On generalized quantum Turing machine and its language classes

SATOSHI IRIYAMA
Tokyo University of Science
Department of Information Sciences
Yamasaki 2641, Noda City, Chiba
JAPAN

MASANORI OHYA
Tokyo University of Science
Department of Information Sciences
Yamasaki 2641, Noda City, Chiba
JAPAN

Abstract: Ohya and Volovich have proposed a new quantum computation model with chaotic amplification to solve the SAT problem, which went beyond usual quantum algorithm. In this paper, we generalize quantum Turing machine by rewriting usual quantum Turing machine in terms of channel transformation. Moreover, we define some computational classes of generalized quantum Turing machine and show that we can treat the Ohya-Volovich (OV) SAT algorithm.

Key–Words: Quantum Information, Quantum Algorithm, Quantum Turing Machine, Quantum Channel

1 Introduction

The problem whether NP-complete problems can be P problem has been considered as one of the most important problems in theory of computational complexity. Various studies have been done for many years. Ohya and Volovich [1, 2, 11] proposed a new quantum algorithm with chaotic amplification to solve the SAT problem, which went beyond usual quantum algorithm. This quantum chaos algorithm enabled to solve the SAT problem in a polynomial time [1, 2, 3, 11].

In this paper we generalize quantum Turing machine so that it enables to describe non-unitary evolution of states. This study is based on mathematical studies of quantum communication channels [4, 5]. It is discussed in this generalized quantum Turing machine (GQTM) that we can treat the OV SAT algorithm.

In Section 1, we generalize QTM by rewriting usual QTM in terms of channel transformation so that it contains both dissipative and unitary dynamics. In Section 3, the SAT problem is reviewed and fundamental quantum unitary gates are presented. In Section 4, based on the papers [3, 7, 12], we concretely construct the fundamental gates needed for computation of the SAT problem. In Section 5, we rewrite the total process including a measurement process and amplifier process with chaotic dynamics by GQTM.

2 Generalized Quantum Turing Machine

In this section, we first explain the notations used in this paper, and mention Classical Turing Machine(TM or CTM), deterministic TM and multi track TM brielfy.

2.1 Classical Turing Machine

CTM M_{cl} is defined by a triplet (Q, Σ, δ), where Σ is a finite alphabets with an identified blank symbol $\#$, Q is a finite set of processor states (with an initial state q_0 and a set of final states $\{q_F\}$). CTM has a processor, a sequence of alphabets $\Sigma^* = \Sigma \times \cdots \times \Sigma$ called a tape and a tape head. A current state of CTM called a configuration is represented as

$$\rho = (q, A, i) \in Q \times \Sigma^* \times \mathbb{Z}$$

where $A : \mathbb{Z} \to \Sigma$ is a tape state.

Let $\delta : Q \times \Sigma \to 2^{Q\times\Sigma\times\{0,\pm1\}}$ be a transition function. Note that $\{0, \pm1\}$ indicates moving direction of the tape head of TM. The deterministic TM has a deterministic transition function $\delta : Q \times \Sigma \to Q \times \Sigma \times \{0, \pm1\}$, that is, δ is a non-branching map, in other words, the range of δ for each $(q, a) \in Q \times \Sigma$ is unique. A TM M is called non-deterministic if it is not deterministic. A n-multi track CTM is defined as $M_{cl} = (Q, \Sigma^n, \delta)$ likely to above CTM. This has n dimensional alphabet set as a tape symbol set.

We also define the d multi-track TM, which tape symbol is $\Sigma = \Sigma^d$. A multi-track TM has some workspaces for calculation, whose tracks are independent each other. This independence means that the TM can operate only one track at one step and all tracks do not affect each other.

2.2 Generalized Quantum Turing Machine

Quantum Turing machine (QTM) was introduced by Deutsch [8] and was studied by many reserchers. Bernstein and Vazirani showed some theory in CTM can be expanded to QTM [9] and they proved there exists universal QTM which computs the functions given by its codes as input data.

Here, we introduce a Generalized Quantum Turing machine (GQTM) which contains QTM as a special case. We define GQTM by a completely positive quantum channel (see below) instead of a unitary operator.

GQTM M_{gq} is defined by quadruplet $(Q, \Sigma, \mathcal{H}, \Lambda_\delta)$, where Λ_δ is a quantum transition function from a configuration to a configuration. Q and Σ are represented by a density operator on Hilbert space $\mathcal{H}_Q$ and $\mathcal{H}_\Sigma$,which are spanned by canonical basis $\{|q\rangle ; q \in Q\}$ and $\{|a\rangle ; a \in \Sigma\}$, respectively. A tape configuration A is a sequence of elements of Σ represented by a density operator on Hilbert space $\mathcal{H}_\Sigma$ spanned by a canonical basis $\{|A\rangle ; A \in \Sigma^*\}$, where Σ^* is the set of all sequences of alphabets in Σ. A position of tape head is represented by a density operator on Hilbert space $\mathcal{H}_Z$ spanned by a canonical basis $\{|i\rangle ; i \in \mathbb{Z}\}$. Then a configuration ρ of GQTM M_{gq} is described by a density operator in $\mathcal{H} \equiv \mathcal{H}_Q \otimes \mathcal{H}_\Sigma \otimes \mathcal{H}_Z$. Let $\mathfrak{S}(\mathcal{H})$ be the set of all density operators in Hilbert space $\mathcal{H}$.

Here, we define the transition function

$$\begin{aligned}\delta : \mathbb{R} \times Q \times \Sigma \times Q \times \Sigma \times Q \times \Sigma \times \{0, \pm 1\} \\ \times Q \times \Sigma \times \{0, \pm 1\} \to \mathbb{C}.\end{aligned}$$

A quantum transition function is given by a completely positive (CP) channel

$$\Lambda_\delta : \mathfrak{S}(\mathcal{H}) \to \mathfrak{S}(\mathcal{H}),$$

satysfying the following condition.

Definition 1 *Λ_δ is called a quantum transition channel if there exists a transition funtion δ such that for all quantum configuration $\rho = \sum_k \lambda_k |\psi_k\rangle \langle\psi_k|$, $|\psi_k\rangle = \sum_l \alpha_{k,l} |q_{k,l}, A_{k,l}, i_{k,l}\rangle$, $\sum_k \lambda_k = 1, \forall\lambda_k \geq 0, \sum_l |\alpha_{k,l}|^2 = 1, \forall\alpha_{k,l} \in \mathbb{C}$, it holds*

$$\begin{aligned}&\Lambda_\delta(\rho) \\ &= \sum_{k,l,p,b,d,p',b',d'} \delta\left(\lambda_k, q_{k,l}, A_{k,l}(i_{k,l}), p, b, d, p', b', d'\right) \\ &\times |p, B, i_{k,l} + d\rangle \langle p', B', i_{k,l} + d'|\end{aligned}$$

$$B(j) = \begin{cases} b & j = i_{k,l} \\ A_{k,l}(j) & \textit{otherwise} \end{cases}$$

$$B'(j) = \begin{cases} b' & j = i_{k,l} \\ A_{k,l}(j) & \textit{otherwise} \end{cases}$$

where the RHS of the first equation is a state.

Definition 2 *$M_{gq} = (Q, \Sigma, \mathcal{H}, \Lambda_\delta)$ is called a LQTM(Linear Quantum Turing Machine) if there exists a transition funtion*

$$\begin{aligned}\delta : Q \times \Sigma \times Q \times \Sigma \times Q \times \Sigma \times \{0, \pm 1\} \\ \times Q \times \Sigma \times \{0, \pm 1\} \to \mathbb{C}\end{aligned}$$

such that for all quantum configuration ρ_k, Λ_δ is written as

$$\begin{aligned}&\Lambda_\delta(\rho_k) \\ &= \sum_{l,p,b,d,p',b',d'} \delta\left(q_{k,l}, A_{k,l}(i_{k,l}), p, b, d, p', b', d'\right) \\ &\times |p, B, i_{k,l} + d\rangle \langle p', B', i_{k,l} + d'|\end{aligned}$$

where the RHS is a state, and for all quantum configuration $\sum_k \lambda_k \rho_k$, Λ_δ is affine;

$$\Lambda_\delta\left(\sum_k \lambda_k \rho_k\right) = \sum_k \lambda_k \Lambda_\delta(\rho_k)$$

Definition 3 *A GQTM M_{gq} is called unitary QTM (UQTM), if the quantum transition channel Λ_δ is unitary channel implemented: $\Lambda_\delta = Ad_{U_\delta}$. U_δ is given by, for $|\psi\rangle = |q, A, i\rangle$,*

$$\begin{aligned}U_\delta |\psi\rangle &= U_\delta |q, A, i\rangle \\ &= \sum_{p,b,r} \delta(q, A(i), p, b, d) |p, B, i + d\rangle\end{aligned}$$

where

$$\delta : Q \times \Sigma \times Q \times \Sigma \times \{0, 1\} \to \mathbb{C}$$

is a transition function and it satisfies for any $q \in Q, a \in \Sigma, q' (\neq q) \in Q, a' (\neq a) \in \Sigma$,

$$\sum_{p,b,d} |\delta(q, a, p, b, d)|^2 = 1.$$

$$\sum_{p,b,d} \delta\left(q', a', p, b, d\right)^* \delta(q, a, p, b, d) = 0.$$

Several studies have been done on UQTM whose transition channel is represented by unitary channel, in which various theoly and computational classes in UQTM were discussed in [9, 10].

Remark 4 *For any $q, p \in Q, a, b \in \Sigma, d \in \{0, \pm 1\}$, define $\delta(q, a, p, b, d) = 0$ or 1 then UQTM is a reversal CTM.*

A chaos amplifier used in [1, 2, 11] is a non-linear quantum channel, the details of this channel and its application to the SAT problem will be discussed in the sequel.

2.3 Computation process of GQTM

Let $M = (Q, \Sigma, \mathcal{H}, \Lambda_\delta)$ and $\rho_0 = |\psi_0\rangle \langle\psi_0|$ where $|\psi_0\rangle = |q_0, A, 0\rangle$, we call this state is an initial state and A is an input of M. Computation of GQTM proceeds applying Λ_δ to ρ_0 till the processor state becomes $q_f \in \{q_F\}$, then it halts. This process is described bt the products of Λ_δ as

$$\Lambda_\delta \circ \cdots \circ \Lambda_\delta(\rho_0) = \rho_f$$

ρ_f is called a final state in the form

$$\rho_f = \sum_k \lambda_k \rho_k + \sum_l \mu_l \sigma_l$$

$$\sum_k \lambda_k + \sum_l \mu_l = 1, \quad \forall \lambda_k, \mu_l \geq 0$$

where σ_l is a state includes the final (proccessor) states q_f. We call $p = \sum_l \mu_l$ a holting probability.

2.3.1 Computational class for GQTM

In this section, we explain the language classes defined by GQTM. Let L be the set of alphabet sequences, we call this a *language* if there exists TM (or GQTM) M that holts with any input $x \in L$ and doesn't holt $x \notin L$, and we say that M recognizes L.

Let us review some language classes defined by CTM.

Definition 5 *We say that the language is in class P if its language is recognized by a deterministic Turing machine in polynomial time of input size.*

Definition 6 *We say that the language is in class NP if there is a non-deterministic Turing machine in polynomial time of input size. Besides, if a language L_1 $\in$NP and L_1 reduces to L_2 $\in$NP in polynomial time, a language L_1 is NP-complete.*

Definition 7 *If languages are accepted by non-deterministic Turing machine in polynomial time of input size with a certain probability p, then this class of languages is called a bounded probability polynomial time(BPP).*

If there is a polynomial time algorithm to solve it in the above sense, it implies P=NP. The existence of such a algorithm is demonstrated in [1, 2, 11] in an extended quantum domain, as is reviewed in the next section. We will show that this OV algorithm can be written by GQTM in the sequel section.

Then, we define the *recognition* of GQTM and some classes of languages.

Definition 8 *Given GQTM M_{gq} and a language L, if there exists N steps when M_{gq} recognizes L by the probability p, we say that the GQTM M_{gq} recognizes L by the probability p and its computational complexity is N.*

Definition 9 *A language L is bounded quantum probability polynomial time GQTM(BGQPP) if there is a polynomial time GQTM M_{gq} which accepts L with probability $p \geq \frac{1}{2}$.*

Similarly, we can define the class of languages BUQPP (=BQPP) and BLQPP corresponding to UQTM and LQTM, respectively.

In Section 2, it is pointed out that LQTM includes classical TM, which implies

$$BPP \subseteq BLQPPL \subseteq BGQPP.$$

Moreover, if NLQTM accepts the SAT OV algorithm in polynomial time with probability $p \geq \frac{1}{2}$, then we may have the inclusion

$$NP \subseteq BGQPP.$$

3 SAT Problem

Let $X \equiv \{x_1, \ldots, x_n\}, n \in \mathbf{N}$ be a set. x_k and its negation $\overline{x}_k\ (k = 1, \ldots, n)$ are called literals Let $\overline{X} \equiv \{\overline{x_1}, \ldots, \overline{x_n}\}$ be a set, then the set of all literals is denoted by $X' \equiv X \cup \overline{X} = \{x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}\}$. The set of all subsets of X' is denoted by $\mathcal{F}(X')$ and an element $C \in \mathcal{F}(X')$ is called a clause. We take a truth assignment to all variables x_k. If we can assign the truth value to at least one element of C, then C is called satisfiable. When C is satisfiable, the truth value $t(C)$ of C is regarded as true, otherwise, that of C is false. Take the truth values as "true ↔1, false ↔0". Then $C$is satisfiable iff $t(C) = 1$.

Let $L = \{0, 1\}$ be a Boolean lattice with usual join $\vee$ and meet $\wedge$, and $t(x)$ be the truth value of a literal x in X. Then the truth value of a clause C is written as $t(C) \equiv \vee_{x \in C} t(x)$.

Moreover the set $\mathcal{C}$ of all clauses $C_j\ (j = 1, 2, \cdots, m)$ is called satisfiable iff the meet of all truth values of C_j is 1; $t(\mathcal{C}) \equiv \wedge_{j=1}^m t(C_j) = 1$. Thus the SAT problem is written as follows:

Definition 10 *SAT Problem: Given a Boolean set $X \equiv \{x_1, \cdots, x_n\}$and a set $\mathcal{C} = \{C_1, \cdots, C_m\}$ of clauses, determine whether $\mathcal{C}$ is satisfiable or not.*

That is, this problem is to ask whether there exists a truth assignment to make $\mathcal{C}$ satisfiable. It is known in usual algorithm that it is polynomial time to check the satisfiability only when a specific truth assignment is given, but we can not determine the satisfiability in polynomial time when an assignment is not specified.

In [3] we discussed the quantum algorithm of the SAT problem, which was rewritten in [7] with showing that the OM SAT-algorithm is combinatorial. In [1, 2] it is shown that the chaotic quantum algorithm can solve the SAT problem in polynomial time.

Ohya and Masuda pointed out [3] that the SAT problem, hence all other NP problems, can be solved in polynomial time by quantum computer if the superposition of two orthogonal vectors $|0\rangle$ and $|1\rangle$ is physically detected. However this detection is considered not to be possible in the present technology. The problem to be overcome is how to distinguish the pure vector $|0\rangle$ from the superposed one $\alpha|0\rangle + \beta|1\rangle$, obtained by the OM SAT-quantum algorithm, if β is not zero but very small. If such a distinction is possible, then we can solve the NPC problem in the polynomial time. In [1, 2] it is shown that it can be possible by combining nonlinear chaos amplifier with the quantum algorithm, which implies the existence of a mathematical algorithm solving NP=P. The algorithm of Ohya and Volovich is not known to be in the framework of quantum Turing algorithm or not. This aspect is studied in this talk.

4 SAT Algorithm

In this section, we explain the algorithm of the SAT problem which has been introduced by Ohya-Masuda [3] and developed by Accardi-Sabbadini [7]. The computation of the truth value can be done by by a combination of the unitary operators on a Hilbert space $\mathcal{H}$, so that the computation is described by the unitary quantum algorithm. The detail of this section is given in the papers [3, 7, 11, 12].

Throughout this section, let n be the total number of Boolean variables used in the SAT problem. Let $\mathcal{C}$ be a set of clauses whose cardinality is equal to m. Let $\mathbb{C}$ be the set of all complex numbers, and $|0\rangle$ and $|1\rangle$ be the two unit vectors $\binom{1}{0}$ and $\binom{0}{1}$, respectively. Then, for any two complex numbers α and β satisfying $|\alpha|^2 + |\beta|^2 = 1$, $\alpha|0\rangle + \beta|1\rangle$ is called a qubit. For any positive integer N, let $\mathcal{H}$ be the tensor product Hilbert space defined as $\left(\mathbb{C}^2\right)^{\otimes N}$ and let $\left\{|e_i\rangle; 0 \le i \le 2^{N-1}\right\}$ be the basis. For any two qubits $|x\rangle$ and $|y\rangle$, $|x, y\rangle$ and $\left|x^N\right\rangle$ is defined as $|x\rangle \otimes |y\rangle$ and $\underbrace{|x\rangle \otimes \cdots \otimes |x\rangle}_{N \text{ times}}$, respectively.

Let $\mathcal{H} = \left(\mathbb{C}^2\right)^{\otimes n+\mu+1}$ be a Hilbert space and $|v_0\rangle$ be the initial state $|v_0\rangle = |0^n, 0^\mu, 0\rangle$, where μ is the number of dust qubits which is determined in the paper [12]. Let $U_{\mathcal{C}}^{(n)}$ be a unitary operator for the computation of the SAT:

$$U_{\mathcal{C}}^{(n)} |v_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |e_i, x^\mu, t_{e_i}(\mathcal{C})\rangle \equiv |v_f\rangle$$

where x^μ denotes the μ strings in the dust bits and $t_{e_i}(\mathcal{C})$ is the truth value of $\mathcal{C}$ with e_i.

Applying the above unitary operator to the initial state, we obtain the final state ρ.The result of the computation is registered in the last section of the final vector, which will be taken out by a projection $P_{n+\mu,1} \equiv I^{\otimes n+\mu} \otimes |1\rangle\langle 1|$ onto the subspace of $\mathcal{H}$ spanned by the vectors $|\varepsilon^n, \varepsilon^\mu, 1\rangle$.

The following theorem is easily seen.

Theorem 11 *$\mathcal{C}$ is SAT if and only if*

$$P_{n+\mu,1} U_{\mathcal{C}}^{(n)} |v_0\rangle \neq 0$$

According to the standard theory of quantum measurement, after a measurement of the event $P_{n+\mu,1}$, the state $\rho = |v_f><v_f|$ becomes

$$\rho \to \frac{P_{n+\mu,1} \rho P_{n+\mu,1}}{Tr \rho P_{n+\mu,1}} =: \overline{\rho}$$

Thus the solvability of the SAT problem is reduced to check that $\rho' \neq 0$. The difficulty is that the probability

$$Tr \overline{\rho} P_{n+\mu,1} = \| P_{n+\mu,1} |v_f\rangle \|^2 = \frac{|T(\mathcal{C}_0)|}{2^n}$$

is very small in some cases, where $|T(\mathcal{C}_0)|$ is the cardinality of the set $T(\mathcal{C}_0)$, of all the truth functions t such that $t(\mathcal{C}_0) = 1$.

We put $q \equiv \sqrt{\frac{r}{2^n}}$ with $r \equiv |T(\mathcal{C}_0)|$. Then if r is suitably large to detect it, then the SAT problem is solved in polynomial time. However, for small r, the probability is very small so that we in fact do not get an information about the existence of the solution of the equation $t(C_0) = 1$, hence in such a case we need further deliberation.

Let go back to the SAT algorithm. After the quantum computation, the quantum computer will be in the state

$$|v_f\rangle = \sqrt{1-q^2} |\varphi_0\rangle \otimes |0\rangle + q |\varphi_1\rangle \otimes |1\rangle$$

where $|\varphi_1\rangle$ and $|\varphi_0\rangle$ are normalized $n+\mu$ qubit states and $q = \sqrt{r/2^n}$. Effectively our problem is reduced to the following 1 qubit problem: The above state $|v_f\rangle$ is reduced to the state

$$|\psi\rangle = \sqrt{1-q^2}\,|0\rangle + q\,|1\rangle\,,$$

and we want to distinguish between the cases $q = 0$ and $q > 0$(small positive number). Let us denote the correspondence from $\rho_0 \equiv |v_0\rangle\langle v_0|$ with ρ by a channel Λ_I; $\rho = \Lambda_I \rho_0$.

4.1 Chaotic dynamics

Various aspects of classical and quantum chaos have been the subject of numerous studies ([4, 11] and ref's therein). Here we will briefly review how chaos can play a constructive role in computation (see [1, 2] for the details).

Chaotic behavior in a classical system usually is considered as an exponential sensitivity to initial conditions. It is this sensitivity we would like to use to distinguish between the cases $q = 0$ and $q > 0$ discussed in the previous subsection.

Consider the so called logistic map which is given by the equation

$$x_{n+1} = ax_n(1-x_n) \equiv g(x), \quad x_n \in [0,1]\,.$$

The properties of the map depend on the parameter a. If we take, for example, $a = 3.71$, then the Lyapunov exponent is positive, the trajectory is very sensitive to the initial value and one has the chaotic behavior [2]. It is important to notice that if the initial value $x_0 = 0$, then $x_n = 0$ for all n.

The state $|\psi\rangle$ of the previous subsection is transformed into the density matrix of the form

$$\overline{\rho} = q^2 P_1 + \left(1-q^2\right) P_0$$

where P_1 and P_0 are projectors to the state vectors $|1\rangle$ and $|0\rangle$. One has to notice that P_1 and P_0 generate an Abelian algebra which can be considered as a classical system. The density matrix ρ above is interpreted as the initial data, and we apply the channel $\Lambda \equiv \Lambda_{CA}$ due to the logistic map as

$$\Lambda_{CA}\left(\overline{\rho}\right) = \frac{\left(I + g\left(\overline{\rho}\right)\sigma_3\right)}{2},$$

where I is the identity matrix and σ_3 is the z-component of Pauli matrices.

$$\overline{\rho}_k = \Lambda_{CA}^k\left(\overline{\rho}\right)$$

To find a proper value k we finally measure the value of σ_3 in the state ρ_k such that

$$M_k \equiv tr\overline{\rho}_k\sigma_3.$$

We obtain [2]

Theorem 12

$$\overline{\rho}_k = \frac{(I + g^k(q^2)\sigma_3)}{2}, \text{ and } M_k = g^k(q^2).$$

Thus the question is whether we can find such a k in polynomial steps of n satisfying the inequality $M_k \geq \frac{1}{2}$ for very small but non-zero q^2. Here we have to remark that if one has $q = 0$ then $\overline{\rho} = P_0$ and we obtain $M_k = 0$ for all k. If $q \neq 0$, the chaotic dynamics leads to the amplification of the small magnitude q in such a way that it can be detected. The transition from $\overline{\rho}$ to $\overline{\rho}_k$ is nonlinear and can be considered as a classical evolution because our algebra generated by P_0 and P_1 is abelian. The amplification can be done within at most 2n steps due to the following propositions. Since $g^k(q^2)$ is x_k of the logistic map $x_{k+1} = g(x_k)$ with $x_0 = q^2$, we use the notation x_k in the logistic map for simplicity.

Theorem 13 *For the logistic map $x_{n+1} = ax_n(1-x_n)$ with $a \in [0,4]$ and $x_0 \in [0,1]$, let x_0 be $\frac{1}{2^n}$ and a set J be $\{0,1,2,\ldots,n,\ldots,2n\}$. If a is 3.71, then there exists an integer k in J satisfying $x_k > \frac{1}{2}$.*

Theorem 14 *Let a and n be the same in above theorem. If there exists k in J such that $x_k > \frac{1}{2}$, then $k > \frac{n-1}{\log_2 3.71 - 1}$.*

Corollary 15 *If $x_0 \equiv \frac{r}{2^n}$ with $r \equiv |T(\mathcal{C})|$ and there exists k in J such that $x_k > \frac{1}{2}$, then there exists k satisfying the following inequality if $\mathcal{C}$ is SAT.*

$$\left[\frac{n-1-\log_2 r}{\log_2 3.71 - 1}\right] \leq k \leq \left[\frac{5}{4}(n-1)\right].$$

From these theorems, for all k, it holds

$$M_k \begin{cases} = 0 & \text{iff } \mathcal{C} \text{ is not SAT} \\ > 0 & \text{iff } \mathcal{C} \text{ is SAT} \end{cases}$$

5 SAT algorithm in GQTM

In this section, we construct the 3 multi-track GQTM $M_{SAT} = \left(Q, \Sigma^3, \mathcal{H}, \Lambda_\delta\right)$ that achieves OVM SAT algorithm. This GQTM doesn't belongs to LQTM and UQTM because the chaos amplification process is described by a non-linear CP channel not a uitary and linear one. The OVM algorithm runs from an initial state $\rho_0 \equiv |v_0\rangle\langle v_0|$ to $\overline{\rho}_k$ through $\rho \equiv |v_f\rangle\langle v_f|$ explained above. The computation from $\rho_0 \equiv |v_0\rangle\langle v_0|$ to $\rho \equiv |v_f\rangle\langle v_f|$ is due to unitary channel $\Lambda_C \equiv U_C \bullet U_C$, and that from $\rho \equiv |v_f\rangle\langle v_f|$ to $\overline{\rho}_k$ is due to a non-unitary channel $\Lambda_{CA}^{k} \circ \Lambda_{I}$, so that all computation can be done by $\Lambda_{CA}^{k} \circ \Lambda_{I} \circ \Lambda_{C}$, which is a completely positive, so the whole computation process is deterministic.

Let us explain our computation by a multi-track GQTM. The first track stores the input data and literals. The second track is used for the computation of $f(C_i), (i = 1, \cdots, m)$, and the thired track is used for the computation of $f(\mathcal{C})$ denoting the result. This algorithm is represented by the following 5 steps:

- Step 1 : Apply the Hadamard transform to Track 1.
- Step 2 : Calculate $f(C_1), \cdots f(C_m)$ and store them in Track 2.
- Step 3 : Calculate $f(\mathcal{C})$, and store it in Track 3.
- Step 4 : Empty the working space.
- Step 5 : Apply the chaos amplifier to the result state and repeat this step.

The detail of this quantum algorithm is explained in the paper [12].

5.1 Computational complexity of the SAT algorithm

We define the computational complexity of the OV SAT algorithm as the product of $T_Q\left(U_{\mathcal{C}}^{(n)}\right)$ and $T_{CA}(n)$,where $T_Q\left(U_{\mathcal{C}}^{(n)}\right)$ is the complexity of unitary computation and $T_{CA}(n)$ is that of chaos amplification.

The following theorem is essentially discussed in [2, 3, 12].

Theorem 16 *For a set of clauses $\mathcal{C}$ and n Boolean variables, the computational complexity of the OV SAT algorithm including the chaos amplifier, denoted by $T(\mathcal{C}, n)$, is obtained as follows.*

$$T_{GQTM}(\mathcal{C}, n) = T_Q\left(U_{\mathcal{C}}^{(n)}\right) T_{CA}(n) = \mathcal{O}(poly(n)),$$

where $poly(n)$ denotes a polynomial of n.

The computational complexity of quantum computer is determined by the total number of logical quantum gates. This inequality implies that the computational complexity of SAT algorithm is bounded by $\mathcal{O}(n)$ for the size of input n while a classical algorithm is bounded by $\mathcal{O}(2^n)$.

References:

[1] M.Ohya and I.V.Volovich, *Quantum computing and chaotic amplification*, J. opt. B, **5**,No.6 639-642, 2003.

[2] M.Ohya and I.V.Volovich, *New quantum algorithm for studying NP-complete problems*, Rep.Math.Phys., **52**, No.1,25-33 2003.

[3] M.Ohya and N.Masuda, *NP problem in Quantum Algorithm*, Open Systems and Information Dynamics, 7 No.1 33-39, 2000.

[4] M.Ohya, *Complexities and Their Applications to Characterization of Chaos*, Int. Journ. of Theoret. Physics, **37** 495, 1998.

[5] L.Accardi and M.Ohya, Compound channels, transition expectations, and liftings, Appl. Math. Optim., Vol.39, 33-59, 1999.

[6] L.Accardi and M.Ohya (2004) A Stochastic Limit Approach to the SAT Problem,Open Systems and Information dynamics, 11,1-16.

[7] L.Accardi and R.Sabbadini, On the Ohya–Masuda quantum SAT Algorithm, Preprint Volterra, N. **432**, 2000.

[8] D.Deutsch, Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer, Proc. Roy. Soc, A, **400** 97-117, 1985.

[9] E.Bernstein and U.Vazirani, *Quantum Complexity Theory*, In Proc. 25th ACM Symp. on Theory of Computation, 11–20, 1993.

[10] H. Nishimura and M. Ozawa: Computational Complexity of Uniform Quantum Circuit Families and Quantum Turing machines, quant-ph/9906095, 2000.

[11] M.Ohya and I.V.Volovich, Mathematical Foundation of *Quantum Information and Quantum Computation,* to be published.

[12] S.Iriyama and M.Ohya, Rigorous Estimate for OMV SAT Algorithm to appear in OSID, 2007.