Theoretical Computer Science 598 (2015) 2–14

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Computing power of Turing machines in the framework of unsharp quantum logic [☆]

Yun Shang [a,b,*], Xian Lu [c], Ruqian Lu [a,b,d]

[a] Institute of Mathematics, AMSS, CAS, Beijing 100190, PR China
[b] NCMIS, AMSS, CAS, Beijing 100190, PR China
[c] State Key Laboratory of Low-Dimensional Quantum Physics and Department of Physics, Tsinghua University, PR China
[d] CAS Key Lab of IIP, Institute of Computing Technology, CAS, PR China

ARTICLE INFO

Article history:
Received 20 December 2012
Received in revised form 21 November 2014
Accepted 17 December 2014
Available online 30 December 2014
Communicated by M. Hirvensalo

Keywords:
Turing machines
Unsharp quantum logic
Computational power

ABSTRACT

We present recursion theoretical characterizations of the computational power of Turing machines in the framework of unsharp quantum logic. For unsharp quantum logic, let a lattice ordered quantum multiple valued (QMV) algebra be its truth value lattice. When lattice ordered QMV algebras satisfy a locally finite condition (that is, every non-zero element has a finite order) and its meet operation, $\wedge$, is computable, we prove that $\Sigma_1^0 \cup \Pi_1^0 \subseteq L_d^T(\mathcal{E}, \Sigma)$(or $L_w^T(\mathcal{E}, \Sigma)) \subseteq \Pi_2^0$, where $L_d^T(\mathcal{E}, \Sigma)$ (respectively $L_w^T(\mathcal{E}, \Sigma)$) denotes the class of languages accepted by these Turing machines in the depth-first method (respectively the width-first method). For sharp quantum logic, similar results are obtained for a general orthomodular truth value lattice.

1. Introduction

Classical computational theory involves fundamental computational problems such as computability, computational complexity, and computational apparatuses and models. Among these, the computational apparatuses and models are the most important, and include significant concepts such as recursive functions, lambda calculus, finite state automata, pushdown automata, and Turing machines. These concepts form the foundation of computational theory, which is a mature branch of the computational sciences. However, a new concept, quantum computing, was proposed by Feynman [8] in the 1980s. This concept required a new theory of computation, the theory of quantum computation, in which the theory of quantum automata has been one of the most interesting topics.

The study of quantum automata has been technically divided in two directions. Because of the well-known probabilistic character of quantum mechanics, the first direction of quantum automata research is based on the probabilistic representation of quantum information. Deutsch [6] proposed a model of quantum Turing machines, and Moore and Crutchfield [21] introduced the first model of quantum finite state automata. Moore and Crutchfield's quantum automata model is known as the measure-once model because the measurement is performed only when the computation is finished. Kondacs and Watrous [16] proposed a different type of quantum finite state automata, the measure-many model, in which measurements are performed whenever a letter is read. Other models of quantum automata include the one-way (or two-way) one-counter

☆ This work was supported by NSFC projects under Grant Nos. 61472412 and 61073023, National Center for Mathematics and Interdisciplinary Sciences, CAS.

* Corresponding author at: Institute of Mathematics, AMSS, CAS, Beijing 100190, PR China.
E-mail addresses: shangyun602@163.com (Y. Shang), luxian@amss.ac.cn (X. Lu), rqlu@math.ac.cn (R. Lu).

http://dx.doi.org/10.1016/j.tcs.2014.12.015
0304-3975/

quantum automata model [1,36]. In general, all these models are quantum generalizations of probabilistic automata. For more details, please refer to the review [28].

Other quantum automata researches have been conducted at a more abstract level. In the 1930s, Birkhoff and von Neumann [3] proposed quantum logic based on the observation that basic propositions about quantum systems can be properly described by closed subspaces of its state Hilbert space, and the closed subspaces of a Hilbert space form an orthomodular lattice. Since the algebraic model of quantum logic consists of orthomodular lattices, it is possible to take an orthomodular lattice, rather than a Boolean algebra, as the truth value domain of the acceptance property in a quantum automaton. Whereas 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 a degree ($0 \leq$ acceptance degree ≤ 1) that is represented by a truth value element in an orthomodular lattice, where 0 and 1 are the smallest and largest elements, respectively, of the lattice. Ying first introduced the concept of finite state automata and pushdown automata based on quantum logic in a series of papers [37–40]. In particular, Ying [38–40] found that some properties of classical automata are equivalent to the distributivity of their underlying logic, and therefore are not valid in quantum finite state automata. The validity of these properties for classical automata comes from the distributive law in Boolean logic that is a basis for classical automata theory. However, these properties are absent in quantum automata because the distributive law is absent in an orthomodular lattice. The theory of automata based on quantum logic has been further developed in a series of papers [18,24–28]. Recently, lattice automata have been pursued by Kupferman et al. and many applications are described [15]. A theory of Turing machines based on quantum logic (or orthomodular lattice-valued Turing machines) was developed by Li et al. [17].

One of the most important supporting pillars of (sharp) quantum logic based on an orthomodular lattice is the projection valued (PV) measurement. The PV measurement corresponds to sharp observables, which is why Birkhoff–von Neumann quantum logic is called sharp quantum logic. Because PV measurements are generalized to positive operator valued (POV) measurements, quantum logic becomes unsharp quantum logic that embodies unsharp observables [20]. This means that quantum events do not satisfy the non-contradiction law of unsharp quantum logic, but they do satisfy the non-contradiction law of sharp quantum logic. Many algebraic structures have been proposed to represent quantum events in unsharp quantum logic. In 1994, Foulis introduced effect algebra, which is the main algebraic model for unsharp quantum logic [9]. Multiple valued (MV) algebras, the algebraic model of MV logic, play an analogous role to that of Boolean algebras in sharp quantum logic [4,7]. MV algebras are the blocks of lattice-ordered-effect algebras [29]. Quantum MV (QMV) algebras are another important unsharp quantum structure [10]. They are a non-lattice theoretic generalization of MV algebras, as well as a non-idempotent generalization of an orthomodular lattice. The center of a QMV algebra is an MV algebra [11]. The algebraic models of unsharp quantum logic considered in this paper are the extended lattice-ordered-effect algebras and the lattice ordered QMV algebras [32].

Because unsharp quantum logic is more general than sharp quantum logic, Shang et al. [30,31] proposed a theory of finite state automata and pushdown automata based on unsharp quantum logic. They found that certain 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. Turing machines are at the core of computational theory, and Shang et al. [32] continue this line of research by studying Turing machines based on unsharp quantum logic. Turing machines in the framework of unsharp quantum logic are different from the quantum Turing machines proposed by Deutsch [6]. Deutsch's notion of a quantum Turing machine was generalized by Perdrix [23] to the observable quantum Turing machine. Perdrix and Jorrand [22] introduced classically controlled Turing machines and Bernstein [2] presented universal quantum Turing machines. Different quantum Turing machines can recognize different languages, but what about the relationship between them? It is a very interesting branch and worthy of study.

One of the most interesting results achieved by Shang et al. [32] is that deterministic Turing machines based on quantum logic are not equivalent to non-deterministic Turing machines based on quantum logic. That is, non-deterministic Turing machines can recognize the union of recursive enumerable (r.e.) languages and co-r.e. languages. Since Wiedermann [35] proved that classical fuzzy Turing machines also possess super-Turing computational powers, it has been important to clarify the relationship between the computational powers of Wiedermann's fuzzy Turing machines and Turing machines based on quantum logic [32].

We partially solve the problem outlined above. From the arithmetical hierarchy viewpoint, we find that the language class of Turing machines based on quantum logic is between the first and the second level of the arithmetical hierarchy; the lower boundary of language recognized by Turing machines based on quantum logic is class $\Sigma_1^0 \cup \Pi_1^0$ (the union of r.e. language (class Σ_1^0) and co-r.e. language (class Π_1^0)) and the upper boundary is class Π_2^0 (where a Π_2^0 formula begins with universal quantifiers followed by existential quantifiers). However, the language recognized by classical Turing machines is class Σ_1^0 and language accepted by classical fuzzy Turing machine is class $\Sigma_1^0 \cup \Pi_1^0$ [34,35]. Hence, Turing machines based on quantum logic are more powerful than Wiedermann's fuzzy Turing machines and classical Turing machines. The power of computing models depends heavily on the underlying logic.

In Section 3, basic knowledge regarding extended lattice-ordered-effect algebras and lattice ordered QMV algebras is revisited. We use the symbol $\mathcal{E}$ to denote these two algebras, and some basic details about $\mathcal{E}$-valued Turing machines are presented. In Section 4, we discuss some variants of $\mathcal{E}$-valued Turing machines and the relationship among these variants. In particular, we prove that $\mathcal{E}$-valued non-deterministic Turing machines ($\mathcal{E}$NTMs) are not equivalent to $\mathcal{E}$-valued deterministic Turing machines ($\mathcal{E}$DTMs) and $\mathcal{E}$NTMs have stronger power than $\mathcal{E}$DTMs. In Section 5, we prove that the language of $\mathcal{E}$NTMs