The computing power of Turing machine based on quantum logic

Yun Shang[1,2]*, Xian Lu[1] and Ruqian Lu[1,2,3]

[1] Institute of Mathematics, AMSS, CAS, Beijing 100190, P.R.China
[2] NCMIS, AMSS, CAS, Beijing, P.R.China
[3] CAS Key Lab of IIP, Institute of Computing Technology, CAS
shangyun602@163.com, luxian@amss.ac.cn, rqlu@math.ac.cn

Abstract

Super-Turing computational power has been invoked by models of computation that make use of new physical principles or of different underlying logic of computation. Recently, it has been observed that Turing machines based on quantum logic can solve undecidable problems. In this paper we will give recursion-theoretical characterization of the computational power of this kind of quantum Turing machines. In detail, for the unsharp case, it is proved that $\Sigma_1^0 \cup \Pi_1^0 \subseteq L_d^T(\mathcal{E}, \Sigma)(L_w^T(\mathcal{E}, \Sigma)) \subseteq \Pi_2^0$ when the truth value lattice is locally finite and the operation $\wedge$ is computable, where $L_d^T(\mathcal{E}, \Sigma)(L_w^T(\mathcal{E}, \Sigma))$denotes the class of quantum language accepted by these Turing machine in depth-first model (respectively, width-first model); for the sharp case, we can obtain similar results for usual orthomodular lattices.

1 Introduction

The classical theory of computation deals with basic problems of computation such as computability, computational complexity, apparatus and models of computation, algorithmic theory, etc. Among these, the apparatus and models of computation play an important role since the thirties of last century. The most significant concepts belong to this area include theory of recursive functions, the lambda calculus, the automata theory, and many others. They form the basis and foundation of computation and become a mature branch of computing science. Nonetheless, despite of our optimistic feeling that the classical theory of computation may provide us with everything we need in classical computing, a brand new concept of computation was born in early eighties of last century. This new concept is called quantum computer and was proposed by the great physicist Feynman [7]. People have soon found that we need also a brand new theory of computation—the theory of quantum computation, of which the theory of quantum automata has been one of the most interesting topics.

The study of quantum automata was technically divided in two directions. Due to the well known probabilistic character of quantum mechanics, the first line of quantum automata research is based on probabilistic representation of quantum information. Only two years after the publication of Feynman's seminal paper, it was Deutsch who has published a model of quantum Turing machine [5]. Moore and Crutchfield published the first model of quantum finite state automata, which is a measure once model [18]. (Measuring is done only once when the computation is finished). Almost at the same time Kondacs and Watrous published their model of quantum finite automata, which is a measure many model (Measuring is done many

*This work was supported by NSFC projects under Grant No.61073023. 973project under Grant NO. 2009CB320701,K.C.Wong Education Foundation, Hong Kong, National Center for Mathematics and Interdisciplinary Sciences, CAS

times whenever a letter is read in)[15]. Other models of quantum automata include the one way (or two way) one counter quantum automata [1, 28]. Roughly speaking, all these models are generalizations of probabilistic automata.

Another direction of quantum automata research has followed a more abstract line. Considering that the quantum logic proposed by von Neumann [2] has the orthomodular lattice as its algebraic model, it is very suggestive to take the orthomodular lattice as the value domain of accepting a letter string by a quantum automaton, instead of taking the Boolean algebra as value domain. More exactly, while a classical finite state automaton either accepts (acceptance degree = 1) an input string, or rejects it (acceptance degree = 0), a quantum finite state automaton accepts an input string 'to some degree' ($0 \leq$ acceptance degree ≤ 1), which is an element of the orthomodular lattice. Note that the orthomodular lattice is a theoretical model of Hilbert space (i.e. representing the set of all its closed linear subspaces). It was Mingsheng Ying who has published the first paper in this direction [29, 30]. In this and the following papers, Ying studied quantum finite state automata based on orthomodular lattices and established the first framework of theory of quantum computation based on such automata [31, 32]. In particular, Ying found that many good properties of classical finite state automata do not exist in quantum finite state automata. The lack of these good properties is due to the absence of distributive law in orthomodular lattice, while this law plays a fundamental role in Boolean algebra. For other works on automata based on orthomodular lattice see [16, 19, 20].

One of the most important supporting pillars of quantum logic based orthomodular lattice is the PV (projection valued) measurement. With the PV measurement developed into POV measurement(positive operator valued) measurement, quantum logic is developed into unsharp quantum logic where the unsharpness comes from unsharp observables in quantum measurement [17]. Certainly, PV measurements correspond to sharp observables, and von Neumann quantum logic is called sharp quantum logic. From the logical point of view, it means that quantum events don't satisfy the non-contradiction law in unsharp quantum logic, but do satisfy the non contradiction law in sharp quantum logic. Many algebraic structures were proposed to represent quantum events in unsharp quantum logic. In 1994, Foulis introduced effect algebras, which is the main model for unsharp quantum logic [8]. MV algebras, as the algebraic model of multiple valued logic, play an analogous role to that of Boolean algebras in sharp quantum logic [3, 6]. They are the blocks of lattice ordered effect algebras [21]. Quantum MV (QMV) algebras are another important kind of unsharp quantum structures [9]. They are not only a non-lattice theoretic generalization of MV algebras, but also a non-idempotent generalization of orthomodular lattice. It is known that the center of a QMV algebra is an MV algebra [10].

Since unsharp quantum logic is more universal than sharp quantum logic, Shang etc. set up finite state automata and pushdown automata theory based on unsharp quantum logic. We found that some important properties of classical finite state automata and classical pushdown automata are universally valid in unsharp quantum logic based automata if and only if the underlying algebraic model is an MV algebra [22, 23]. We have also studied unsharp quantum logic based Turing machine, whose algebraic models are extended lattice ordered effect algebras and lattice ordered QMV algebras [24]. Yongming Li etc. studied Turing machine based on orthomodular lattice [14]. From the recursion-theoretical viewpoint, we find that deterministic Turing machine based on quantum logic is not equivalent to non deterministic Turing machine. In detail, we find that non deterministic Turing machine possess a super-Turing computational power: it can recognize r.e. language and co-r.e. language. That is, it can solve the undecidable problem [24]. Since J.Wiedermann proved that classical fuzzy Turing machine also possess the super-Turing computational power in [27], we want to know what is the relation between these two kinds of Turing machines? In detail, what is the boundary between their language? Who has more computing power?

In this paper, we will partly answer the question from computability and complexity aspects. In detail, from the arithmetical hierarchy viewpoint, we find that the class of the language of Turing machine based on quantum logic is between the first and the second level of the arithmetical hierarchy. Namely, the lower boundary of language of Turing machine based on quantum logic is $\Sigma_1^0 \cup \Pi_1^0$ (the union of r.e. language (class Σ_1^0) and co-r.e. language(class Π_1^0)), and its upper boundary is class Π_2^0. Compared with the results of J. Wiedermann[26, 27], we can conclude that Turing machine based on quantum logic is more powerful than classical fuzzy Turing machine and classical Turing machine. From these results, it is easy to see that the power of computing models depends heavily on the underlying logic.

The rest of this paper is organized as follows: the basic knowledge about extended lattice ordered effect algebra and lattice ordered QMV algebra will be recalled in Section 2. We use the symbol $\mathcal{E}$ to denote these two algebras. In section 3, some basic knowledge about $\mathcal{E}$-valued Turing machine will be recalled. In section 4 we prove that the language of $\mathcal{E}$-valued Turing machine is between $\Sigma_1^0 \cup \Pi_1^0$ and Π_2^0 when $\mathcal{E}$ is locally finite and the operation $\wedge$ is computable. In Section 5, let the orthomodular lattices (denoted as $\mathcal{L}$) to be the truth value domain of von Neumann quantum logic, and we obtain the similar results for the general orthomodular lattices. Finally, we give some conclusions in the last paragraph.

2 Lattice ordered QMV algebras

First, we recall some notions and results in unsharp quantum logic.

Definition 2.1. [4] A quantum MV algebra (QMV algebra) is a structure $\mathcal{M} = (M, \boxplus, ', \mathbf{0}, \mathbf{1})$ where $\boxplus$ is a binary operation, $'$ is a unary operation, and $\mathbf{0}$, $\mathbf{1}$ are special distinct elements of M. For any $a, b \in M$: $a \odot b := (a' \boxplus b')'$, $a \sqcap b := (a \boxplus b') \odot b$, $a \sqcup b := (a \odot b') \boxplus b$. Assume that the following conditions hold:

(QMV1) $a \boxplus b = b \boxplus a$;
(QMV2) $a \boxplus (b \boxplus c) = (a \boxplus b) \boxplus c$;
(QMV3) $a \boxplus a' = \mathbf{1}$;
(QMV4) $a \boxplus \mathbf{0} = a$;
(QMV5) $a \boxplus \mathbf{1} = \mathbf{1}$;
(QMV6) $a'' = a$;
(QMV7) $a \boxplus [(a' \sqcap b) \sqcap (c \sqcap a')] = (a \boxplus b) \sqcap (a \boxplus c)$.

Axiom (QMV1) says that the operation $\oplus$ is commutative. Axiom (QMV2) says further that it is associative. Axiom (QMV3) represents the law of excluded middle with regard to $\boxplus$ if we consider a' is the negation of a and $\boxplus$ as some kind of “or” operation. Axiom (QMV4) shows that $\boxplus$ has a neutral element $\mathbf{0}$. It can be also seen as a generalized join operation of $\mathbf{0}$ and a. On the other hand, Axiom (QMV5) can be seen as a generalized join of $\mathbf{1}$ and a. Axiom (QMV6) implies the double negation principle.

It is easy to infer the following facts:

Proposition 2.1. [4] (i) $\mathbf{0}' = \mathbf{1}$;
(ii) $\sqcap$ is commutative iff $\sqcup$ is commutative;
(iii) $\sqcap$ is commutative if the QMV algebra degenerates to MV algebra.

Definition 2.2. For all elements of a QMV algebra, let $a \leq b$ iff $a = a \sqcap b$.

In this way we introduced a partial ordering in the QMV algebra. However, even with this partial ordering, a QMV algebra is not necessarily a lattice.

Definition 2.3. A QMV algebra is called a lattice ordered QMV algebra, if the partial ordering introduced in definition 2.2 forms a lattice. Let $\vee$ denotes the join operation and $\wedge$ denotes the meet operation in a lattice ordered QMV algebra, where $a \vee b = c$ iff $c = \min\{d|a, b \leq d\}$, $a \wedge b = c$ iff $c = \max\{d|d \leq a, b\}$. If $a \leq b$, there is $a = a \sqcap b = a \wedge b$. But in general $a \sqcap b \neq a \wedge b$.

For an lattice ordered QMV algebra, there is $a \wedge a' \neq \mathbf{0}$ and $a \vee a' \neq \mathbf{1}$ in general. So the non-contradiction law and the law of excluded middle do not hold. For example

Example 2.1. Consider QMV algebra $M = \{0, 1, a, b\}$ with the operations $\boxplus$ and $'$ defined as $a = a' \neq b = b'$, $a \boxplus b = \mathbf{1}$ and $\mathbf{1} \boxplus x = 1$ for any $x \in M$. As shown in Figure 1. This QMV algebra is called $\mathfrak{M}_4$ [9], it is lattice ordered and in fact is the smallest QMV algebra that is not an MV algebra. In $\mathfrak{M}_4$ there is $a \wedge a' = a \neq \mathbf{0}$ and $a \vee a' = a \neq \mathbf{1}$.

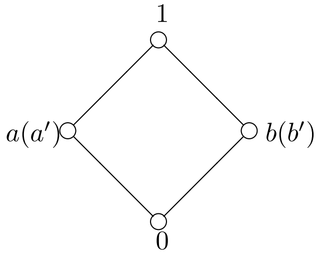

Figure 1: $\mathfrak{M}_4$

Every MV algebra is a QMV algebra, however, there exists a QMV algebra which is not an MV algebra (Example 2.7, [22]).

Definition 2.4. [6] A QMV algebra is said to be locally finite iff $\forall a \in \mathcal{M}$ s.t. $a \neq 0$, $\exists n \in \mathbb{N}$ s.t. $n \cdot a = 1$.

Theorem 2.2. [22] Let $\mathcal{E} = (E, \boxplus, ', \mathbf{0}, \mathbf{1})$ be a lattice ordered QMV algebra. The following conditions are equivalent:
(i) $\mathcal{E}$ is an MV algebra.
(ii) $(a \boxplus b) \wedge (a \boxplus c) = a \boxplus (b \wedge c)$ for any $a, b, c \in E$.

3 $\mathcal{E}$-valued Turing machine

Now, let unsharp quantum logic be the truth value set of the propositions, we set up Turing machine based on unsharp quantum logic. In the following, $\mathcal{E}$ denotes a lattice ordered QMV algebra. If $\mathcal{E}$ denotes an extended lattice ordered effect algebra, we can obtain Turing machine based on extended lattice ordered effect algebra without literal modification.

Definition 3.1. An $\mathcal{E}$-valued non-deterministic Turing machine ($\mathcal{E}$NTM) is a 7-tuple: $M = (Q, \Sigma, \Gamma, \delta, B, I, T)$, where

1. Q is a finite nonempty state set.
2. Σ is the finite set of input symbols.
3. Γ is the complete set of tape symbols; $\Sigma \subseteq \Gamma/B$.
4. $\delta : Q \times \Gamma \times Q \times \Gamma \times \{L, S, R\} \longrightarrow \mathcal{E}$ is the transition function. The symbols L, R and S indicate that the head of the $\mathcal{E}$NTM moves left, right or keep stationary.

5. B is the blank symbol. The blank symbol appears initially in all but the finite number of initial cells that hold input symbols.
6. $I : Q \longrightarrow \mathcal{E}$ is the initial state function.
7. $T : Q \longrightarrow \mathcal{E}$ is the final or accepting state function.

As defined in classical Turning machines, a configuration or instantaneous description (ID) of an $\mathcal{E}$NTM M is a sequence $C = \alpha_1 q \alpha_2$, where $q \in Q$ and $\alpha_1\alpha_2$ is the finite sequence between the leftmost and the rightmost nonblanks. We denote the state of C to be $St(C)$ and denote $ID(M)$ to be the set of all instantaneous descriptions of M. An $\mathcal{E}$NTM in ID $\alpha_1 q \alpha_2$ means the current state is q and the reading head is looking at the first symbol of α_2. The value of M transforms from C_1 to C_2 is described as

$$\delta^{\dagger}(C_1, C_2) = \begin{cases} \delta(p,a,q,b,L), & \text{if } C_1 = \alpha cpa\beta \text{ and } C_2 = \alpha qcb\beta \\ \delta(p,a,q,b,S), & \text{if } C_1 = \alpha pa\beta \text{ and } C_2 = \alpha qb\beta \\ \delta(p,a,q,b,R), & \text{if } C_1 = \alpha pa\beta \text{ and } C_2 = \alpha bq\beta \\ 1, & \text{otherwise} \end{cases}$$

where $a, b, c \in \Gamma$ and $\alpha, \beta \in \Gamma^*$ with that the leftmost symbol of α and the rightmost symbol of β are not B. We denote $\vdash (C_1, C_2) = (p, a, q, b, D)$ if the $\mathcal{E}$NTM could transform to C_2 from C_1 through the transition (p, a, q, b, D).

Example 3.1. Figure 2 shows the transfer function of a nondeterministic quantum Turing machine based on lattice ordered QMV algebra. Starting from state p, where the $\mathcal{E}$NTM's reading head points to the symbol a, it overwrites the symbol a with symbol b, changes the state p to some new state q, then moves left (L), moves right (R), or keeps still (S). The "fuzzy degree" of the first possibility is x, while that of the second possibility is y. Both x and y are elements of the lattice shown below. It means that moving left or remaining at original place are both possible. But it is impossible to move to right (fuzzy degree is $\mathbf{1}$).

$$\delta^{\dagger}(C_1, C_2) = \begin{cases} \delta(p,a,q,b,L) = x, & \text{if } C_1 = \alpha cpa\beta \text{ and } C_2 = \alpha qcb\beta \\ \delta(p,a,q,b,S) = y, & \text{if } C_1 = \alpha pa\beta \text{ and } C_2 = \alpha qb\beta \\ \delta(p,a,q,b,R) = \mathbf{1}, & \text{if } C_1 = \alpha pa\beta \text{ and } C_2 = \alpha bq\beta \end{cases}$$

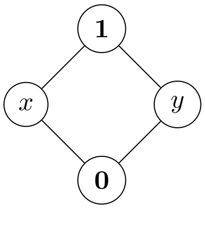

Figure 2: non-deterministic quantum Turing machine

Similar to finite state automata theory based on unsharp quantum logic, by interacting $\wedge$ and $\boxplus$, we can adapt depth-first and width-first models respectively to define the acceptance degree of language recognized by Turing machine. It is proved that only when the truth-lattice is an MV algebra, the two models coincide.

Definition 3.2. A path of an $\mathcal{E}$NTM M is a finite sequence of IDs.

Definition 3.3. The $\mathcal{E}$-valued language accepted by an $\mathcal{E}$NTM M in depth-first model is defined to be

$$|M|_d(s) = \bigwedge_{n\geq 1} \bigwedge_{C_i} \bigwedge_{q_0 \in Q} I(q_0) \boxplus \delta^\dagger(q_0 s, C_1) \boxplus \delta^\dagger(C_1, C_2) \boxplus \cdots \boxplus T(St(C_n)) \tag{1}$$

for any $s \in \Sigma^+$.

Definition 3.4. The $\mathcal{E}$-valued language accepted by an $\mathcal{E}$NTM M in width-first model is defined to be

$$\begin{aligned} |M|_w(s) = \bigwedge_{n\geq 1} \Bigg[\bigwedge_{C_n} \Bigg(\cdots \Bigg(\bigwedge_{C_2} \Bigg(\bigwedge_{C_1} \Bigg(\bigwedge_{q_0} I(q_0) \boxplus \delta^\dagger(q_0 s, C_1) \Bigg) \boxplus \delta^\dagger(C_1, C_2) \Bigg) \\ \boxplus \delta^\dagger(C_2, C_3) \Bigg) \cdots \Bigg) \boxplus T(St(C_n)) \Bigg] \end{aligned} \tag{2}$$

for any $s \in \Sigma^+$.

Remark 3.1. Similar to classical Turing machines, an $\mathcal{E}$NTM M halts when gets into some state q with $T(q) < 1$, or gets into some ID C with $T(St(C)) = 1$ and $\delta^\dagger(C, C') = 1$ for any ID C'. Each path in Equations (1) and (2) is required to halt. If the machine does not halt for some input s in all pathes, then the $\mathcal{E}$-value of s accepted by M is not defined.

Remark 3.2. Since there is no product operation but "$\boxplus$" in effect algebras, and in physics $\boxplus$ depicts the coexistence among effects, we use it to connect the effects of adjacent automaton actions. The counterintuitive result is that, 0 means totally acceptance and 1 means absolutely rejection.

Definition 3.5. An $\mathcal{E}$-valued deterministic Turing machine ($\mathcal{E}$DTM) is an $\mathcal{E}$NTM whose transition function δ satisfies that, for any $p \in Q$ and $a \in \Gamma$ there exists at most one set $\{q, b, D\}$ such that $\delta(p, a, q, b, D) \neq 1$.

The classes of all $\mathcal{E}$NTM and $\mathcal{E}$DTM over alphabet Σ are denoted by NTM($\mathcal{E}, \Sigma$) and DTM($\mathcal{E}, \Sigma$) respectively. Denote $L_d^T(\mathcal{E}, \Sigma) = \{|M|_d : M \in \text{NTM}(\mathcal{E}, \Sigma)\}$ and $L_w^T(\mathcal{E}, \Sigma) = \{|M|_w : M \in \text{NTM}(\mathcal{E}, \Sigma)\}$.

Definition 3.6. An partial function $L : \Sigma^+ \to \mathcal{E}$ is called $\mathcal{E}$-valued d-recursively enumerable (d-R.E.) language or $\mathcal{E}$-valued w-recursively enumerable (w-R.E.) language if $L \in L_d^T(\mathcal{E}, \Sigma)$ or $L \in L_w^T(\mathcal{E}, \Sigma)$.

Proposition 3.1. [24]

(i) $|M|_w \leq |M|_d$ for any $\mathcal{E}$NTM M.

(ii) $|M|_w = |M|_d$ for any $\mathcal{E}$NTM M iff $\mathcal{E}$ is an MV algebra.

4 The language of $\mathcal{E}$NTM

We consider the arithmetical hierarchy [25] of the languages of $\mathcal{E}$NTM in depth-first model as well as in width-first model. Before that we list some useful results. It was proved that general $\mathcal{E}$NTM are equivalent to $\mathcal{E}$NTM in special forms.

Definition 4.1. Let $M = (Q, \Sigma, \Gamma, \delta, B, I, T) \in \mathcal{E}$NTM. We call δ to be classical if $\delta(p, a, q, b, D) = 0$ or 1, $\forall p, q \in Q$, $\forall a, b \in \Gamma$ and $\forall D \in \{L, S, R\}$. Similarly we call I (T) to be classical if $I(p) = 0$ or 1 ($T(p) = 0$ or 1), $\forall p \in Q$.

Lemma 4.1. [24] For any $M \in \mathrm{NTM}(\mathcal{E}, \Sigma)$ there exists M_I with classical initial state function such that $|M|_d = |M_I|_d$ and $|M|_w = |M_I|_w$.

In fact we could always assume there is single initial state, and denote $M = (Q, \Sigma, \Gamma, \delta, B, p_I, T)$ if needed.

Denote the range of a map f to be $R(f)$. For an $\mathcal{E}$NTM $M = (Q, \Sigma, \Gamma, \delta, B, I, T)$ denote $R_M = R(I) \cup R(\delta) \cup R(T)$. We already have the following conclusions:

Theorem 4.2. [24] Let M be an $\mathcal{E}$NTM and let S_M denote the subalgebra generated by R_M. If S_M is finite, there exists an $\mathcal{E}$NTM $\bar{M}$ with classical transitions such that $|M|_w = |\bar{M}|_w$.

Lemma 4.3. [24] Let M be an $\mathcal{E}$NTM. If $\mathcal{E}$ is locally finite, there exists some $\mathcal{E}$NTM M^c with classical transitions which accepts the same $\mathcal{E}$-valued language.

So in both models the transitions could be classical in some conditions.

Definition 4.2. A path $(C_0, \cdots, C_n)$ is effective if $\delta^\dagger(C_{i-1}, C_i) \neq 1$ for $i = 1, \cdots, n$. In an effective path each $\delta^\dagger(C_{i-1}, C_i) = \delta(St(C_{i-1}), a, St(C_i), b, D)$ for some $a, b \in \Gamma$ and $D \in \{L, S, R\}$.

Definition 4.3. Let $M = (Q, \Sigma, \Gamma, \delta, B, p_I, T)$ be an $\mathcal{E}$NTM. For any $s \in \Sigma^+$, define

$$ID_M(s, 1) = \{C \in ID(M) : (p_I s, C) \text{ is an effective path}\}$$

and

$$ID_M(s, n+1) = \{C \in ID(M) : (C', C) \text{ is an effective path for some } C' \in ID_M(s, n)\}$$

$n = 1, 2, \cdots$. Denote $ID_M(s) = \bigcup_n ID_M(s, n)$ which is comprised by all the IDs that is achievable from $p_I s$. We omit the subscript M if no confusion is possible.

In [27] Wiedermann proved that the language of classical Fuzzy-NTMs is exactly the union of r.e. languages (class Σ_1^0) and their complements (class Π_1^0). In the following we obtain the similar properties for $\mathcal{E}$NTM, with computable "$\wedge$" operation.

Theorem 4.4. $\Sigma_1^0 \cup \Pi_1^0 \subseteq L_d^T(\mathcal{E}, \Sigma)$.

Proof. Let $L \in \Sigma_1^0 \cup \Pi_1^0$. If $L \in \Sigma_1^0$, assume the classical Turing machine accepting L is $M' = (Q', \Sigma, \Gamma, \delta', q_0, B, F)$ (referring to [12]). Similar to Example 4.1 in [24], construct the $\mathcal{E}$NTM $M = (Q, \Sigma, \Gamma, \delta, B, I, T)$ as:

- $Q = Q' \cup \{q_1\}$.
- $I(q_0) = 0$ and $I(q) = 1$ otherwise.
- $T(q) = 0$ if $q \in F$, $T(q_1) = 0$ and $T(q) = 1$ otherwise.
- $\delta(p, a, q, b, D) = 0$ if (p, a, q, b, D) is a transition of M' and $\delta = 1$ otherwise.

Easy to check that $|M|_d(s) = 0$ for any $s \in L$ and $|M|_d(s) = 1$ for all $s \notin L$. So $\Sigma_1^0 \subseteq L_d^T(\mathcal{E}, \Sigma)$.

If $L \in \Pi_1^0$, assume the classical Turing machine accepting $\overline{L}$ (the complement of L) is M'. As above, there is a $\mathcal{E}$NTM M such that $|M|_d(s) = 0$ for any $s \notin L$ and $|M|_d(s) = 1$ for all $s \in L$. So $\Pi_1^0 \subseteq L_d^T(\mathcal{E}, \Sigma)$. □

Theorem 4.5. $\Sigma_1^0 \cup \Pi_1^0 \subseteq L_w^T(\mathcal{E}, \Sigma)$.

Proof. Note that in Theorem 4.4, the $\mathcal{E}$NTM M is in fact based on a trivial subalgebra $\{0, 1\}$ of $\mathcal{E}$. Obviously $\{0, 1\}$ is a MV algebra. According to Proposition 3.1 there is $|M|_d = |M|_w$. It follows that we could prove $\Sigma_1^0 \subseteq L_w^T(\mathcal{E}, \Sigma)$ and $\Pi_1^0 \subseteq L_w^T(\mathcal{E}, \Sigma)$ with the same reasoning in Theorem 4.4. □

Theorem 4.6. If $\mathcal{E}$ is locally finite, then $L_d^T(\mathcal{E}, \Sigma) \subseteq \Pi_2^0$.

Proof. Let M be an $\mathcal{E}$NTM. Here we treat any $\mathcal{E}$-valued language $L \in L_d^T(\mathcal{E}, \Sigma)$ equivalently as its graph $\{s\#L(s) : s \in \Sigma^*\}$, and assume there is single classical initial state as described before. Since $\mathcal{E}$ is locally finite, the transitions could be classical according to Lemma 4.3. So the $\mathcal{E}$-value of each path is totally determined by the final state. Define

$$\begin{aligned} L_1 =& \{s\#e : \forall C \in ID(s), T(St(C)) \geq e\} \\ L_2 =& \{s\#e : \exists \varepsilon \in \mathcal{E}, \forall C \in ID(s), T(St(C)) \geq e \boxplus \varepsilon \text{ and } \varepsilon \neq 0\}. \end{aligned}$$

For $\forall s\#e \in L_1$, e is a lower bound of $|M|_d(s)$; for $\forall s\#e \in L_2$, e is not the greatest lower bound of the set $\{T(St(C)) : C \in ID(s)\}$. By the definition of Equation (1), $|M|_d(s)$ should be the greatest lower bound of the $\mathcal{E}$-value of all paths accepting s, further the greatest lower bound of $\{T(St(C)) : C \in ID(s)\}$ by Lemma 4.3. Therefore the language of M is $L_1 - L_2$. First, consider a nondeterministic Turing machine M_1 that guesses C to see whether $C \in ID(s)$. Then, since $\wedge$ is computable, M_1 can decide the order relation of $T(St(C))$ and e as follows:

1. computing $T(St(C)) \wedge e$;
2. compare $T(St(C)) \wedge e$ with $T(St(C))$, and with e:
   (a) If $T(St(C)) \wedge e = T(St(C))$, then $e \geq T(St(C))$;
   (b) If $T(St(C)) \wedge e = e$, then $T(St(C)) \geq e$;
   (c) If $T(St(C)) \wedge e \neq T(St(C))$ nor e, then $T(St(C))$ and e are not compatible.

That is, if $\wedge$ is computable, so is the order $\geq$. Thus $L_1 \in \Pi_1^0$, $L_2 \in \Sigma_2^0$ and $L_1 - L_2 = L_1 \cap L_2' \in \Pi_2^0$. □

Theorem 4.7. If $\mathcal{E}$ is finite, then $L_w^T(\mathcal{E}, \Sigma) \subseteq \Pi_2^0$.

Proof. Let M be an $\mathcal{E}$NTM. Since S_M is finite, we could assume that the transitions of M are classical by Theorem 4.2. That is, the $\mathcal{E}$-value of each input s is determined by the final ID. With the same technology of Theorem 4.6 we obtain the result. □

If the $\mathcal{E}$ is linear additionally, the classes $L_d^T(\mathcal{E}, \Sigma)$ as well as $L_w^T(\mathcal{E}, \Sigma)$ coincide with the language of classical fuzzy Turing machines[27].

Theorem 4.8. If $\mathcal{E}$ is locally finite and linear, then $L_d^T(\mathcal{E}, \Sigma) = L_w^T(\mathcal{E}, \Sigma) = \Sigma_1^0 \cup \Pi_1^0$.

Proof. First, Theorem 4.4 and Theorem 4.5 show that $\Sigma_1^0 \cup \Pi_1^0 \subseteq L_d^T(\mathcal{E}, \Sigma)$ and $\Sigma_1^0 \cup \Pi_1^0 \subseteq L_w^T(\mathcal{E}, \Sigma)$.

Let $M = (Q, \Sigma, \Gamma, \delta, B, p_I, T)$ be an $\mathcal{E}$NTM, and assume the operation $\wedge$ is computable. Since $\mathcal{E}$ is locally finite, we assume the transitions are classical. Define

$$\begin{aligned} L_1 =& \{s\#e : \exists C \in ID(s), T(St(C)) = e\} \\ L_2 =& \{s\#e : \forall C \in ID(s), T(St(C)) \geq e\}. \end{aligned}$$

Then there is a classical nondeterministic Turing machine M_1 that guesses C and simulates M on s to check whether $C \in ID(s)$ and $T(St(C)) = e$. So $L_1 \in \Sigma_1^0$ and similarly $L_2 \in \Pi_1^0$, the language is $L_1 \cap L_2 \in \Sigma_1^0 \cup \Pi_1^0$. Note that since $\mathcal{E}$ is linear, for any $s \in \Sigma^*$ there is $s\#|M|_d(s) \in L_1$. Otherwise there may be $L_1 \cap L_2 = \emptyset$. That is $L_d^T(\mathcal{E}, \Sigma) \subseteq \Sigma_1^0 \cup \Pi_1^0$, and therefore $L_d^T(\mathcal{E}, \Sigma) = \Sigma_1^0 \cup \Pi_1^0$.

Since $\mathcal{E}$ will degenerate to be an MV algebra when it is linear, the operation $\boxplus$ distributes over $\wedge$ according to Theorem 2.2. So Equations (1) and (2) coincide if $\mathcal{E}$ is linear. It follows that $L_d^T(\mathcal{E}, \Sigma) = L_w^T(\mathcal{E}, \Sigma) = \Sigma_1^0 \cup \Pi_1^0$. □

5 The language of $\mathcal{L}$NTM

It is well known that, unsharp quantum logic is the generalization of sharp quantum logic. The algebraic model of sharp quantum logic is orthomodular lattice [13].

Definition 5.1. An orthomodular lattice is a 7-tuple $\mathcal{L} = (L, \leq, \wedge, \vee, *, 0, 1)$ where:

1. $(L, \leq, \wedge, \vee, 0, 1)$ is a bounded lattice.
2. $*$ is a unary operation on L, satisfying
   (a) $a \wedge a^* = 0$, $a \vee a^* = 1$;
   (b) $a^{**} = a$;
   (c) $a \leq b \Rightarrow b^* \leq a^*$.
3. The orthomodular law holds: $\forall a, b \in L$, $a \leq b \Rightarrow a \vee (a^* \wedge b) = b$.

If $\mathcal{E}$ degenerated to an orthmodular lattice $\mathcal{L}$, the $\mathcal{E}$NTM defined in Definition 3.1 is the Turing machine based on sharp quantum logic. Denote these Turing machine based on sharp quantum logic as $\mathcal{L}$NTM, note that the "$\boxplus$" operations in Equation (1) and Equation (2) turn into "$\vee$" in $\mathcal{L}$NTM.

When $\mathcal{E}$ degenerated into $\mathcal{L}$ in the proof of Theorem 4.4 and Theorem 4.5, we get a proof of the following result:

Theorem 5.1. $\Sigma_1^0 \cup \Pi_1^0 \subseteq L_d^T(\mathcal{L}, \Sigma)$.

Theorem 5.2. $\Sigma_1^0 \cup \Pi_1^0 \subseteq L_w^T(\mathcal{L}, \Sigma)$.

Let M be an $\mathcal{L}$NTM. In Theorem 2.2 of [14] it was proved that there exists some $\mathcal{L}$NTM M^C with classical transitions, and that M and M^C are equivalent in the depth-first model. Analogous to Theorem 4.6, it follows that

Theorem 5.3. $L_d^T(\mathcal{L}, \Sigma) \subseteq \Pi_2^0$.

Since the S_M (the subalgebra of the R_M) is not finite for general orthomodular lattice, we have the following conclusions for special lattice.

Theorem 5.4. If $\mathcal{L}$ is finite, then $L_w^T(\mathcal{L}, \Sigma) \subseteq \Pi_2^0$.

Corollary 5.5. If $\mathcal{L}$ is linear, namely a Boolean algebra, then $L_d^T(\mathcal{L}, \Sigma) = L_w^T(\mathcal{L}, \Sigma) = \Sigma_1^0 \cup \Pi_1^0$.

6 Conclusion

Since Turing machine based on quantum logic can solve undecidable problem, namely, it has some super-Turing computational power. In this paper, we continue to discuss the computational power of Turing machine based on quantum logic. It is proved that their language level is between $\Sigma_1^0 \cup \Pi_1^0$ and Π_2^0 from the point of arithmetical hierarchy view. Obviously, it surpasses the computational power of classical fuzzy Turing machine and classical Turing machine.

References

[1] Andris Ambainis and Rusins Freivaldsy, 1-way quantum finite automata: strengths, weaknesses and generalization, In proceedings of the 39th Annual symposium on foundations of computer science, (1998) 332-341.

[2] Garrett Birkhoff, John Von Neumann, The logic of quantum mechanics, Annals of Mathematics, 379 (1936) 823-843.

[3] Chen Chung Chang, Algebraic analysis of many valued logics, Transactions of the American Mathematical Society, 88 (1958) 467-490.

[4] M. Dalla Chiara, Roberto Giuntini, *Richard Greechie, Reasoning in Quantum Theory—Sharp and Unsharp Quantum Logics*, Kluwer: Academic Publishers, 2004.

[5] David Deutsch, Quantum theory, the church-Turing principle and the universal quantum computer. In Proceedings of the Royal Society of London A400,1985, 97-117.

[6] Anatolij Dvurecenskij, Sylvia Pulmannova, *New Trends in Quantum Structures*, Kluwer, Dordrecht; Ister Science, Bratislava, 2000.

[7] Richard P. Feynman. Simulating physics with computers, International Journal of Theoretical Physics, 21 (1982) 467-488.

[8] David. J. Foulis, Mary Katherine Bennett, Effect algebras and unsharp quantum logics, Foundations of Physics, 24 (1994) 1331-1352.

[9] Roberto Giuntini, Quantum MV algebras, Studia logica, 56 (1996) 393-417.

[10] Roberto Giuntini, Quantum MV-Algebras and Commutativity, International Journal of Theoretical Physics, 37 (1998) 65-74.

[11] Stanley Gudder, Total extensions of effect algebras, Foundations of Physics Letters, 8 (1995) 243-252.

[12] John E Hopcroft, Jeffrey D Ullman , *Introduction to automata theory, languages, and computation*, Addision-Wesley, Reading, MA, 1979.

[13] Kodi Husimi, Studies on the foundation of quantum mechanics. Proceedings of the Physico-Mathematical Society of Japan, 19 (1937) 766-789.

[14] Yongming Li, Ping Li, Turing machine based on quantum logic. Chinese Journal of Computers, Submitted.

[15] Attila Kondacs, John Watrous, On the power of quantum finite state automata, In Proceedings of the 38th Annual symposium on foundations of computer science, Miami, Florida, 1997, 66-75.

[16] Ruqian Lu, Hong Zheng, Lattices of quantum automata, International journal of theoretical physics,42 (2003) 1435-1459.

[17] Gunther Ludwig, *Foundations of quantum mechanics*, Vol.1, Springer, Berlin, 1983.

[18] Cristopher Moore, James P. Crutchfield. Quantum automata and quantum grammars. Theoretical Computer Science, 237 (2000) 275-306.

[19] Daowen Qiu, Automata and grammars theory based on quantum logic, Journal of Software, 1 (2003) 23-27.

[20] Daowen Qiu, Automata theory based on quantum logic: Some characterizations, Information and Computation, 190 (2004) 179-195.

[21] Zdenka Rieanová, Generalization of blocks for D-Lattices and lattice-ordered effect algebras, International Journal of Theoretical Physics, 39 (2000) 231-237.

[22] Yun Shang, Xian Lu, Ruqian Lu, Automata theory based on unsharp quantum logic, Mathematical Structures in Computer Science, 19 (2009) 737-756.

[23] Yun Shang, Xian Lu, Ruqian Lu, A theory of computation based on unsharp quantum logic: finite state automata and pushdown automata (2012),doi:10.1016/j.tcs2012.02.018.

[24] Yun Shang, Xian Lu, Ruqian Lu, Turing machine based on unsharp quantum logic, Proceeding of the 8th QPL 2011: quantum physics and logic, 255-262, October 27-29, Nijmegen, NL.

[25] R. Irving Soare, *Recursively Enumerable Sets and Degrees: A Study of Computable Functions and Computably Generated Sets*, Springer, 2004.

[26] Jiri Wiedermann, Fuzzy Turing machines revised. Comput. Artificial Intelligence 21 (2002) 1-13.

[27] Jiri Wiedermann, Characterizing the super-Turing computing power and efficiency of classical fuzzy Turing machines. Theoretical Computer Science 317 (2004) 61-69.

[28] Tomohiro Yamasaki, Hirotada Kobayashi, Hiroshi Imai, Two-way quantum one counter automata,ArXiv: cs. CC/0110005v1 2 Oct 2001.

[29] Mingsheng Ying, Automata theory based on quantum logic (I), International Journal of Theoretical Physics, 39 (2000) 985-995.

[30] Mingsheng Ying, Automata theory based on quantum logic (II), International Journal of Theoretical Physics, 39 (2000) 2545-2557.

[31] Mingsheng Ying, A theory of computation based on quantum logic (I), Theoretical Computer Science, 344 (2005) 134-207.

[32] Mingsheng Ying, Quantum logic and automata theory, Dov Gabbay, Daniel Lehmanm and Kurt Engesser eds, Handbook of quantum structures and quantum logic, North-Holland(Elservier), 2007.