Invited Paper

Automata with Quantum and Classical Resources

Masaki Nakanishi†

Quantum automata have been studied as simple quantum computation models. They can be considered models of small (or restricted) quantum computers. In this paper, we give descriptions of several kinds of quantum automata and show their power in comparison to their classical counterparts. We also give descriptions of quantum automata that have additional classical computational resources. Introducing classical computational resources can enhance the power of quantum automata, since this approach relaxes such restrictions as reversible state transitions.

1. Introduction

Several kinds of quantum finite automata[4),7),9),12),13)] and quantum pushdown automata[6),9),11)] have been proposed as simple quantum computation models. In particular, some quantum finite automata can be implemented with a constant number of qubits, and thus they are important quantum computation models given the current scale of quantum computer development.

In this paper, we give descriptions of a variety of quantum finite automata and quantum pushdown automata and show results demonstrating the power of quantum automata in comparison to their classical counterparts.

We expect quantum automata to be more powerful than their classical counterparts. However, this is not always the case, since quantum computation models are required to obey such restrictions as reversible state transitions. In order to overcome such restrictions, introducing classical computational resources into quantum automata may be an effective solution. In some cases, such hybrid models can be more powerful than purely quantum models. In this paper we address two quantum-classical hybrid models[4),11)] and describe their power in language recognition.

This paper is organized as follows. In Section 2, we define several kinds of quantum finite automata and compare their power with their classical counterparts. In Section 3, we define quantum pushdown automata and analyze some results in language recognition. In Section 4, we describe two quantum-classical hybrid models: finite automata with quantum and classical states, and quantum pushdown automata with classical stack operations. Section 5 concludes this paper.

2. Quantum Finite Automata

In this section, we describe quantum finite automata and compare them with their classical counterparts.

Finite automata are the simplest computation models. Various kinds of finite automata can be defined according to head movements (one-way or two-way) and types of transitions (deterministic, non-deterministic, or probabilistic). We give a definition of two-way probabilistic finite automata below. Note that one-way or deterministic finite automata can be defined as a restricted model of the two-way probabilistic finite automata.

Definition 1 A two-way probabilistic finite automaton (2PFA) is defined as the following 6-tuple:

$$M = (Q, \Sigma, \delta, q_0, Q_{\rm acc}, Q_{\rm rej}),$$

where Q is a set of states, Σ is a set of input symbols including the left and right endmarkers $\{\not\mathrm{c}, \$\}$, respectively, δ is a state transition function ($\delta : (Q \times \Sigma \times Q \times \{-1, 0, +1\}) \longrightarrow [0, 1]$), q_0 is an initial state, $Q_{\rm acc}$ ($\subseteq Q$) is a set of accepting states, and $Q_{\rm rej}$ ($\subseteq Q$) is a set of rejecting states, where $Q_{\rm acc} \cap Q_{\rm rej} = \emptyset$. □

In the common definition, only a set of accepting states ($Q_{\rm acc}$) is given, while no set of rejecting states ($Q_{\rm rej}$) is given. However, in this paper, we also give a set of rejecting states in order to be consistent with the definition of quantum finite automata.

$\delta(q, a, q', D) = \alpha$ means that the probability of the transition from q to q' moving the head to D is α when reading input symbol a. Note that, for each input symbol, the sum of the weights

† Nara Institute of Science and Technology

(i.e., the probabilities) of outgoing transitions of a state must be 1. The computation of a finite automaton starts with initial state q_0 and the head position at the leftmost symbol. At every step, a finite automaton changes its state and the head position according to the state transition function, and the computation halts when it enters the accepting or rejecting states. If the computation halts at an accepting (resp. rejecting) state, it outputs 'accept' (resp. 'reject'). For language L, when a finite automaton accepts any string in L with probability at least $\frac{1}{2}+\varepsilon$ and rejects any string not in L with probability at least $\frac{1}{2}+\varepsilon$, we say that the automaton recognizes language L with probability $\frac{1}{2}+\varepsilon$, where ε is a constant that does not depend on an input.

We define a deterministic finite automaton as a finite automaton in which the image of a state transition function is restricted to {0,1}. We define a one-way finite automaton as a finite automaton in which head movements are restricted to '+1'.

Next, we define quantum finite automata. Quantum finite automata can be considered an extended model of probabilistic finite automata, where edge weights represent *amplitudes*. As in the case of classical finite automata, various kinds of quantum finite automata can be defined. We give the definition of (measure-many) two-way quantum finite automata below.

Definition 2 A two-way quantum finite automata (2QFA) is defined as the following 6-tuple:

$$M = (Q, \Sigma, \delta, q_0, Q_{\mathrm{acc}}, Q_{\mathrm{rej}}),$$

where Q is a set of states, Σ is a set of input symbols including the left and right endmarkers $\{\text{¢}, \$\}$, respectively, δ is a state transition function ($\delta : (Q \times \Sigma \times Q \times \{-1, 0, +1\}) \longrightarrow \mathbb{C}$), q_0 is an initial state, Q_{acc} ($\subseteq Q$) is a set of accepting states, and Q_{rej} ($\subseteq Q$) is a set of rejecting states, where $Q_{\mathrm{acc}} \cap Q_{\mathrm{rej}} = \emptyset$.

□

$\delta(q, a, q', D) = \alpha$ means that the amplitude of the transition from q to q' moving the head to D is α when reading an input symbol a. A configuration of a 2QFA is a pair (q, k), where k is the position of the head and q is in Q. A superposition of configurations of a 2QFA is any element of $l_2(Q \times \mathbb{Z}_n)$ of unit length, where $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$. For each configuration, we define a column vector $|q, k\rangle$ as follows:

- $|q, k\rangle$ is an $n|Q| \times 1$ column vector.
- The row corresponding to (q, k) is 1, and the other rows are 0.

For input string $\boldsymbol{x}$, we define time evolution operator $U^{\boldsymbol{x}}$ as follows:

$$U^{\boldsymbol{x}}(|q,k\rangle) = \sum_{q' \in Q, D \in \{-1,0,1\}} \delta(q, x(k), q', D)\,|q', k+D\rangle\,,$$

where $x(k)$ is the k-th input symbol of input $\boldsymbol{x}$. If $U^{\boldsymbol{x}}$ is unitary (for any input string $\boldsymbol{x}$), that is, $U^{\boldsymbol{x}}U^{\boldsymbol{x}\dagger} = U^{\boldsymbol{x}\dagger}U^{\boldsymbol{x}} = I$, where $U^{\boldsymbol{x}\dagger}$ is the transpose conjugate of $U^{\boldsymbol{x}}$, then the corresponding 2QFA is well-formed.

The computation of a quantum finite automaton starts with initial state q_0 and the head position at the leftmost symbol. We define $|\psi_0\rangle$ as $|\psi_0\rangle = |q_0, 0\rangle$. We also define E_{acc}, E_{rej}, and E_{non} as follows:

$$E_{\mathrm{acc}} = span\{|q,k\rangle\,|\,q \in Q_{\mathrm{acc}}\},$$
$$E_{\mathrm{rej}} = span\{|q,k\rangle\,|\,q \in Q_{\mathrm{rej}}\},$$
$$E_{\mathrm{non}} = span\{|q,k\rangle\,|\,q \in Q \setminus (Q_{\mathrm{acc}} \cup Q_{\mathrm{rej}})\}.$$

We define observable $\mathcal{O}$ as $\mathcal{O} = E_{\mathrm{acc}} \oplus E_{\mathrm{rej}} \oplus E_{\mathrm{non}}$. We also define the outcome of a measurement corresponding to E_{acc}, E_{rej} and E_{non} as 'acc', 'rej' and 'non', respectively.

The 2QFA computation proceeds as follows:

(a) $U^{\boldsymbol{x}}$ is applied to $|\psi_i\rangle$. Let $|\psi_{i+1}\rangle = U^{\boldsymbol{x}}\,|\psi_i\rangle$.

(b) $|\psi_{i+1}\rangle$ is measured with respect to observable $\mathcal{O}$. Let $|\phi_j\rangle$ be the projection of $|\psi_{i+1}\rangle$ to E_j, where j is 'acc', 'rej' or 'non'. Then each outcome j is obtained with probability $|\,|\phi_j\rangle\,|^2$. Note that this measurement causes $|\psi_{i+1}\rangle$ to collapse to $\frac{1}{|\,|\phi_j\rangle\,|}\,|\phi_j\rangle$, where j is the obtained outcome.

We call the above (a) and (b) 'one step' collectively. We repeat the above (a) and (b) until 'acc' or 'rej' is measured. The language recognized by a 2QFA is defined in the same way as that by a classical finite automaton.

We define a one-way quantum finite automaton (1QFA) as a quantum finite automaton in which head movements are restricted to '+1'. In this case, it can be shown that for a well-formed 1QFA, there exists a unitary operator U_σ for each $\sigma \in \Sigma$ such that

$$\delta(q, \sigma, q', +1) = \langle q'|\,U_\sigma\,|q\rangle\,.$$

Then we describe a configuration of a 1QFA as $|q\rangle$ ($q \in Q$). The 1QFA computation starts with the initial configuration $|q_0\rangle$ ($= |\psi_0\rangle$) and

proceeds as follows:

(a) U_{σ_i} is applied to $|\psi_i\rangle$, where σ_i is the i-th input symbol. Let $|\psi_{i+1}\rangle = U_{\sigma_i}|\psi_i\rangle$.

(b) $|\psi_{i+1}\rangle$ is measured with respect to the observable $\mathcal{O} = E_{\rm acc} \oplus E_{\rm rej} \oplus E_{\rm non}$, which is defined similarly as in the definition of 2QFAs.

We repeat (a) and (b) until we obtain 'acc' or 'rej'.

Note that a configuration of a 1QFA can be described by a basis vector of a finite dimensional Hilbert space. Thus, a 1QFA can be implemented with a constant number of qubits, while 2QFAs needs $O(\log n)$ qubits to be implemented, where n is the input length. This causes a computational gap between the one-way and the two-way models. The following theorems [7] show that 2QFAs are strictly more powerful than 1QFAs.

Theorem 1 [7] For any one-way deterministic finite automaton A, there exists a reversible two-way deterministic finite automaton M such that, for any $w \in \Sigma^*$, if A accepts w then M accepts w in $O(|w|)$ steps, and if A does not accept w, then M rejects w in $O(|w|)$ steps.

□

Note that a reversible deterministic finite automaton is a restricted quantum finite automaton whose transition amplitudes may only take the values 0 and 1, and also note that the class of languages recognized by one-way deterministic finite automata is the same as that recognized by polynomial-time two-way probabilistic finite automata. This class of languages is called *regular languages*. Thus Theorem 1 says that 2QFAs are at least as powerful as 2PFAs.

Theorem 2 [7] Let $w \in \{a, b\}^*$. For every positive integer N, there exists a 2QFA M_N such that if $w \in \{a^m b^m | m \geq 1\}$ then M_N accepts w with certainty, and otherwise M_N rejects w with probability at least $1 - \frac{1}{N}$. In either case M_N halts after $O(N|w|)$ steps with certainty.

□

Theorem 3 [7] The class of languages recognized by 1QFAs is a proper subset of regular languages.

□

Some other results concerning quantum finite automata should be reviewed. Nayak proposed enhanced one-way quantum finite automata [13], in which any orthogonal measurement can be used instead of $\mathcal{O} = E_{\rm acc} \oplus E_{\rm rej} \oplus E_{\rm non}$. It was shown that any bounded error enhanced one-way quantum finite automaton recognizing the language $\{wa | w \in \{a, b\}^*, |w| \leq n\}$ has $2^{\Omega(n)}$ states. Nakanishi et al. showed that nondeterministic quantum finite automata are strictly more powerful than classical non-deterministic finite automata [12], where a non-deterministic quantum finite automaton recognizing L is such that it accepts an input string in L with probability greater than 0 and rejects an input string not in L with certainty.

3. Quantum Pushdown Automata

Quantum pushdown automata were introduced by Moore and Crutchfield [9]. In their definition, state transitions of quantum pushdown automata are not necessarily described by unitary operators. Golovkins thus introduced unitarity to quantum pushdown automata and investigated their power [6].

Theorem 4 [6] Every regular language is recognizable by some QPA.

□

Theorem 5 [6] Language $L_{\rm eq} = \{w \in \{a, b, c\}^* | |w|_a = |w|_b = |w|_c\}$ is recognizable by a QPA with probability $\frac{2}{3}$, where $|w|_\sigma$ is the number of occurrences of symbol σ in w.

□

Theorem 6 [6] Language $L_{\rm xor} = \{w \in \{a, b, c\}^* | |w|_a = |w|_b \text{ xor } |w|_a = |w|_c\}$ is recognizable by a QPA with probability $\frac{4}{7}$.

□

Note that neither $L_{\rm eq}$ nor $L_{\rm xor}$ can be recognized by deterministic pushdown automata. The above results show that quantum pushdown automata can be more powerful than deterministic pushdown automata. However, it remains an open question whether quantum pushdown automata can be more powerful than probabilistic pushdown automata. Murakami et al. showed that in exact computation, quantum pushdown automata can be more powerful than classical pushdown automata [10].

Theorem 7 [10] There exists a promise problem that can be solved by quantum pushdown automata with certainty but cannot be solved by deterministic pushdown automata.

□

The other models similar to quantum pushdown automata are quantum counter automata and quantum multi-stack machines. Quantum counter automata were first proposed in Ref. 8). Comparison between 1-way quantum 1-

counter automata and 1-way classical 1-counter automata is discussed in Refs. 5), 15). Two-way quantum one-counter automata and 1-way quantum k-counter automata are investigated in Ref. 16). Also in Ref. 14), quantum multi-counter machines and quantum multi-stack machines are investigated in terms of simulation of quantum Turing machines.

4. Automata with Quantum and Classical Resources

Purely quantum automata may be less powerful than their classical counterparts because of restrictions such as reversible state transitions. For example, Theorem 3 says that purely quantum 1-way finite automata are less powerful than their classical counterparts. Introducing classical computational resources to quantum automata may relax such restrictions. In this section, we address two types of automata with quantum and classical resources: two-way finite automata with quantum and classical states [4], and quantum pushdown automata with classical stack operations [11]. Especially, later in this section, we focus on quantum pushdown automata with classical stack operations and explain in detail how classical devices (classical stacks in this case) work.

First, we describe two-way finite automata with quantum and classical states [4]. A two-way finite automaton with quantum and classical states (2QCFA) is a two-way finite automaton that has a quantum finite state control in addition to a classical finite state control; it has a classical input tape, a classical tape head, a classical finite state control, and a quantum finite state control.

The quantum portion is controlled by the classical portion; the classical state and the input symbol pointed by the classical tape head determine the unitary operator applied to the quantum portion. The classical portion is partly controlled by the result of the measurement of the quantum portion; the result of the measurement, in addition to the input symbol and the classical state, determines the state transition of the classical portion. Note that this model has a classical finite automaton as its component. Thus this model is at least as powerful as a finite automaton.

The following theorems are shown in Ref. 4).

Theorem 8 [4] For any $\varepsilon > 0$ there exists a 2QCFA M operating as follows. For any input $x \in \{a, b\}^*$, if x is a palindrome then M accepts x with certainty, and if x is not a palindrome then M accepts x with probability at most ε and rejects x otherwise.

□

Theorem 9 [4] For any $\varepsilon > 0$, there is a 2QCFA M that accepts any $x \in \{a^n b^n | n \in \mathbb{N}\}$ with certainty, rejects $x \notin \{a^n b^n | n \in \mathbb{N}\}$ with probability at least $1 - \varepsilon$ and halts in expected time $O(m^4)$ where m is the length of the string x.

□

It is known that the language in Theorem 8 cannot be recognized by 2PFAs. It is also known that the language in Theorem 9 cannot be recognized by 2PFAs in polynomial time. Thus these results show that 2QCFAs are strictly more powerful than 2PFAs.

Next, we describe a pushdown automaton that has a quantum finite state control and a classical stack. In purely quantum pushdown automata, stack operation is highly restricted since *pop* operation is a deleting operation and deletion is not a reversible operation. Sacrificing utilization of superposition on a stack (i.e., implementing a stack with a classical device), a quantum pushdown automaton with classical stack operations can control its stack without such restrictions.

A quantum pushdown automaton with classical stack operations (QCPDA) has an input tape to which a quantum head is attached and a classical stack to which a classical stack top pointer is attached. A QCPDA has a quantum finite state control. The quantum finite state control reads the stack top symbol pointed by the classical stack top pointer and the input symbol pointed by the quantum head. Stack operations are determined solely by the results of measurements of a quantum finite state control. We define QCPDAs formally as follows.

Definition 3 A Quantum Pushdown Automaton with Classical Stack Operations (QCPDA) is defined as the following 8-tuple:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, \sigma, Q_{\mathrm{acc}}, Q_{\mathrm{rej}}),$$

where Q is a set of states, Σ is a set of input symbols including the left and right endmarkers $\{\not\vdash, \$\}$, respectively, Γ is a set of stack symbols including the bottom symbol Z, δ is a quantum state transition function ($\delta : (Q \times \Sigma \times \Gamma \times Q \times \{0, 1\}) \longrightarrow \mathbb{C}$), q_0 is an initial state, σ is a function by which stack operations are determined ($\sigma : Q \backslash (Q_{\mathrm{acc}} \cup Q_{\mathrm{rej}}) \longrightarrow \Gamma^+ \cup \{-, pop\}$), Q_{acc} ($\subseteq Q$) is a set of accepting states, and

$Q_{\rm rej}$ ($\subseteq Q$) is a set of rejecting states, where $Q_{\rm acc} \cap Q_{\rm rej} = \emptyset$. We impose a restriction that the length of any pushed string is finite. As another restriction, for all q, q', a, D, if $\sigma(q') = pop$, then $\delta(q, a, Z, q', D) = 0$.

□

$\delta(q, a, b, q', D) = \alpha$ means that the amplitude of the transition from q to q' moving the quantum head to D ($D = 1$ means 'right' and $D = 0$ means 'stay') is α when reading input symbol a and stack symbol b. The configuration of the quantum portion of a QCPDA is a pair (q, k), where k is the position of the quantum head and q is in Q.

For input string $\boldsymbol{x}$ and stack symbol a, we define a time evolution operator $U_a^{\boldsymbol{x}}$ as follows:

$$U_a^{\boldsymbol{x}}(|q,k\rangle) = \sum_{q' \in Q, D \in \{0,1\}} \delta(q, x(k), a, q', D)\,|q', k+D\rangle\,,$$

where $x(k)$ is the k-th input symbol of input $\boldsymbol{x}$. If $U_a^{\boldsymbol{x}}$ is unitary (for any $a \in \Gamma$ and for any input string $\boldsymbol{x}$), that is, $U_a^{\boldsymbol{x}} U_a^{\boldsymbol{x}\dagger} = U_a^{\boldsymbol{x}\dagger} U_a^{\boldsymbol{x}} = I$, then the corresponding QCPDA is well-formed. A well-formed QCPDA is considered valid in terms of the quantum theory. We consider only well-formed QCPDAs.

Now we describe how the quantum portion and the classical stack of a QCPDA work.

Let the initial quantum state and the initial position of the head be q_0 and '0', respectively. We define $|\psi_0\rangle$ as $|\psi_0\rangle = |q_0, 0\rangle$. We also define E_w, $E_{\rm acc}$ and $E_{\rm rej}$ as follows:

$$E_w = span\{|q,k\rangle\,|\sigma(q) = w\},$$
$$E_{\rm acc} = span\{|q,k\rangle\,|q \in Q_{\rm acc}\}, \text{and}$$
$$E_{\rm rej} = span\{|q,k\rangle\,|q \in Q_{\rm rej}\}.$$

We define observable $\mathcal{O}$ as $\mathcal{O} = \oplus_j E_j$, where j is $w \in \Gamma^+ \cup \{-, pop\}$, 'acc', or 'rej'. For notational simplicity, we define the outcome of a measurement corresponding to E_j as j.

The QCPDA computation proceeds as follows:

For input string $\boldsymbol{x}$, the quantum portion works as follows:

(a) $U_a^{\boldsymbol{x}}$ is applied to $|\psi_i\rangle$. Let $|\psi_{i+1}\rangle = U_a^{\boldsymbol{x}}\,|\psi_i\rangle$, where a is the stack top symbol.

(b) $|\psi_{i+1}\rangle$ is measured with respect to the observable $\mathcal{O} = \oplus_j E_j$. Let $|\phi_j\rangle$ be the projection of $|\psi_{i+1}\rangle$ to E_j. Then each outcome j is obtained with probability $||\phi_j\rangle|^2$. Note that this measurement causes $|\psi_{i+1}\rangle$ to collapse to $\frac{1}{||\phi_j\rangle|}\,|\phi_j\rangle$, where j is the obtained outcome. Then go to (c).

The classical stack works as follows:

(c) Let the outcome of the measurement be j. If j is 'acc' ('rej', resp.), then it outputs 'accept' ('reject', resp.), and the computation halts. If j is '$-$', then the stack is unchanged. If j is 'pop', then the stack top symbol is popped. Otherwise, string j is pushed (j is a string in Γ^+ in this case). Then go to (a) and repeat.

We call the above (a), (b), and (c) 'one step' collectively.

For language L, if there exists a constant ε ($0 \le \varepsilon < 1$) that does not depend on inputs, a QCPDA accepts any string in L with a probability greater than $1 - \varepsilon$, and it rejects any string that is not in L with certainty, then we say that L is recognized by the QCPDA with one-sided error.

For simplicity, we handle only a subclass of QCPDAs, called *simplified* QCPDAs, so that we can decompose the quantum state transition function into two functions: one for changing states and the other for moving the quantum head. For $a \in \Sigma$ and $b \in \Gamma$, we adopt a linear operator $V_{a,b} : l_2(Q) \longrightarrow l_2(Q)$ for changing states and a function $\Delta : Q \longrightarrow \{0, 1\}$ for moving the quantum head. In *simplified* QCPDAs, the direction of the movement of the head is determined solely by the state to which the current state makes a transition. Then transition function δ is described as follows:

$$\delta(q, a, b, q', D) = \begin{cases} \langle q'|\,V_{a,b}\,|q\rangle & (\Delta(q') = D) \\ 0 & (\Delta(q') \neq D), \end{cases}$$

where $\langle q'|\,V_{a,b}\,|q\rangle$ is the coefficient of $|q'\rangle$ in $V_{a,b}\,|q\rangle$.

The condition of being well-formed for simplified QCPDAs is shown in Ref. 11).

Theorem 10 [11] A simplified QCPDA is well-formed if, for any $a \in \Sigma$ and $b \in \Gamma$, the linear operator $V_{a,b}$ satisfies the following condition:

$$\sum_{q'} \overline{\langle q'|\,V_{a,b}\,|q_1\rangle}\,\langle q'|\,V_{a,b}\,|q_2\rangle = \begin{cases} 1 & (q_1 = q_2) \\ 0 & (q_1 \neq q_2). \end{cases}$$

□

It is straightforward to see that (even deterministic) pushdown automata can recognize the language $\{x\%x^R | x \in \{a,b\}^*\}$. However, it is known that the language $\{x\%x | x \in \{a,b\}^*\}$ is

not recognized by non-deterministic pushdown automata. This is because of the 'Last-in/First-out' manner of the stack structure. However, QCPDAs can overcome this difficulty with appropriate *hints* in an input string. It is known that QCPDAs can recognize the following language L_1 with one-sided error, while probabilistic pushdown automata cannot do this with one-sided error [11].

$$L_1 = \left\{ u\#v\natural w\flat x\%y \;\middle|\; \begin{array}{l} u,v,w,x,y \in \{a,b\}^*, \\ \neg\left(\begin{array}{l} |u| = |v| = |w| = |x| \\ \text{and } v = w \end{array} \right) \\ \text{and } \neg\left(\begin{array}{l} |u| = |x| \text{ and} \\ |v| = |w| = 0 \end{array} \right) \\ \text{and } (y = v^R \text{ or } y = w^R) \end{array} \right\}.$$

Note that an automaton that recognizes L_1 may have to check whether $v = w$. Informally, this is why classical pushdown automata cannot recognize L_1 with one-sided error. In the following, we show that with the condition $|u| = |v| = |w| = |x|$, which we use as a hint, QCPDAs can recognize L_1 with one-sided error.

Theorem 11 [11] QCPDAs can recognize L_1 with one-sided error.

(Proof)

We show QCPDA M, which recognizes L_1. An outline of M is as follows. M rejects any input that is not of the form $u\#v\natural w\flat x\%y = \{a,b\}^*\#\{a,b\}^*\natural\{a,b\}^*\flat\{a,b\}^*\%\{a,b\}^*$, and it consists of three components M_1, M_2, and M_3, each of which is a deterministic automaton. M_1 processes substring $u\#v\natural w\flat x$ as follows:

(1) M_1 reads u taking $2|u|$ steps with the stack unchanged,
(2) pushes each symbol of v one by one,
(3) and reads w and x taking $|w| + |x|$ steps with the stack unchanged.

M_2 processes substring $u\#v\natural w\flat x$ as follows:

(1) M_2 reads u and v taking $|u| + |v|$ steps with the stack unchanged,
(2) pushes each symbol of w one by one,
(3) and reads x taking $2|x|$ steps with the stack unchanged.

M_3 processes substring y checking whether the string on the stack is y^R or not.

M_1 and M_2 process substring $u\#v\natural w\flat x\%$ in parallel using superposition. First, we consider the case that ($|u| = |v| = |w| = |x|$ and $v = w$) or ($|u| = |x|$ and $|v| = |w| = 0$). In this case, M_1 and M_2 push the same symbol at the same time (or no symbol is pushed in the case $|u| = |x|$ and $|v| = |w| = 0$) and also read '%' at the same time. Thus, when reading '%', M_1 and M_2 interfere with each other and move to the rejecting state with certainty (See **Fig. 1**).

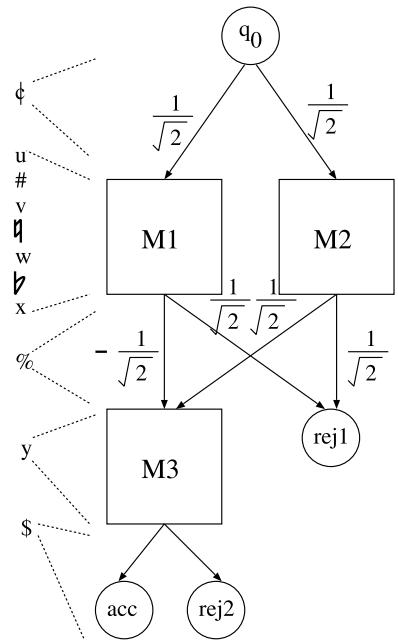

Fig. 1 QCPDA M.

Second, we consider the case that $\neg(|u| = |v| = |w| = |x|$ and $v = w)$ and $\neg(|u| = |x|$ and $|v| = |w| = 0)$. In this case, the stack operations differ between M_1 and M_2 at some time. Then the measurement, whose result corresponds to the stack operation, causes the superposition to collapse to either M_1 or M_2. Thus M moves to the third component M_3 with probability 1/2 with v or w on the stack. Therefore, if $v = y^R$ or $w = y^R$, M accepts the input with probability 1/4.

We define M in detail in the following.

M has the following states:

- $q_{1,u,1,-}$, $q_{1,u,0,-}$, $q_{1,\#,0,-}$, $q_{1,v,1,-}$, $q_{1,v,0,a}$, $q_{1,v,0,b}$, $q_{1,w,1,-}$, $q_{1,x,1,-}$.
- $q_{2,u,1,-}$, $q_{2,v,1,-}$, $q_{2,w,1,-}$, $q_{2,w,0,a}$, $q_{2,w,0,b}$, $q_{2,x,1,-}$, $q_{2,x,0,-}$, $q_{2,\%,0,-}$.
- $q_{3,y,1,-}$, $q_{3,y,0,pop}$.
- q_0, q_{acc}, q_{rej1}, q_{rej2}, q_{rej3}, q_{rej4}, q_{rej5}, q_{rej6}, q_{rej7}, q_{rej8}, q_{rej9}, q_{rej10}, q_{rej11}.

q_{acc} is an accepting state. $q_{rej1}, \ldots, q_{rej11}$ are rejecting states. The index of $q_{i,z,d,c}$ denotes that the state is used to process substring z in M_i, $\Delta(q_{i,z,d,c}) = d$, and c represents the value of $\sigma(q_{i,z,d,c})$ as follows: the stack top symbol is popped ($c =$ 'pop'), the stack is unchanged ($c = -$), and c is pushed (otherwise). The initial state is q_0.

The state transition diagrams of components

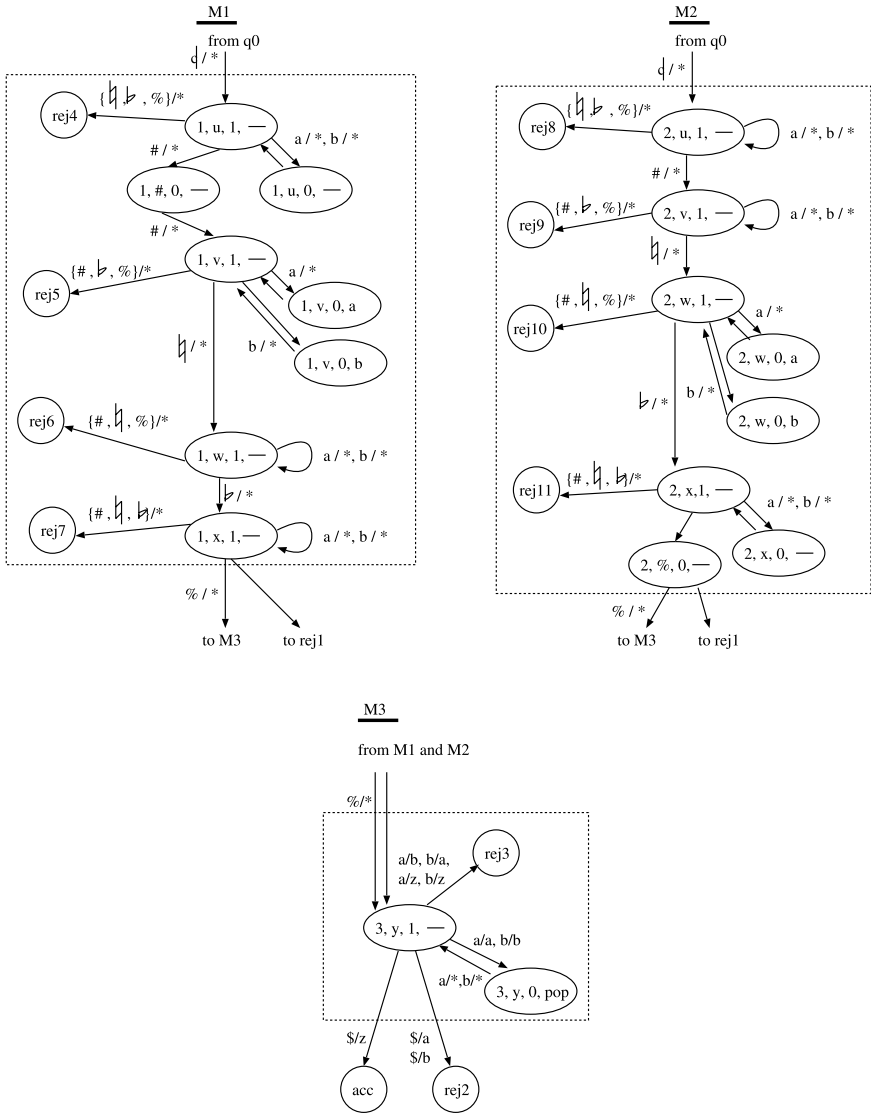

Fig. 2 M_1, M_2, and M_3. Each edge label represents a pair of an input symbol and a stack top symbol, where '*' is a wild card.

M_1, M_2, and M_3 are illustrated in **Fig. 2**. It is straightforward to see that the corresponding time evolution operators can be extended to be unitary by properly determining thus far undefined transitions.

□

It is also shown in Ref. 11) that L_1 cannot be recognized by nondeterministic pushdown automata. This implies that L_1 cannot be recognized by probabilistic pushdown automata with one-sided error, either. Moreover, it is shown that any probabilistic pushdown automaton can be simulated by QCPDAs with the same acceptance probability [11]. Thus one-sided error QCPDAs are strictly more powerful than one-sided error probabilistic pushdown automata.

5. Conclusion

In this paper, we described the power of several quantum automata in language recognition. It was shown that some models are more powerful than their classical counterparts, while others are not. It should be noted that criteria other than language recognition can be considered, such as the number of states and the error rate; actually, there have been research efforts along this line [1)~3)].

Restrictions such as reversible state transitions sometimes make quantum automata less powerful. However, by introducing classical resources in certain parts, it may be possible to relax such restrictions. Given the current scale of quantum computer development, quantum-classical hybrid models (i.e., automata that have a small amount of quantum resources and a large amount of classical resources) offer a promising solution.

References

1) Ambainis, A., Bonner, R., Freivalds, R. and Ķikusts, A.: Probabilities to accept languages by quantum finite automata, *Proc. COCOON'99*, pp.174–183 (1999).
2) Ambainis, A. and Freivalds, R.: 1-way quantum finite automata: strengths, weakness and generalizations, *Proc. 39th Symp. on Foundations of Computer Science*, pp.332–341 (1998).
3) Ambainis, A. and Ķikusts, A.: Exact results for accepting probabilities of quantum automata, *Theoretical Computer Science*, Vol.295, pp.3–25 (2003).
4) Ambainis, A. and Watrous, J.: Two-way finite automata with quantum and classical states, *Theoretical Computer Science*, Vol.287, No.1, pp.299–311 (2002).
5) Bonner, R., Freivalds, R. and Kravtsev, M.: Quantum versus probabilistic one-way finite automata with counter, *Proc. 28th Conference on Current Trends in Theory and Practice of Informatics* (*SOFSEM2001*), LNCS 2234, pp.181–190 (2001).
6) Golovkins, M.: Quantum pushdown automata, *Proc. 27th Conference on Current Trends in Theory and Practice of Informatics* (*SOFSEM2000*), LNCS 1963, pp.336–346 (2000).
7) Kondacs, A. and Watrous, J.: On the power of quantum finite state automata, *Proc. 38th Symp. on Foundations of Computer Science*, pp.66–75 (1997).
8) Kravtsev, M.: Quantum finite one-counter automata, *Proc. 26th Conference on Current Trends in Theory and Practice of Informatics* (*SOFSEM1999*), LNCS 1725, pp.432–442 (1999).
9) Moore, C. and Crutchfield, J.P.: Quantum automata and quantum grammars, *Theoretical Computer Science*, Vol.237, No.1-2, pp.275–306 (2000).
10) Murakami, Y., Nakanishi, M., Yamashita, S. and Watanabe, K.: Quantum versus classical pushdown automata in exact computation, *IPSJ Journal*, Vol.46, No.10, pp.2471–2480 (2005).
11) Nakanishi, M., Hamaguchi, K. and Kashiwabara, T.: Expressive power of one-sided error quantum pushdown automata with classical stack operations, *IEICE Trans. Inf. & Syst.* (in press).
12) Nakanishi, M., Indoh, T., Hamaguchi, K. and Kashiwabara, T.: On the power of non-deterministic quantum finite automata, *IEICE Trans. Inf. & Syst.*, Vol.E85-D, No.2, pp.327–332 (Feb. 2002).
13) Nayak, A.: Optimal lower bounds for quantum automata and random access codes, *Proc. 40th Annual Symposium on Foundations of Computer Science*, pp.369–376 (1999).
14) Qiu, D.: Simulations of quantum Turing machines by quantum multi-stack machines, e-print, quant-ph/0501176 (2005).
15) Yamasaki, T., Kobayashi, H., Tokunaga, Y. and Imai, H.: One-way probabilistic reversible and quantum one-counter automata, *Theoretical Computer Science*, Vol.289, No.2, pp.963–976 (2002).
16) Yamasaki, T., Kobayashi, H. and Imai, H.: Quantum versus deterministic counter automata, *Proc. Computing and Combinatorics Conference*, LNCS 2387, pp.584–594 (2002).

(Received February 8, 2005)
(Accepted July 4, 2005)
(Online version of this article can be found in the IPSJ Digital Courier, Vol.1, pp.442–449.)

Masaki Nakanishi was born in Osaka, Japan, in 1973. He received the B.E., M.E. and Ph.D. degrees from Osaka University, Japan, in 1996, 1998 and 2002 respectively. He is currently with the Graduate School of Information Science, Nara Institute of Science and Technology, as a Research Associate. His current interests include quantum computation and design of combinatorial algorithms.