On the Power of Quantum Pushdown Automata with a Classical Stack and 1.5-way Quantum Finite Automata

Masaki Nakanishi[†], Takao Indoh[‡], Kiyoharu Hamaguchi[‡],
and Toshinobu Kashiwabara[‡]
[†]Graduate School of Information Science, Nara Institute of Science and Technology,
Ikoma, Nara 630-0101, Japan
[‡]Graduate School of Engineering Science, Osaka University,
Toyonaka, Osaka 560-8531, Japan

Abstract

One of important questions on quantum computing is whether there is a computational gap between the models that is allowed to use quantum effects and the models that is not. Several types of quantum computational models have been proposed, including quantum finite automata, quantum pushdown automata and quantum branching programs, and it has been shown that some computational models are more powerful than classical counterparts and some are not since quantum computational models are required to obey some restrictions such as reversible state transitions. In this paper, we introduce quantum pushdown automata whose stack is implemented as a classical device. We show that our quantum pushdown automaton model can recognize some non-context-free languages with arbitrarily large acceptance probability and every deterministic context-free language with zero error. We also show that 1.5-way quantum finite automata can recognize the non-context-free language with probability less than $\frac{2}{3}$.

1 Introduction

One of important questions on quantum computing is whether there is a computational gap between the models that is allowed to use quantum effects and the models that is not. Several types of quantum computational models have been proposed, including quantum finite automata, quantum pushdown automata and quantum branching programs [1, 3, 2, 4]. In [2], it has been shown that quantum push down automata can recognize every regular language and some non-context-free languages. Their model has a quantum tape head and a quantum stack, and needs $O(n)$ qbits for realization, where n is an execution time.

In this paper, we introduce another model of quantum pushdown automata whose stack is implemented as a classical device. This means that our model needs $\lceil \log m \rceil$ qbits for specifying the position of the head, in addition to the constant number of qbits for representing the finite state control, where m is an input length. We show that our quantum automaton model can recognize some non-context-free languages with arbitrarily large acceptance probability and every deterministic context-free language with zero error. The quantum part of quantum pushdown automata with a classical stack can be regarded as a 1.5-way quantum finite automaton. We also show that 1.5-way quantum finite automata can recognize one of the non-context-free languages with probability less than $\frac{2}{3}$.

2 Preliminaries

In this section, we define quantum pushdown automata with a classical stack (QCPA). A QCPA has two state controls. One is a quantum state control for moving its tape head, and the other is a classical state control for dealing with the stack. A QCPA also has an input tape to which a quantum head is attached, and has a classical stack to which a classical stack top pointer is attached. The classical state control reads the stack top symbol and the result of an observation of the quantum part. The quantum state control reads the stack top symbol pointed by the classical stack top pointer, and the input symbol pointed by the quantum head. The outputs are determined based on a state of the classical state control. We define QCPAs formally as follows.

Definition 1 *Quantum Pushdown Automata with a Classical Stack*

A Quantum Pushdown Automaton with a Classical Stack (QCPA) is defined as the following 11-tuple:

$$M = (Q_q, Q_c, \Sigma, \Gamma, \delta, \sigma, qq_0, qc_0, \mathcal{O}, S_{acc}, S_{rej}),$$

where Q_q is the set of quantum states, Q_c is the set of classical states, Σ is the set of input symbols including the left and the right end-marker $\{\cent, \$\}$ respectively, Γ is the set of stack symbols including the bottom symbol Z, δ is the quantum state transition function ($\delta : (Q_q \times \Sigma \times \Gamma \times Q_q \times \{0,1\}) \longrightarrow \mathbb{C}$), σ is the classical state transition function ($\sigma : (Q_c \times \Gamma \times S) \longrightarrow (Q_c \times \Gamma^*)$, where S is the set of possible results of the observations), qq_0 is the initial quantum state, qc_0 is the initial classical state, $\mathcal{O} = \oplus_i E_i$ is the observable, $S_{acc}(\subseteq Q_c)$ is the set of accepting states, and $S_{rej}(\subseteq Q_c)$ is the set of rejecting states, where $S_{acc} \cap S_{rej} = \emptyset$.

$\square$

$\delta(q, a, b, q', D) = \alpha$ means that the amplitude of the transition from q to q' moving its head to D (1 means 'right' and 0 means 'stay') and reading an input symbol a and a stack symbol b is α. $\sigma(q, a, r) = (q', w)$ means that the classical state changes from q to q' and w is pushed when the stack top symbol is a and the result of the observation is r. A configuration of the quantum part is a pair (q, k), where $q \in Q_q$, k is the position of the quantum head. It is obvious that the number of configurations of the quantum part is $n|Q_q|$, where n is an input length.

A superposition of configurations of QCPA M is any element of $l_2(Q_q \times \mathbb{Z}_n)$ of unit length, where $\mathbb{Z} = \{0, 1, \ldots, n-1\}$. For each configuration, we define a column vector $|q, k\rangle$ as follows:

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

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

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

where $x(k)$ is the k-th input symbol in the input $\boldsymbol{x}$. If $U_a^{\boldsymbol{x}}$ is unitary, that is, $U_a^{\boldsymbol{x}} U_a^{\boldsymbol{x}\dagger} = U_a^{\boldsymbol{x}\dagger} U_a^{\boldsymbol{x}} = I$, where $U_a^{\boldsymbol{x}\dagger}$ is a transpose conjugate of $U_a^{\boldsymbol{x}}$, then we say that the corresponding QCPA is well-formed. This means that the QCPA is considered to be valid in terms of the quantum theory. We consider only well-formed QCPAs in the following.

We define the notion of "words accepted by a QCPA" as follows.

Let the initial quantum state be qq_0. Let the initial position of the quantum head is '0'. We define $|\psi_0\rangle = |qq_0, 0\rangle$. Initially, there is a stack bottom symbol Z in the stack. The computation of the QCPA proceeds as follows:

<u>For an input word $\boldsymbol{x}$, the quantum part works as follows:</u>

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

(b) We observe $|\psi_{i+1}\rangle$ with respect to the observable $\mathcal{O}$. Note that this observation causes $|\psi_{i+1}\rangle$ to be projected onto the subspace compatible with the observation. Then we go to (c).

<u>The classical part works as follows:</u>

(c) Let the current classical state, the current stack top symbol, and the result of the observation be q, a, and r respectively. Suppose that $\sigma(q,a,r) = (q', w)$. Then, the classical state changes from q to q', the stack top symbol is popped, and the word w is pushed. If the new classical state is in S_{acc} (in S_{rej}, resp.), it outputs '1' ('0', resp.). Otherwise, we repeat (a).

We call the above (a), (b) and (c) 'one step' collectively. For a word $\boldsymbol{x}$, if the probability of outputting 1 is greater than $\frac{1}{2}+\varepsilon$ $(0<\varepsilon\leq\frac{1}{2})$, we say that the QCPA accepts the word w. If the probability of outputting 0 is less than $\frac{1}{2}-\varepsilon$ $(0<\varepsilon\leq\frac{1}{2})$, we say that it rejects the word w.

If a QCPA accepts any word in a language L and rejects any word not in L, we say that the language L is recognized by the QCPA. To check well-formedness, we introduce the following theorem.

Theorem 1 *QCPA M is well-formed. $\Leftrightarrow$ The quantum state transition function of M satisfies the following conditions for any $a \in \Gamma$:*

(1)

$$\forall q_1, q_2, b \quad \sum_{q',D} \overline{\delta(q_1,b,a,q',D)}\delta(q_2,b,a,q',D) = \begin{cases} 1 & (q_1 = q_2) \\ 0 & (q_1 \neq q_2) \end{cases},$$

(2)

$$\forall q_1, q_2, b_1, b_2 \quad \sum_{q'} \overline{\delta(q_1,b_1,a,q',0)}\delta(q_2,b_2,a,q',1) = 0,$$

where $\overline{\delta(\cdot,\cdot,\cdot)}$ is the conjugate of $\delta(\cdot,\cdot,\cdot)$.

(Proof)

A matrix $U_a^{\boldsymbol{x}}$ is unitary if and only if the vectors $U_a^{\boldsymbol{x}}|q,k\rangle$ are orthonormal. Equivalently,

(a) $\forall q, k \quad ||U_a^{\boldsymbol{x}}|q,k\rangle|| = 1$

(b) $\forall q_1, q_2, k_1, k_2 \ (q_1,k_1) \neq (q_2,k_2) \quad U_a^{\boldsymbol{x}}|q_1,k_1\rangle \perp U_a^{\boldsymbol{x}}|q_2,k_2\rangle$

The condition (a) is equivalent to the following condition.

$$\forall q, k \quad \sum_{q',D} \overline{\delta(q,x(k),a,q',D)}\delta(q,x(k),a,q',D) = 1 \qquad (\mathrm{a}')$$

This condition (a$'$) is equivalent to the case $q_1 = q_2$ of the condition (1). The condition (b) is equivalent to the following condition.

$\forall q_1, q_2, k_1, k_2 \ (q_1,k_1) \neq (q_2,k_2)$

$$\sum_{q',D} \delta(q_1,x(k_1),a,q',D)|q',k_1+D\rangle \perp \sum_{q',D} \delta(q_2,x(k_2),a,q',D)|q',k_2+D\rangle \quad (\mathrm{b}')$$

We consider the following two cases.

Case 1 $k_1 = k_2$

In this case, $x(k_1) = x(k_2)$. Thus, the above condition is equivalent to the following condition.

$$\forall q_1, q_2 \ (q_1 \neq q_2) \quad \sum_{q',D} \overline{\delta(q_1,b,a,q',D)}\delta(q_2,b,a,q',D) = 0$$

This condition is equivalent to the case $q_1 \neq q_2$ of the condition (1).

Case 2 $k_1 \neq k_2$

We assume that $k_1 > k_2$ without loss of generality. If $k_1 \geq k_2 + 2$, it is obvious that the condition (b′) holds. We assume that $k_1 = k_2 + 1$. Then, the condition (b′) is equivalent to the condition (2).

□

For simplicity, we handle a subclass of QCPAs such 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 $V_{a,b} : l_2(Q_q) \longrightarrow l_2(Q_q)$ for changing states, and a function $D : Q_q \longrightarrow \{0, 1\}$ for moving the head. We assume that direction of movement of the head is determined based on the state to which the current state makes a transition. Then we can define δ 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 $V_{a,b} |q\rangle$. Then, it is straightforward to prove the following theorem from Theorem 1.

Theorem 2 *QCPA M is well-formed.* $\Leftrightarrow$ *For any $a \in \Sigma$ and $b \in \Gamma$, the linear operator $V_{a,b}$ satisfies the following condition:*

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

□

3 On the power of QCPAs

In this section, we show that QCPAs can recognize some non-context-free languages L_1 and L_2, and also show that QCPAs can recognize every deterministic context-free language. We define the non-context-free languages L_1 and L_2 as follows.

$$L_1 = \{a^n db^n dc^n | n \geq 0\}, \quad L_2 = \{a^i db^j dc^k | i \geq j = k\}$$

It is straight forward to prove that L_1 and L_2 are not context-free languages by the pumping lemma. Firstly, we show the QCPA that recognizes the language L_2 in the following theorem.

Theorem 3 *QCPAs can recognize the language L_2.*

(Proof)

We construct a QCPA that recognizes the language L_2 as follows.

$$\begin{aligned} Q_c &= \{q_{c0}, q_{c1}, q_{c2}, q_{c3}, q_{c4}\}, \\ S_{\text{acc}} &= \{q_{c3}\}, \quad S_{\text{rej}} = \{q_{c4}\}, \\ Q_q &= \{q_0, q_1\} \\ &\quad \cup \{r_{j,k} | 1 \leq j \leq N, 0 \leq k \leq j\} \\ &\quad \cup \{r'_{j,k} | 1 \leq j \leq N, 0 \leq k \leq N - j + 1\} \\ &\quad \cup \{s_{j,k} | 1 \leq j \leq N, 0 \leq k \leq j\} \\ &\quad \cup \{s'_{j,k} | 1 \leq j \leq N, 0 \leq k \leq N - j + 1\} \\ &\quad \cup \{s''_j | 1 \leq j \leq N\}, \\ \Sigma &= \{a, b, c, d\}, \quad \Gamma = \{A, B, Z\}, \end{aligned}$$

where N is a positive integer ($N \geq 3$) that does not depend on an input length. We define the observable $\mathcal{O}$ as follows:

$$\begin{aligned} Q_{\text{acc}} &= \{s''_N\} \\ Q_{\text{rej}} &= \{q_1\} \\ &\cup\{s_{j,k} | 1 \leq j \leq N, 0 \leq k \leq j\} \\ &\cup\{s'_{j,k} | 1 \leq j \leq N, 0 \leq k \leq N-j+1\} \\ &\cup\{s''_j | 1 \leq j < N\}, \end{aligned}$$

$\mathcal{O} = E_{\text{non1}} \oplus E_{\text{non2}} \oplus E_{\text{acc}} \oplus E_{\text{rej}}$, where

$$\begin{aligned} E_{\text{non1}} &= \mathit{span}\{|q_0, k\rangle\}, \\ E_{\text{non2}} &= \mathit{span}\{|q, k\rangle \, | q \in Q_q \setminus (Q_{\text{acc}} \cup Q_{\text{rej}} \cup \{q_0\}), \\ E_{\text{acc}} &= \mathit{span}\{|q, k\rangle \, | q \in Q_{\text{acc}}\}, \\ E_{\text{rej}} &= \mathit{span}\{|q, k\rangle \, | q \in Q_{\text{rej}}\}. \end{aligned}$$

We define the outcomes of an observation to be "non1", "non2", "acc" and "rej" corresponding to E_{non1}, E_{non2}, E_{acc} and E_{rej} respectively.

We define the head movement function D as follows:

$$\begin{aligned} R_1 &= \{q_0\} \cup \{r_{j,0} | 1 \leq j \leq N\} \cup \{r'_{j,0} | 1 \leq j \leq N\}, \\ R_0 &= Q_q \setminus R_1, \\ \Delta(q) &= \begin{cases} 1 & (q \in R_1) \\ 0 & (q \in R_0). \end{cases} \end{aligned}$$

We define the classical state transition function σ as follows, where $x \in \Gamma$ is any stack symbol, and ε is an empty word.

$$\sigma(q_{c0}, x, \text{“non1”}) = (q_{c1}, Z),$$

$$\sigma(q_{c1}, x, \text{“non1”}) = (q_{c1}, A^{N+3}), \;\; \sigma(q_{c1}, x, \text{“non2”}) = (q_{c2}, A^2),$$

$$\sigma(q_{c2}, A, \text{“non2”}) = (q_{c2}, \varepsilon), \;\; \sigma(q_{c2}, B, \text{“non2”}) = (q_{c2}, B),$$

$$\sigma(q_{c2}, Z, \text{“non2”}) = (q_{c2}, B),$$

$$\sigma(q, x, \text{“acc”}) = (q_{c3}, A), \;\; \sigma(q, x, \text{“rej”}) = (q_{c4}, A) \;[q \in Q_q].$$

We define linear operators as follows:

$$V_{¢,x} |q\rangle = |q\rangle \quad [q \in Q_q], \;\; V_{a,x} |q_0\rangle = |q_0\rangle$$

$$V_{a,x} |r_{j,0}\rangle = |s_{j,0}\rangle \,, \;\; V_{a,x} \left|r'_{j,0}\right\rangle = \left|s'_{j,0}\right\rangle$$

$$V_{b,x} |q_0\rangle = |q_1\rangle \,, \;\; V_{b,x} |r_{j,k}\rangle = |r_{j,k+1 \ mod \ j+1}\rangle$$

$$V_{b,x} \left|r'_{j,0}\right\rangle = \left|s'_{j,0}\right\rangle$$

$$V_{c,x} |q_0\rangle = |q_1\rangle \,, \;\; V_{c,x} |r_{j,0}\rangle = |s_{j,0}\rangle$$

$$V_{c,x} \left|r'_{j,k}\right\rangle = \left|r'_{j,k+1 \ mod \ N-j+1+1}\right\rangle$$

$$V_{d,x} |q_0\rangle = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} |r_{j,0}\rangle \,, \;\; V_{d,x} |r_{j,0}\rangle = \left|r'_{j,0}\right\rangle$$

$$V_{d,x} \left|r'_{j,0}\right\rangle = \left|s'_{j,0}\right\rangle$$

$$V_{\$,x} |q_0\rangle = |q_1\rangle \,, \;\; V_{\$,x} |r_{j,0}\rangle = |s_{j,0}\rangle$$

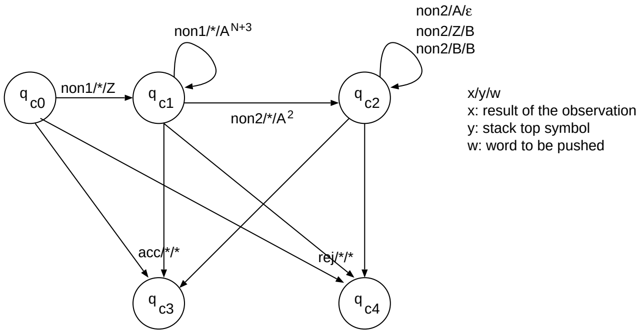

Figure 1: An example of the classical part of the QCPA that recognizes L_2.

$$V_{\$,A} \left| r'_{j,0} \right\rangle = \frac{1}{\sqrt{N}} \sum_{l=1}^{N} exp(\frac{2\pi i}{N} jl) \left| s''_l \right\rangle$$

$$V_{\$,Z} \left| r'_{j,0} \right\rangle = \frac{1}{\sqrt{N}} \sum_{l=1}^{N} exp(\frac{2\pi i}{N} jl) \left| s''_l \right\rangle$$

$$V_{\$,B} \left| r'_{j,0} \right\rangle = \left| s'_{j,0} \right\rangle$$

Then, the QCPA $M = (Q_q, Q_c, \Sigma, \Gamma, \delta, \sigma, q_0, q_{c0}, S_{\text{acc}}, S_{\text{rej}})$ recognizes the language L_2. An example of the QCPA that recognizes L_2 is illustrated in Figure 1 and Figure 2 for $N = 3$. For each state, if it has multiple incoming transitions, then the transitions are so-called quantum Fourier transfrom, otherwise, it has at most one incoming transition. Thus, it is easy to see that these linear operators can be extended to be unitary. It is obvious that this QCPA rejects the words that are not strings of the form $a^l db^m dc^n$. The QCPA acts as follows for an input $a^l db^m dc^n$.

First, we explain the behavior of the classical part. The QCPA pushes A^{N+2} (pops one A, and then, pushes A^{N+3}) until it reads the first d. After reading the first d, the number of A's in the stack is $l(N+2)+1$. Then, A is popped until it reads the right end-marker \$. If the stack bottom symbol is read before \$ is read, then B is pushed.

Next, we explain the behavior of the quantum part. When it reads the first d, the computation branches into N paths. For j-th path, the quantum head moves to right every $j+1$ steps while reading b, and moves to right every $N-j+1$ steps while reading c. Thus, the number of steps needed from reading the first d to reading \$ is $m(j+1) + n(N-j+1) + 1$ for j-th path.

Case 1 $m \neq n$

Note that each computation path reads the right-end-marker at a different time. Thus, there is no cancellation between the states $\{s''_1, s''_2, \ldots, s''_N\}$. Therefore, the QCPA rejects the input with probability at least $\frac{N-1}{N}$.

Case 2 $m = n$

Note that each computation path reads the right-end-marker at the same time, and that the number of steps needed from reading the first d to reading \$ is $m(N+2)+1$. Thus, if the stack top symbol is A or Z (i.e., $l \geq m$), the state changes to s''_N with certainty since $V_{\$,\{A,Z\}} \sum_j \frac{1}{\sqrt{N}} \left| r'_{j,0} \right\rangle = \left| s''_N \right\rangle$,[1] and the QCPA accepts the input. If the stack top symbol is B (i.e., $l < m$), the states changes to $s'_{j,0}$ with certainty, and the QCPA rejects the input.

Therefore, this QCPA recognizes the language L_2 with error rate $\frac{1}{N}$.

□

[1] This equation results from the property of quantum Fourier transform.

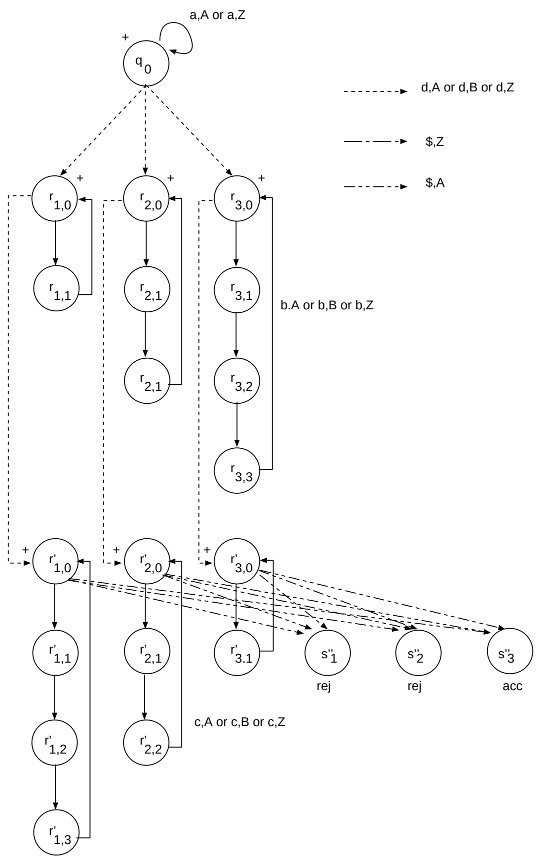

Figure 2: An example of the quantum part of the QCPA that recognizes L_2. The head moves to right when making a transition to the state to which '+' is attached. Some of the rejecting states are omitted.

It is obvious that QCPAs can recognize the language L_1 if we redefine the linear operator $V_{\$,A}$ as follows:

$$V_{\$,A}\left|r'_{j,0}\right\rangle = \left|s'_{j,0}\right\rangle .$$

Next, we show that QCPAs can recognize every deterministic context-free language. That is, QCPAs can recognize every language that is recognized by deterministic pushdown automata.

Theorem 4 *QCPAs can recognize every deterministic context-free language.*

(Sketch of proof)

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, Q_{\text{acc}})$ be a deterministic pushdown automaton, where Q is the set of states, Σ is the set of input symbols, Γ is the set of stack symbols, δ is the state transition function ($\delta : Q\times\Sigma\times\Gamma \longrightarrow Q\times\Gamma^*\times\{0,1\}$), q_0 is the initial state, and Q_{acc} is the set of accepting state. Note that, since M is a deterministic pushdown automaton, $\delta(\cdot,\cdot,\cdot) = (\cdot,\cdot,1)$, that is, the tape head moves to right on each step. We make M a reversible pushdown automaton by adding extra states and stack symbols, where *reversible* means that each state has at most one incoming transition for each stack symbol. Suppose that a state q has multiple incoming transition for a stack symbol a. For each such incoming transition $t : \delta(q', a, x) = (q, w, 1)$, we define a new state q_t. We also add a new stack symbol s_t to Γ and define δ as $\delta(q', a, x) = (q_t, s_t, 0)$ and $\delta(q_t, s_t, x) = (q, w, 1)$. We illustrate how to add new states in Figure 3. The resulting pushdown automaton is reversible. Thus, a QCPA can simulate the resulting pushdown automaton.

□

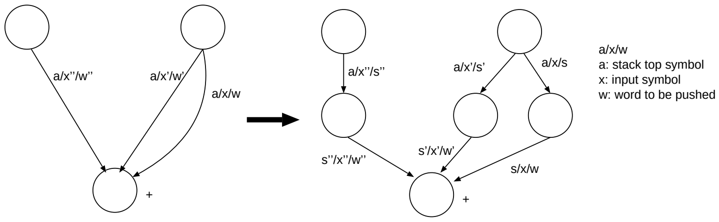

Figure 3: How to add new states.

4 1.5-way Quantum Finite Automata

In this section, we show that 1.5-way quantum finite automata can recognize the non-regular language L_3 and the non-context-free language L_1.

Definition 2 *1.5-way Quantum Finite Automata*
A 1.5-way Quantum Finite Automaton (1.5-QFA) is defined as the following 6-tuple:

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

where Q is the set of quantum states, Σ is the set of input symbols including the left and the right end-marker $\{\cent, \$\}$ respectively, δ is the quantum state transition function ($\delta : (Q \times \Sigma \times Q \times \{0,1\}) \longrightarrow \mathbb{C}$), q_0 is the initial quantum state, Q_{acc} is the set of accepting states, and Q_{rej} is the set of rejecting states, where $Q_{acc} \cap Q_{rej} = \emptyset$.

□

Informally, a 1.5-QFA is considered as a computational model that is a QCPA without a stack, that is, a 1.5-QFA is a quantum part of a QCPA. The observable $\mathcal{O}$ is defined to be $\mathcal{O} = E_{\mathrm{acc}} \oplus E_{\mathrm{rej}} \oplus E_{\mathrm{non}}$, where $E_{\mathrm{acc}} = span\{|q,k\rangle\,|q \in Q_{\mathrm{acc}}\}$, $E_{\mathrm{rej}} = span\{|q,k\rangle\,|q \in Q_{\mathrm{rej}}\}$, and $E_{\mathrm{non}} = span\{|q,k\rangle\,|q \in Q_{\mathrm{non}}\}$. We define the outcomes of an observation to be "non", "acc" and "rej" corresponding to E_{non}, E_{acc} and E_{rej} respectively. Then, the notion of "words accepted by a 1.5-QFA" is the same as that for QCPAs except that (c), the action of the classical part, is replaced with the following:

(c) If the outcome of the observation is "acc" ("rej", resp.), then the 1.5-QFA outputs 1 (0, resp.). Otherwise, we repeat (a).

We define the non-regular language L_3 as follows:

$$L_3 = \{a^m db^n cb^n | n \geq 0,\ m \geq 0\}$$

Theorem 5 *1.5-QFA can recognize the language L_3.*

(Proof)
It is obvious that QCPAs can recognize the language L_3 if we redefine the linear operator $V_{\$,B}$ in Theorem 3 as follows:

$$V_{\$,B}\,|r'_{j,0}\rangle = \frac{1}{\sqrt{N}} \sum_{l=1}^{N} exp(\frac{2\pi i}{N} jl)\,|s''_l\rangle$$

The computation of this QCPA does not depend on the stack top symbol. Thus, 1.5-QFAs can simulate the QCPA.

□

It is easy to see that 1.5-QFAs can recognize $L'_3 = \{a^n db^n dc^m\}$ and $L''_3 = \{a^n db^m dc^n\}$.

Theorem 6 *1.5-QFA can recognize the language* L_1*.*

(Sketch of proof)

Note that, by Theorem 5, 1.5-QFAs can recognize L_3, L_3' and L_3'' with arbitrarily large probability. As in Theorem 3.2 in [2], the computation of the 1.5-QFA branches into three paths equally likely. For the input $\{a^i db^j dc^k\}$, it checks $i = j$ on the first path, and it checks $j = k$ on the second path, it always rejects the input on the third path. Thus, if $i = j = k$, the the first and the second paths accept the input, otherwise, at least two of the three paths reject the input. Therefore, L_1 can be recognized by 1.5-QFAs with probability less than $\frac{2}{3}$.

□

Note that the acceptance probability of the 1.5-QFA in Theorem 6 is less than $\frac{2}{3}$. On the other hand, using the QCPA in Section 3, the acceptance probability can be arbitrarily large.

5 Conclusion

We have shown that quantum pushdown automata with a classical stack can recognize some non-context-free languages and every deterministic context-free language.

It is still the future work to study on the quantum pushdown automata whose tape head is implemented as a classical part as well as the stack.

References

[1] A.Ambainis and J.Watrous, “Two-way finite automata with quantum and classical states,” LANL e-print cs.CC/9911009, 1999.

[2] M.Golovkins, “Quantum Pushdown Automata,” LANL e-print quant-ph/0102054, 2001.

[3] A.Kondacs and J.Watorus, “On the power of quantum finite state automata,” Proc. 38th Symp. on Foundations of Computer Science, pp.66–75, 1997.

[4] M.Nakanishi, K.Hamaguchi, and T.Kashiwabara, “Ordered quantum branching programs are more powerful than ordered probabilistic branching programs under a bounded-width restriction,” Proc. COCOON 2000, LNCS 1858, pp. 467–476, 2000.