)^mq_k\ldots q_{m-1}x_mq_m$ accepts uv^mw with $\text{Accept}_{p(m)}(R, uv^mw) = \text{Accept}(R, s) = a$.

Step 3. Find all paths accepting this new string and calculate its acceptance degree.

Note that $p(m)$ may be not the only path accepting uv^mw. Assume there are r paths accepting uv^mw, $r > 1$. For each fixed m, r must be finite. Assume these paths are ordered in an arbitrary, but fixed way. Denote them with $p(m, i)$, $1 \leq i \leq r$, where $p(m, 1) = p(m)$. Thus $p(m, i)$ means the i-th one among all paths accepting $uv^m w$. Each $p(m, i)$ must have the form

$$q_0^i x_1 q_1^i \ldots \left(q_j^{i1} x_{j+1} q_{j+1}^{i1} \ldots .q_{k-1}^{i1} x_k\right)\left(q_j^{i2} x_{j+1} q_{j+1}^{i2} \ldots .q_{k-1}^{i2} x_k\right) \ldots \times \left(q_j^{im} x_{j+1} q_{j+1}^{im} \ldots .q_{k-1}^{im} x_k\right) q_k^i \ldots q_{m-1}^i x_m q_m^i. \tag{22}$$

Assume they accept $uv^m w$ with the following acceptance degrees

$$\text{Accept}_{p(m,i)}(R, uv^m w) = a(i), \quad 1 \leq i \leq r \tag{23}$$

Where $a(1) = a$. Thus,

$$\text{Accept}(R, uv^m w) = \cup_{1 \leq i \leq r} a(i) \tag{24}$$

In each $p(m, i)$ there are m segments $(q_j^{i1} x_{j+1} q_{j+1}^{i1} \ldots .q_{k-1}^{i1} x_k)(q_j^{i2} x_{j+1} q_{j+1}^{i2} \ldots . q_{k-1}^{i2} x_k) \ldots (q_j^{im} x_{j+1} q_{j+1}^{im} \ldots .q_{k-1}^{im} x_k)$. We call each segment a loop unit. Each loop unit contains the same substring of input symbols: $x_{j+1} \ldots x_k$.

Step 4. Pump the group of accepting paths collectively to get a new group of paths.

Now consider the m states $(q_j^{i1}, q_j^{i2}, \ldots, q_j^{im})$ on $p(m, i)$. Among them there must be at least two states $q_j^{ih(i)}$ and $q_j^{ig(i)}$ with $h(i) < g(i)$, such that $q_j^{ih(i)} = q_j^{ig(i)}$, and such that for any f' and g' with $h(i) \leq f' < g' \leq g(i)$, from $q^{if'} = q^{ig'}$ it always follows that $f' = h(i)$ and $g' = g(i)$. Similar to Proposition 3.2, we can also prove that such loops form a total order for each $p(m, i)$. We choose the first one of them for each i.

$$\text{Let dis}\,(i) = g(i) - h(i), \quad 1 \leq i \leq r \tag{25}$$

$$\text{Let}\, H(p(m)) = \text{lcm}\{\text{dis}(i) | 1 \leq i \leq r\} \tag{26}$$

Where lcm means the least common multiple. On the basis of the construction above, the value of $H(p(m))$ is uniquely determined given automaton R and path $p(m)$.

$$\text{Let}\, G(p(m), i) = H(p(m))/\text{dis}(i) \tag{27}$$

Let $v(i)$ be the symbol string, which is obtained from that part of $p(m, i)$ between $q_j^{i,h(i)}$ and $q_j^{i,g(i)}$ by eliminating all states in it (remember, it is called the label of this path part). It is obvious that each $v(i)$ is contained in v^m. For each i, decompose $uv^m w$ in $u(i)v(i)w(i)$. For each path $p(m, i)$, repeat the loop body between $q_j^{i,h(i)}$ and $q_j^{i,g(i)}$ (consisting of $g(i) - h(i)$ loop units) for $G(p(m), i)$ times. The result is a path $p(m, i)'$ accepting the string $u(i)v(i)^{G(p(m),i)+1}w(i)$ with the same acceptance degree $a(i)$ as $u(i)v(i)w(i)$ is accepted on the path

$p(m, i)$. Since $u(i)v(i)^{G(p(m),i)+1}w(i) = uv^{m+H(p(m))}w$, we can rewrite $p(m, i)'$ as:

$$p(m, i)' = p(m + H(p(m)), i), \tag{28}$$

Step 5. Show that the pumped string is accepted by the new group of paths at least as much as that string is accepted by the old group of paths before it is pumped.

We have

$$\begin{aligned} &\text{Accept}_{p(m.i)'}\big(R, u(i)v(i)^{G(p(m),i)+1}w(i)\big) \\ &\quad = \text{Accept}_{p(m+H(p(m)).i)}\big(R, uv^{m+H(p(m))}w\big) \\ &\quad = \text{Accept}_{p(m,i)}(R, uv^{m}w) = a(i), \quad 1 \le i \le r \end{aligned} \tag{29}$$

$$\text{Accept}_{p(m,i)',1\le i\le r}\big(R, uv^{m+H(p(m))}w\big) = \cup_{1\le i\le r}\, a(i) \tag{30}$$

This shows that there is a number $m + H(p(m)) > m$, such that the acceptance degree of $uv^{m+H(p(m))}w$ is at least equal to the acceptance degree of $uv^{m}w$. We say "at least," because it is not to exclude that there are still paths of R other than those of $p(m, i)'$ for accepting $uv^{m+H(p(m))}w$.

That means we have the following relation:

$$\text{Accept}\big(R, uv^{m+H(p(m))}w\big) \ge \text{Accept}\,(R, uv^{m}w) \tag{31}$$

Step 6. Repeat the steps 4 and 5 once and once again, obtaining in this way an infinite sequence of strings with monotonic nondecreasing acceptance degree, where each string is produced as a pumped string of its previous one.

We take $uv^{m+H(p(m))}w$ as the new input string. Further we take $p(m, 1)'$ as the new accepting path and start the reasoning process above once again. Namely, we calculate the number $H(p(m, 1)')$. If there are other paths accepting $uv^{m+H(p(m))}w$, then we call them $p(m, j)''$ where $1 \le j \le r'$, r' is the number of these paths (note that r' may be zero). We consider the set of all paths $P = \{p(m, i)', p(m, j)''|1 \le i \le r, 1 \le j \le r'\}$ accepting $uv^{m+H(p(m))}w$ and calculate the numbers $G(p(m, 1)', t)$, $1 \le t \le r + r'$. By repeating the loop units we then get a set of new paths, which accept the string $uv^{m+H(p(m,1))+H(p(m,1)')}w$ with an acceptance degree at least equal to the acceptance degree $uv^{m+H(p(m))}w$ is accepted by the set of paths P.

Therefore, in each case (whether or not $p(m, i)'$ are the only paths accepting $uv^{m+H(p(m))}w$) we have found a number $m' > m + H(p(m))$ such that

$$\text{Accept}\,(R, uv^{m'}w) \ge \text{Accept}\,\big(R, uv^{m+H(p(m))}w\big) \tag{32}$$

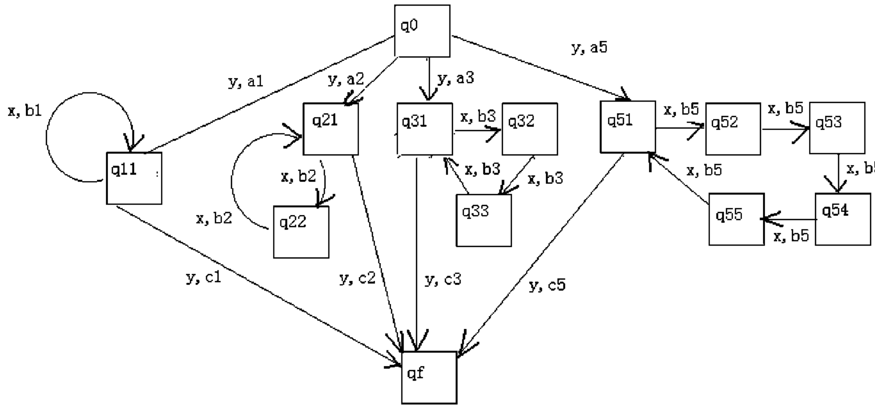

Fig. 4. Lfqa with monotonic pumping property.

In this way, we have proved the existence of a chain $m_1, m_2, m_3, m_4, \ldots$, such that

$$m = m_1 < m_2 < m_3 < m_4 < \ldots,$$

$$\text{Accept}\,(R, uv^{m_1}w) \le \text{Accept}\,(R, uv^{m_2}w) \le \text{Accept}\,(R, uv^{m_3}w) \le \ldots \quad (33)$$

This is just the periodic monotonic super-pumping property of R we wanted to prove. □

Example 3.1. Assume the lfqa has the following form (Fig. 4):

$R = (Q = \{q_0, q_{11}, q_{21}, q_{22}, q_{31}, q_{32}, q_{33}, q_{51}, q_{52}, q_{53}, q_{54}, q_{55}, q_f\}, I = \{q_0\}, T = \{q_f\}, \ \Delta = \{\delta(q_0, y, q_{i1}) = a_i, \delta(q_{i1}, y, q_f) = c_i, \delta(q_{ii}, x, q_{i1}) = b_i, \delta(q_{jk}, x, q_{j,k+1}) = b_j$; where $i = 1, 2, 3, 5; j = 2, 3, 5; k = 1, \ldots, j-1\})$

We follow the steps described in the proof of Theorem 3.3.

Step 1.

The number of states of R is $n = 13$. This automaton accepts the input string $s(1) = y(x)^{17}y$, whose length m is 19, which is larger than 13. There is only one path $p_1 = q_0 y(q_{11}x)^{17} q_{11} y q_f$ of R, which accepts $s(1)$ with the acceptance degree $a_1 \cap b_1 \cap c_1$. The first Jordanian loop on p_1 is $q_{11}xq_{11}$, such that we can decompose s in uvw with $u = y, v = x, w = x^{16}$y.

Step 2.

From the 19th power of the part v, we get the string $s(2) = uv^{19}w = y(x)^{35}y$.

Step 3.

The new string $s(2)$ will not only be accepted on the path $p(19, 1) = q_0 y (q_{11}x)^{35} q_{11} y q_f$, but also on the path $p(19, 2) = q_0 y(q_{51}xq_{52}xq_{53}xq_{54}xq_{55}x)^7 q_{51} y q_f$. Thus we have

$$\text{Accept}\,(\mathcal{R}, s(1)) \leq \text{Accept}\,(\mathcal{R}, s(2)) = (a_1 \cap b_1 \cap c_1) \cup (a_5 \cap b_5 \cap c_5).$$

Step 4.

Consider the loop $L_1 = q_{11}xq_{11}$ on $p(19,1)$ and the loop $L_2 = q_{51}xq_{52}xq_{53}xq_{54}xq_{55}xq_{51}$ on $p(19,2)$. We have $h(1) = 2, g(1) = 3, h(2) = 2, g(2) = 7$, $\text{dis}(1) = 1$, $\text{dis}(2) = 5$, $H(p(19)) = \text{lcm}\{1, 5\} = 5$, $G(p(19), 1) = 5$, $G(p(19), 2) = 1$. That means if we repeat the loop L_1 five times then we get $p(24,1) = q_0 y(q_{11}x)^{40} q_{11} y q_f$ from $p(19,1)$, and if we repeat the loop L_2 once then we get $p(24,2) = q_0 y(q_{51}xq_{52}xq_{53}xq_{54}xq_{55}x)^8 q_{51} y q_f$ from $p(19,2)$. In this way we get two new paths, both of which accept the string $s(3) = uv^{24}w = y(x)^{40}y$ with the same acceptance degree as $s(2)$ was accepted by $\mathcal{R}$.

Step 5.

The reasoning stated above shows that $\mathcal{R}$ accepts $s(3)$ at least as much as it accepts $s(2)$. As a matter of fact, $\mathcal{R}$ accepts $s(3)$ more than accepting $s(2)$, because $s(3)$ is also accepted by $p(24,3) = q_0 y(q_{21}xq_{22}x)^{20} q_{21} y q_f$. Thus we have

$$\text{Accept}\,(\mathcal{R}, s(2)) \leq \text{Accept}\,(\mathcal{R}, s(3)) = (a_1 \cap b_1 \cap c_1) \cup (a_2 \cap b_2 \cap c_2) \cup (a_5 \cap b_5 \cap c_5).$$

Step 6.

Consider the loop $L_3 = q_{21}xq_{22}xq_{21}$ on $p(24,3)$. We have $h(3) = 2$, $g(3) = 4$, $\text{dis}(3) = 2$, $H(p(24)) = \text{lcm}\{1,5,2\} = 10$, $G(p(24),1) = 10$, $G(p(24),2) = 2$, $G(p(24),3) = 5$. That means if we repeat the loop L_1 10 times then we get $p(34,1) = q_0 y(q_{11}x)^{50} q_{11} y q_f$ from $p(24,1)$, and if we repeat the loop L_2 twice then we get $p(34,2) = q_0 y(q_{51}xq_{52}xq_{53}xq_{54}xq_{55}x)^{10} q_{51} y q_f$ from $p(24,2)$, and if we repeat the loop L_3 five times then we get $p(34,3) = q_0 y(q_{21}xq_{22}x)^{25} q_{21} y q_f$ from $p(24,3)$. In this way we get three new paths, all of which accept the string $s(4) = uv^{34}w = y(x)^{50}y$ with the same acceptance degree as $s(3)$ was accepted by $\mathcal{R}$. That means $\mathcal{R}$ accepts $s(4)$ at least as much as it accepts $s(3)$. A simple investigation shows that $s(3)$ and $s(4)$ have the same acceptance degree since this time the number of accepting paths has not been increased by pumping ($p(34,1)$, $p(34,2)$, and $p(34,3)$ are the only paths accepting $s(4)$). That means we have

$$\text{Accept}\,(\mathcal{R}, s(3)) = \text{Accept}\,(\mathcal{R}, s(4))$$

Pumping the three loops L_1, L_2, and L_3 once again we get three new paths accepting $s(5) = uv^{44}w = y(x)^{60}y$. They are $p(44,1) = q_0y(q_{11}x)^{60}q_{11}yq_f$, $p(44,2) = q_0y(q_{51}xq_{52}xq_{53}xq_{54}xq_{55}x)^{12}q_{51}yq_f$ and $p(44,3) = q_0y(q_{21}xq_{22}x)^{30}$ $q_{21}yq_f$.

Now there is no equality between Accept $(R, s(4))$ and Accept $(R, s(5))$, because now there is a fourth path $p(44,4) = q_0y(q_{31}xq_{32}xq_{33}x)^{20}q_{31}yq_f$, which also accepts $s(5)$. It is

$$\text{Accept}_{P(44,4)}(R, s(5)) = (a_3 \cap b_3 \cap c_3)$$

Therefore, the relations between the acceptance degrees are:

$$\begin{aligned}\text{Accept}\,(R, s(4)) &= (a_1 \cap b_1 \cap c_1) \cup (a_2 \cap b_2 \cap c_2) \cup (a_5 \cap b_5 \cap c_5)\\ &\le (a_1 \cap b_1 \cap c_1) \cup (a_2 \cap b_2 \cap c_2) \cup (a_3 \cap b_3 \cap c_3)\\ &\cup (a_5 \cap b_5 \cap c_5) = \text{Accept}\,(R, s(5))\end{aligned}$$

Consider the loop $L_4 = q_{31}xq_{32}xq_{33}xq_{31}$ on $p(44,4)$. We have $h(4) = 2$, $g(4) = 5$, $\text{dis}(4) = 3$, $H(p(44)) = \text{lcm}\{1, 5, 2, 3\} = 30$, $G(p(44), 1) = 30$, $G(p(44), 2) = 6$, $G(p(44), 3) = 15$, $G(p(44), 4) = 10$. That means if we repeat the loop L_1 30 times then we get $p(74,1) = q_0y(q_{11}x)^{90}q_{11}yq_f$ from $p(44,1)$, and if we repeat the loop L_2 six times then we get $p(74,2) = q_0y(q_{51}xq_{52}xq_{53}xq_{54}xq_{55}x)^{18}q_{51}yq_f$ from $p(44,2)$, if we repeat the loop L_3 15 times then we get $p(74,3) = q_0y(q_{21}xq_{22}x)^{45}q_{21}yq_f$ from $p(44,3)$, and if we repeat the loop L_4 10 times then we get $p(74,4) = q_0y(q_{31}xq_{32}xq_{33}x)^{30}q_{31}yq_f$ from $p(44,4)$. In this way we get four new paths, all of which accept the string $s(6) = uv^{74}w = y(x)^{90}y$ with the same acceptance degree as $s(7)$ is accepted. From the definition of R it is easy to see that there is no other path accepting $s(6)$. Therefore we have

$$\text{Accept}\,(R, s(5)) = \text{Accept}\,(R, s(6))$$

It is also easy to check that in general the following relation holds:

$$\text{Accept}\,(R, s(5)) = \text{Accept}\,(R, s(t)), \quad \text{for all} \quad t > 5, \tag{34}$$

where $s(t) = uv^{30t-106}w = y(x)^{30t-90}y$ for all $t > 5$.

We have thus proved the existence of a sequence of accepted input strings $s(t)$ with monotonically nondecreasing acceptance degrees, where each $s(t)$ has the form $uv^{f(t)}w$, where $s(1) = uvw$ is the original input string.

Lemma 3.1. *For any given lfqa R, the set of acceptance degrees it may take when accepting input strings is finite.*

Proof: If the lattice l, on which R is based, is finite, then the conclusion is obvious. Otherwise, we reason as follows. Each lfqa has only finitely many (transition) arcs. On each arc there are only finitely many attached (input symbol, lattice value) pairs.

In case of type A lfqa, the acceptance degree of each input string s is calculated as the disjunction of finitely many conjunctions, where each conjunction consists of finitely many lattice values. In case of type B lfqa, the acceptance degree of each input string s is calculated as the conjunction of finitely many disjunctions, where each disjunction consists of finitely many lattice values. In order to avoid ambiguity, we call the former a second level disjunction of finitely many first level conjunctions, and the latter as a second level conjunction of finitely many first level disjunctions.

Since the number of different lattice values appearing on the transitions of any given automaton is finite, the number of their different combinations (conjunctions) is also finite. That means the number of different values a first level conjunction may produce is finite.

Similarly, we can prove that the number of different values a first level disjunction may produce is finite.

For a type A lfqa, the acceptance degree is calculated as the second level disjunction of finitely many first level conjunctions, which themselves have only finitely many possible values. For a type B lfqa, the acceptance degree is calculated as the second level conjunction of finitely many first level disjunctions, which themselves have only finitely many possible values. Therefore, for both type A lfqa and type B lfqa, the set of acceptance degrees it may take when accepting input strings is finite.

It is easy to complete the proof by using mathematical induction. □

With help of this lemma, we can prove a more exact theorem.

Theorem 3.4. *All lfqa of type A have periodic pumping property.*

Proof: In Theorem 3.3 we have proved the periodic monotonic super-pumping property for all type A lfqa with the existence of an ascending chain $m_1, m_2, m_3, m_4, \ldots$ and a nondescending chain (33). Now we start from the number $m_1 = m$ and do the reasoning procedure in the following way:

$$\text{Let } n_1 = m_1.$$

If for all m_k, whenever $k > 1$ and Accept $(R, uv^k w) \geq$ Accept $(R, uv^m w)$, it is always Accept $(R, uv^{m_k} w) =$ Accept $(R, uv^{m_1} w)$, then the periodic pumping property is already proved.

Otherwise, there must be a $k > 1$ with Accept$(R, uv^{m_k} w) \neq$ Accept$(R, uv^{m_1} w)$

$$\text{Let } n_2 = m_k.$$

If for all j, whenever $j > k$, it is always

$$\text{Accept}\,(\mathcal{R}, uv^{m_j}w) = \text{Accept}\,(\mathcal{R}, uv^{m_k}w), \tag{35}$$

then the periodic pumping property is already proved.

Otherwise, there must be a $j > k$ such that Accept $(\mathcal{R}, uv^{m_j}w) \neq$ Accept $(\mathcal{R}, uv^{m_k}w)$

$$\text{Let } n_3 = m_j.$$

Since according to Lemma 3.1 there are only finitely many acceptance degree values, this procedure of constructing a strictly ascending super-pumping chain must terminate after finitely many steps. That means, the n_i cannot form an infinite sequence. The conclusion we can draw is that there is an infinite ascending sequence of positive integers $p_1, p_2, \ldots$ (which is a subsequence of the sequence $m_1, m_2, \ldots$), such that

$$\text{Accept}(\mathcal{R}, uv^{p_j}w) = \text{Accept}(\mathcal{R}, uv^{p_1}w) \quad \text{for all } j > 0. \qquad \square$$

In fact, a more powerful theorem is provable based on the following:

Example 3.2. The lfqa given in Example 3.1 has periodic pumping property. To be convinced of this fact, let the number m in Theorem 3.3 equal to 60, then for each $i \geq m$, we just let j equal to the least multiple of 30, which is larger than i.

Definition 3.7. (Generalized Pumping Property). A lattice-valued finite state quantum automaton $\mathcal{R}$ is said to have generalized pumping property, if there exists a positive integer n, which depends only on $\mathcal{R}$ such that for each input string $s \in \Sigma^*$, with $|s| > n$, it is always possible to decompose s in $s = uvw$, such that $|v| \geq 1$, and there exists a number $M > 0$, such that for each $i \geq 1$, Accept $(\mathcal{R}, uv^{iM}w) =$ Accept$(\mathcal{R}, uv^{M}w)$.

$\mathcal{R}$ is said to have generalized super-pumping property, if

$$\text{Accept}\,(\mathcal{R}, uv^{M}w) \leq \text{Accept}\,(\mathcal{R}, uv^{iM}w) \tag{36}$$

holds instead. $\mathcal{R}$ is said to have generalized sub-pumping property, if

$$\text{Accept}\,(\mathcal{R}, uv^{iM}w) \leq \text{Accept}\,(\mathcal{R}, uv^{M}w) \tag{37}$$

holds instead.

Two obvious conclusions can be drawn from this definition. First, in case of $m = 1$, the generalized pumping property is equal to the pumping property defined in Definition 2.1. Second, generalized pumping property is a strengthened form of periodic pumping property.

Theorem 3.5. *All lfqa of type A have generalized pumping property.*

Proof: The main idea of proving this theorem is already contained in the proof of Theorem 3.3. But here we will go another way. We do not want to iterate the reasoning steps as we did there. Rather, we will use a global and concise proof procedure by summarizing the idea presented in Theorem 3.3.

Step 1. Find a loop.

Let $\mathcal{R}$ be a lfqa of type A with n states. Let $s = x_1x_2 \ldots x_m$ be an input string accepted by $\mathcal{R}$, where $m > n$. Let $p = q_0x_1q_1x_2q_2 \ldots q_{m-1}x_mq_m$ be one of the paths of $\mathcal{R}$ accepting s with $\text{Accept}_p(\mathcal{R}, s) = a$. Then we can always find two states q_j and q_k on p with $j < k$, such that $q_j = q_k$, and such that for all states $q_h = q_g$ with $j \leq h < g \leq k$, it is always $h = j$ and $g = k$. This shows that $Q = q_jx_{j+1}q_{j+1} \ldots q_{k-1}x_kq_k$ forms a Jordanian loop of the path p. W.l.o.g., we assume Q is the first Jordanian loop of p.

Step 2. First pump.

Let $v = x_{j+1} \ldots x_k$, $u = x_1x_2 \ldots x_j$, $w = x_{k+1} \ldots x_m$, then all uv^iw with $i \geq 1$ are accepted by $\mathcal{R}$ on the paths $p(i) = q_0x_1q_1 \ldots (q_jx_{j+1}q_{j+1} \ldots q_{k-1}x_k)^i q_k \ldots q_{m-1}x_mq_m$ with the same acceptance degree as $s = uvw$ is accepted by $\mathcal{R}$ on p. In particular, the path $p(f) = q_0x_1q_1 \ldots (q_jx_{j+1}q_{j+1} \ldots q_{k-1}x_k)^f q_k \ldots q_{m-1}x_mq_m$ accepts uv^fw with $\text{Accept}_{p(f)}(\mathcal{R}, uv^fw) = \text{Accept}_p(\mathcal{R}, s) = a$, where $f = \text{lcm}\{t | 1 \leq t \leq n+1\}$.

Let the set of paths accepting uv^fw be $p(f, i)$, $1 \leq i \leq r$. Then each $p(f, i)$ must have the form $q_0^ix_1q_1^i \ldots (q_j^{i1}x_{j+1}q_{j+1}^{i1} \ldots q_{k-1}^{i1}x_k)$

$$\left(q_j^{i2}x_{j+1}q_{j+1}^{i2} \ldots q_{k-1}^{i2}x_k\right) \ldots \left(q_j^{if}x_{j+1}q_{j+1}^{if} \ldots q_{k-1}^{if}x_k\right)q_k^i \ldots q_{m-1}^ix_mq_m^i. \quad (38)$$

Let

$$\text{Accept}_{p(f,i)}(\mathcal{R}, uv^fw) = a(i), \quad 1 \leq i \leq r \quad (39)$$

Thus

$$\text{Accept}\,(\mathcal{R}, uv^fw) = \cup_{1 \leq i \leq r}a(i) \quad (40)$$

where $p(f, 1) = p(f)$ and $\text{Accept}_{p(f)}(\mathcal{R}, uv^fw) = a = a(1)$.

Step 3. Find a group of loops.

For each i, consider the f states $(q_j^{i1}, q_j^{i2}, \ldots, q_j^{if})$ of the path $p(f, i)$. Since $f > n$ (the number of states), there must be at least two states $q_j^{ih(i)}$ and $q_j^{ig(i)}$ on $p(f, i)$ with $1 \leq h(i) < g(i) \leq n+1$, such that $q_j^{ih(i)} = q_j^{ig(i)}$, and such that for any f' and g' with $h(i) \leq f' < g' \leq g(i)$, from $q^{if'} = q^{ig'}$ it always follows that

f' and g' are equal to $h(i)$ and $g(i)$. That part $v(i)$ of the path $p(f, i)$, which is between $q_j^{ih(i)}$ and $q_j^{ig(i)}$, forms a loop. For each i, we consider a loop with least $g(i)$. If there are more than one loop with least $g(i)$, we take that one among them with least $h(i)$. It is easy to prove that this loop is uniquely determined for each i. Since $g(i) \leq n + 1$ the number $g(i) - h(i)$ must be a divisor of f.

Step 4. Pump iteratively—each time enlarges the accepting paths by f loop bodies.

Repeat the part $v(i)$ for $f/(g(i) - h(i))$ times for each i, we get a new path $p(2f, i)$ out of $p(f, i)$. Each of the paths $p(2f, i)$ accepts the same string $uv^{2f}w$ with the same acceptance degree as $uv^f w$ is accepted by $p(f, i)$.

It is then easy to see that the string $uv^{jf}w$ with arbitrary $j \geq 1$ is accepted by each path $p(jf, i)$ with

$$\text{Accept}_{p(jf,i)}(R, uv^{jf}w) = a(i), \quad 1 \leq i \leq r \tag{41}$$

where $p(jf, i)$ is produced by pumping the path $p((j-1)f, i)$ when repeating the loop $(q_j^{ih(i)} x_{j+1} q_{j+1}^{ih(i)} \ldots q_{k-1}^{ih(i)} x_k) \ldots (q_j^{i(g(i)-1)} x_{j+1} q_{j+1}^{i(g(i)-1)} \ldots q_{k-1}^{i(g(i)-1)} x_k)$ for $f/\text{dis}(i)$ times.

Step 5. Conclude the proof with a chain of relations.

There may be other paths, which also accept $uv^{jf}w$. Therefore, in general it is

$$\text{Accept}(R, uv^f w) \leq \text{Accept}(R, uv^{jf}w), \quad \text{for any } j \geq 1 \tag{42}$$

This shows that R satisfies the generalized super-pumping property. But this is not yet all. Because of the finiteness of the number of different acceptance degree values in a lfqa, there must be a number $J > 0$, such that

$$\text{Accept}(R, uv^{jf}w) = \text{Accept}(R, uv^{Jf}w), \quad \text{for any } j \geq J \tag{43}$$

In particular,

$$\text{Accept}\,(R, uv^{jJf}w) = \text{Accept}\,(R, uv^{Jf}w), \quad \text{for any } j \geq 1 \tag{44}$$

Rename Jf as M, this is the generalized pumping property described in Definition 3.7. □

Example 3.3. The lfqa given in Example 3.1 satisfies the generalized pumping property. To be convinced of this fact, calculate $f = \text{lcm}\{t | 1 \leq t \leq 14\} = 360360$. It is easy to see that $J = 1$ is enough to make the following relation valid:

$$\text{Accept}\,(R, uv^{jf}w) = \text{Accept}\,(R, uv^{Jf}w), \quad \text{for any } j \geq J$$

Thus it is enough to let $M = f = 360360$.

In fact, the number 360360 is unnecessarily too large in this particular case. We only need to let the number M equal to 60, then for each $i \geq 1$, we have

$$\text{Accept}\,(R, uv^{jM}w) = \text{Accept}\,(R, uv^{M}w), \quad \text{for any } j \geq 1$$

This example shows that the characteristic number M, and therefore the acceptance degree of the input sequence, is not uniquely determined.

Above we have discussed various pumping properties of type A lfqa: the periodic monotonic super-pumping property, the periodic pumping property and the generalized pumping property. Now what about type B lfqa? Do they have the same pumping properties? The answer to this question is not trivial, because we have used the technique of loop repeating in our proof procedure. This may cause a mismatching of acceptance degrees along different paths, which accept the same input string, see Example 3.4.

Example 3.4. In Fig. 5, the input string $s = xyyx$ is accepted on both paths $p_1 = q_0xq_1yq_1yq_2xq_f$ and $p_2 = q_0xq_3yq_4yq_3xq_f$. Whereas s is accepted on p_1 with the degree $a \cap b \cap c \cap d$, it is accepted on p_2 with $e \cap f \cap g \cap h$. The total (combined) acceptance degree is then $(a \cup e) \cap (b \cup f) \cap (c \cup g) \cap (d \cup h)$. If we repeat the loop q_1yq_1 of p_1 twice and the loop $q_3yq_4yq_3$ of p_2 once, then we get two new paths p_1' and p_2'. Each of these two paths accepts the string $s' = x(yy)^2x$ with the same acceptance degree as p_1 and p_2 accepted the string $xyyx$ respectively. It is therefore obvious that the combined acceptance degree of s' remains the same as that of s was, if the lfqa is of type A. But the total (i.e. the combined) acceptance degree of s' is $(a \cup e) \cap (b \cup f) \cap (b \cup g) \cap (b \cup f) \cap (c \cup g) \cap (d \cup h)$ if the lfqa is of type B. Let $a = c = d = f = 1$, then s is accepted with the degree 1, but s' is accepted with the degree $(b \cup g)$.

This shows the difference between type A and type B automata. As a result, we cannot copy the proof procedure of type A for type B. Fortunately, this difference will not prohibit us from proving a theorem for type B lfqa similar to that of type A lfqa.

Theorem 3.6.

All lfqa of type B have periodic monotonic super-pumping property
All lfqa of type B have periodic pumping property
All lfqa of type B have generalized pumping property

Proof: The first three proof steps are similar to those in the proof of Theorem 3.5. But Step 4 has to be changed to: pump iteratively—each time enlarges the accepting paths by $2f$ loop bodies. That means, we get a new path $p(3f, i)$ out of $p(f, i)$ by repeating the loop body $v(i)$ for $2f/(g(i) - h(i))$ times for each i.

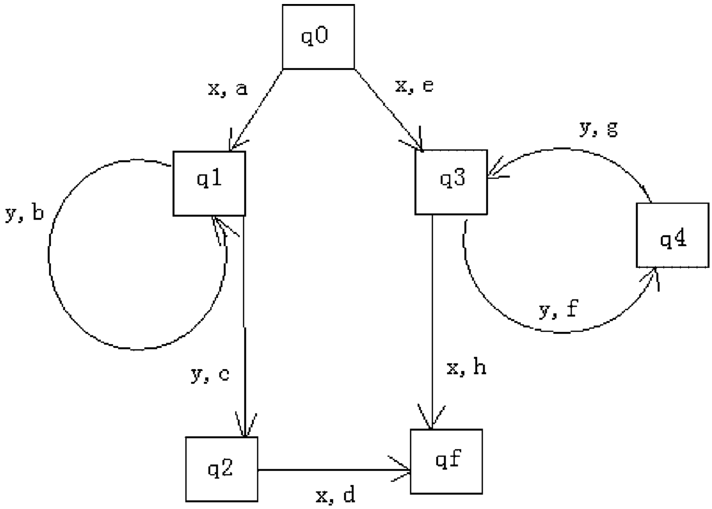

Fig. 5. Repeating loops may cause mismatch of acceptance degrees.

In order to calculate the acceptance degrees of input strings, assume that for any n, the number of paths accepting $uv^n w$ is $r(n)$ and

$$\text{Accept}_{p(n,i),1\leq i\leq r(n)}(\mathcal{R}, uv^n w) = a(n),$$

where

$$\begin{aligned} a(n) &= a1(n) \wedge a2(n) \wedge a3(n), \\ &\text{Accept}_{p(n,i),1\leq i\leq r(n)}(\mathcal{R}, u, uv^n w) = a1(n), \\ &\text{Accept}_{p(n,i),1\leq i\leq r(n)}(\mathcal{R}, v^n, uv^n w) = a2(n), \\ &\text{Accept}_{p(n,i),1\leq i\leq r(n)}(\mathcal{R}, w, uv^n w) = a3(n), \end{aligned}$$

where

$$\text{Accept}_P(\mathcal{R}, s_i, s_1 s_2 \ldots s_m) = a$$

means that the (partial) acceptance degree of s_i by the path set P is a, where s_i is a substring of the input string $s = s_1 s_2 \ldots s_m$, which is accepted by P.

Now we consider the path set $p(3f, i)$, $1 \leq i \leq r(3f)$. Let

$$i_{\max} = \text{Min}\,\{i|1 \leq i \leq r(3f)\,\text{and}\,g(i) = \text{Max}\,\{g(t)|1 \leq t \leq r(3f)\}\}$$

then we have: for all i,

$$g(i) \leq g(i_{\max}) < g(i_{\max}) + f \leq g(i) + 2f \tag{45}$$

Assume that $g(i_{\max})$ corresponds to the p-th state of the path $p(3f, i_{\max})$. Since all paths of $p(3f, i)$ have the same length, it follows from relation (45) that each path $p(3f, i)$ contains a segment $q_p^i x_{p+1} q_{p+1}^i \dots q_{p+f-1}^i x_{p+f} q_{p+f}^i$, where $q_{p+f}^i = q_p^i$ for all i. The last assertion is guaranteed by the fact that f is a common multiple of all $g(i) - h(i)$. This shows that all paths have a loop $L_i = q_p^i x_{p+1} q_{p+1}^i \dots q_{p+f-1}^i x_{p+f} q_p^i$ of the same size and with the same indices.

Pumping these paths by repeating this loop will yield new paths. Each of them accepts the string $uv^{(t+3)f}w$ after t times of pumping.

By a simple reasoning similar to that in the proof of Theorem 3.5 it is then easy to see that the string $uv^{jf}w$ with arbitrary $j \geq 3$ is accepted at least by the path set $\{p(jf, i)|1 \leq i \leq r((j-1)f)\}$ where $\{p(jf, i)|1 \leq i \leq r((j-1)f)\}$ is produced by pumping the path set $\{p((j-1)f, i)|1 \leq i \leq r((j-1)f)\}$ simultaneously when repeating the loop L_i for $f/\text{dis}(i)$ times. Since each $p(jf, i)$ can be transformed in $p((j+1), i)$ only by increasing the number of loop L_i once more, we have

$$r(jf) \leq r((j+1)f)$$

In the following we will use the notation $\{p(jf, i)\}$ for $\{p(jf, i)|1 \leq i \leq r(jf)\}$.

$$\begin{aligned}&\text{Accept}_{\{(jf,i)\}}(R, L_i, uv^{jf}w)\\&\quad= \text{Accept}_{\{p((j+1)f,i)|1\leq i\leq r(jf)\}}(R, (L_i)^2, uv^{(j+1)f}w) \quad \text{for } j \geq 3\end{aligned}$$

therefore,

$$\text{Accept}_{\{(jf,i)\}}(R, u, uv^{jf}w) = \text{Accept}_{\{p((j+1)f,i)|1\leq i\leq r(jf)\}}\big(R, u, uv^{(j+1)f}w\big)$$

$$\text{Accept}_{\{(jf,i)\}}(R, v^{jf}, uv^{jf}w) = \text{Accept}_{\{p((j+1)f,i)|1\leq i\leq r(jf)\}}\big(R, v^{(j+1)f}, uv^{(j+1)f}w\big)$$

$$\text{Accept}_{\{(jf,i)\}}(R, w, uv^{jf}w) = \text{Accept}_{\{p((j+1)f,i)|1\leq i\leq r(jf)\}}\big(R, w, uv^{(j+1)f}w\big)$$

In general it is

$$\{p((j+1)f, i)|1 \leq i \leq r(jf)\} \subseteq \{p((j+1)f, i)\} \quad \text{for any } j \geq 3$$

therefore,

$$\begin{aligned}&\text{Accept}\,(R, uv^{jf}w) = \text{Accept}_{\{p(jf,i)\}}(R, uv^{jf}w) = \text{Accept}_{\{p((j+1)f,i)|1\leq i\leq r(jf)\}}\\&\big(R, u, uv^{(j+1)f}w\big) \leq \text{Accept}_{\{p((j+1)f,i)\}}\big(R, uv^{(j+1)f}w\big) = \text{Accept}\,(R, uv^{jf}w)\end{aligned} \tag{46}$$

In this way we get a chain of input strings $uv^{jf}w (j \geq 3)$ with monotonic non-decreasing acceptance degrees, where we have made use of Proposition 2.1. Thus the monotonic super-pumping property of type B quantum automata is proved. With help of Lemma 3.1 the validness of periodic pumping property for Type B quantum automata is also obvious.

That means there exists a number $J \geq 3$ such that

$$\text{Accept}\,(R, uv^{jf}w) = \text{Accept}\,(R, uv^{Jf}w) \quad j \geq J$$

Let $M = Jf$. It is then clear that

$$\text{Accept}\,(R, uv^{M}w) = \text{Accept}\,(R, uv^{jM}w) \quad j \geq 1$$

This is nothing else than the generalized pumping property of type B quantum automata. □

Example 3.5. Reconsider the lfqa discussed in Example 3.4. Calculate $f = \text{lcm}\{t|1 \leq t \leq 6\}12 = 30$. Let $s = xyyx = uvw$, where $u = x, v = y, w = yx$. Raise v to the 30th power, we have $uv^{30}w$. It is easy to see that there are two path accepting $uv^{30}w$, where $p(30,1) = p_1 = q_0x(q_1y)^{30}q_1yq_2xq_f$ and $p(30,2) = q_0x(q_3yq_4y)^{15}q_3xq_f$. Repeat the loop in $p(30,1)$ 60 times more and that in $p(30,2)$ 45 times more, we get the $3f = 90$th power of v and two paths $p(90,1) = p_1 = q_0x(q_1y)^{90}q_1yq_2xq_f$ and $p(90,2) = q_0x(q_3yq_4y)^{45}q_3xq_f$. We have $h(1) = 2$, $g(1) = 3$, $h(2) = 2$, $g(2) = 4$. Thus

$$i_{\max} = 2,\ g(i_{\max}) = 4,\ g(i_{\max}) + f = 34$$

If we unfold them in a plain sequence and rename the states as q' to avoid notational ambiguity, then

$$P(90,\,1) = q_0xq_1'y \ldots q_3'yq_4'yq_5' \ldots q_{31}'yq_{32}'yq_{33}' \ldots q_{91}yq_{92}xq_f.$$

$$P(90,\,2) = q_0xq_1''y \ldots q_3''yq_4''yq_5'' \ldots q_{31}''yq_{32}''yq_{33}'' \ldots Q_{91}''yq_{92}''xq_f$$

where

$$\text{for all}\, 0 < i < 92,\ q_i' = q_1,\ q_{92}' = q_2,$$

$$\text{for all}\, 0 < i < 92,\ q_i'' = q_3 \text{ if } i \text{ is odd}, \quad q_i'' = q_4 \text{ if } i \text{ is even.}$$

In particular we have $q_3' = q_{33}' =$ and $q_3'' = q_{33}''$.

This means that the two paths $p(90,1)$ and $p(90,2)$ have a loop of the same size and with the same indices. They are $q_3'yq_4'yq_5' \ldots q_{31}'yq_{32}'yq_{33}'$ and $q_3''yq_4''yq_5'' \ldots q_{31}''yq_{32}''yq_{33}''$ respectively. Repeating this loop on both paths simultaneously generates a sequence of path pairs, which accept the input sequence $uv^{jf}w(j \geq 3)$ with monotonically nondecreasing acceptance degrees. But we have got more. We have shown the periodic pumping property as well. Finally we have also the generalized pumping property by letting $M = 90$.

It may be adequate to give a note at this place: the periodic pumping property can be derived from the generalized pumping property. On the other hand, the monotonic super-pumping property can be derived from the periodic pumping property.

4. CONCLUSIONS

It is well known that the classical finite state automata satisfy the pumping lemma. So we want to see whether lattice-valued quantum automata also have pumping property or not. In this paper it is shown that not all lattice-valued quantum automata possess the strict pumping property. Then we have extended the definition of pumping property and proposed the concepts of super(sub)-pumping property, periodic pumping property, monotonic pumping property, and generalized pumping property. We have proved that all lattice-valued quantum automata (including those of type A and those of type B) have monotonic super-pumping property. In addition, it is proved that all lattice-valued quantum automata have periodic pumping property and generalized pumping property.

REFERENCES

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

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

Greechie, R. (1981). A non-standard quantum logic with a strong set of states. In *Current Issues in Quantum Logic*, *Vol. 8*, E. G. 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.

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

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

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

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

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