１テープ線形時間量子チューリング機械 (予備的結果報告)

林政弘,[†] 只木孝太郎,[‡] 山上智幸[†]

[†]*School of Information Technology and Engineering, University of Ottawa, 150 Louis Pasteur Ottawa, Ontario Canada, K1N 6N5*

[‡]科学技術振興事業団 創造科学技術推進事業 今井量子計算機構プロジェクト
〒113-0033 東京都文京区本郷 5-28-3 第二本郷ホワイトビル 201

概要

１テープ線形時間チューリング機械は極めて限定的な計算能力しか持たない。そのようなチューリング機械では、テープ上の入力文字列をコピーすることすら出来ないのである。1965 年、Hennie は、計算能力において１テープ線形時間チューリング機械は有限オートマトンと等価であることを示した。本研究で、我々は様々なタイプの１テープ線形時間チューリング機械を考察する。はじめに、古典的チューリング機械について、Hennie の結果を一般化し、１テープ線形時間の非決定論的、可逆的、そして確率的な各チューリング機械が、正則言語しか受理できないことを示す。そして、Bernstein と Vazirani による量子チューリング機械のモデルに厳格に従いながら、あるタイプの１テープ線形時間量子チューリング機械が、非正則言語を受理できることを示す。

One-Tape Linear-Time Quantum Turing Machine (Preliminary Report)

Jack C. H. Lin,[†] Kohtaro Tadaki,[‡] and Tomoyuki Yamakami[†]

[†]*School of Information Technology and Engineering, University of Ottawa, 150 Louis Pasteur Ottawa, Ontario Canada, K1N 6N5*

[‡]*Quantum Computation and Information Project, ERATO, Japan Science and Technology Corporation, Daini Hongo White Bldg. 201, 5-28-3, Hongo, Bunkyo, Tokyo 113-0033, Japan*

Abstract

One-tape linear-time Turing machines only have very low computational power. In 1965 Hennie showed that one-tape linear-time deterministic Turing machines are computationally equal to deterministic finite automata. In this paper, we consider several types of one-tape linear-time Turing machines. By generalizing Hennie's method, it is shown that several types of classical one-tape linear-time Turing machine, i.e., nondeterministic one, reversible one, and probabilistic one, are all computationally equal to a finite automaton. In what follows, we show that a certain type of one-tape linear-time quantum Turing machine can recognize a non-regular language.

1 Introduction

One-tape linear-time Turing machines only have very low computational power. For example, such Turing machines cannot even make a copy of an input string onto another region of the tape since it takes n^2 steps to make a copy of a string of the length n. In 1965 Hennie [8] showed using crossing sequence argument that one-tape linear-time deterministic Turing machines are computationally equal to deterministic finite automata. Kobayashi [11] then generalized this result to one-tape $o(n \log n)$-time deterministic Turing machines, where n is the length of the input. We consider the computational power of several types of one-tape linear-time Turing machines, i.e., nondeterministic one, reversible one, probabilistic one, and quantum one. Apart from one-tape linear-time quantum Turing machines, we proved that the above types of classical one-tape linear-time Turing machine are all computationally equal to a finite automaton. Thus, such Turing machines can only recognize a regular language.

In this paper, we first generalize Hennie and Kobayashi's crossing sequence argument to one-tape linear-time nondeterministic Turing machines to show that such Turing machines can only recognize a regular language. Using the method of a reversible simulation of a deterministic finite automaton given by Kondacs and Watrous [12], we then see that one-tape linear-time reversible Turing machines can recognize any regular language. By generalizing crossing sequence argument further, we can show that one-tape linear-time probabilistic Turing machines can only recognize a regular language with bounded error probability.

We adopt Bernstein-Vazirani's model of quantum Turing machine in order to study the property of one-tape linear-time quantum Turing machines. Since every reversible Turing machine is also a quantum Turing machine, any complexity class which is defined by one-tape linear-time quantum Turing machines includes the set of regular languages. Using the method for 2qcfa to recognize $L_{ab} = \{a^n b^n | n \geq 0\}$ given by Ambainis and Watrous [2], it is shown that the set of regular languages is properly included in **NQLIN**, which is a one-tape linear-time analogue of **NQP**. However, it is still an open problem whether or not a one-tape linear-time quantum Turing machine can recognize a non-regular language with bounded error probability.

2 One-Tape Nondeterministic Turing Machines

Let $\mathbb{N}$ be the set of natural numbers (i.e., non-negative integers) and $\mathbb{N}^+$ the set of positive integers. We denote the set of regular languages by **REG**.

Definition 2.1 (one-tape $T(n)$-time NTM). *Let $T\colon \mathbb{N} \to \mathbb{N}$. A one-tape $T(n)$-time nondeterministic Turing machine (abbreviated $T(n)$-1NTM) is a nondeterministic Turing machine M such that*

(1) M only has one tape of cells which has a left end and infinite cells to the right, and

(2) for each input x, the length of every computation path of M on input x is at most $T(|x|)$.

We say a $T(n)$-1NTM M recognizes a language L if for every x, $x \in L$ if and only if there exists an accepting computation path of M on input x.

Definition 2.2. *Let* $T\colon \mathbb{N} \to \mathbb{N}$*. The class* 1-$\mathbf{NTime}(T(n))$ *is defined by*

$$1\text{-}\mathbf{NTime}(T(n)) \equiv \{\, L \mid \exists\, M : T(n)\text{-}1NTM\ \ M \ \textit{recognizes}\ L \,\}. \tag{1}$$

Then the class 1-**NLIN** *is defined by*

$$1\text{-}\mathbf{NLIN} \equiv \bigcup_{c=1}^{\infty} 1\text{-}\mathbf{NTime}(cn + c). \tag{2}$$

Definition 2.3 (crossing sequence). *Let M be a $T(n)$-1NTM. Each pair of adjacent cells on the tape of M is separated by an intercell boundary. In a computation path s of M, consider an intercell boundary b and the sequence of states of M at the steps when the head crosses b, first from left to right, and then alternatingly in both directions. This orderd sequence of states is called the crossing sequence at the intercell boundary b in the computation path s of M.*

Theorem 2.4 and Theorem 2.7 are generalizations of one given in [11] which deals with the deterministic case.

Theorem 2.4. *Let $T\colon \mathbb{N} \to \mathbb{N}$, and let M be a $T(n)$-1NTM. Suppose that $T(n) = o(n \log n)$. Then there exists $c \in \mathbb{N}$ such that, for every x, the length of crossing sequence of every intercell boundary in every computation path of M on input x is at most c.*

For each language L over an alphabet Σ. Myhill-Nerode equivalence relation R_L on Σ^* is defined by $xR_Ly \iff \forall\, z \in \Sigma^* \,(xz \in L \iff yz \in L)$.

The following lemma is an immediate result of Myhill-Nerode theorem which states that the number of equivalence classes of R_L is finite if and only if L is regular.

Lemma 2.5. *Let L be a language over an alphabet Σ. Suppose that there exists an euivalence relation E on Σ^* such that (i) the number of equivalence classes of E is finite and (ii) $xEy \Longrightarrow xR_Ly$. Then L is regular.*

The following theorem is a generalization of one given in [8] which deals with the deterministic case. Its proof uses Lemma 2.5.

Theorem 2.6. *Let L be a language over an alphabet Σ, and let M be a $T(n)$-1NTM which recognizes L. If there exists $c \in \mathbb{N}$ such that, for every $x \in L$, the length of crossing sequence of every intercell boundary in every computation path of M on input x is at most c, then L is regular.*

Proof. Let S be the set of sequences of states of M whose lengths are at most c. For each $x \in \Sigma^*$ and each $v \in S$, we say that x supports v if there exists $z \in \Sigma^*$ such that $xz \in L$ and v is the crossing sequence of the intercell boundary between x and z in some accepting computation path of M on input xz. For each $x \in \Sigma^*$, let $\mathrm{Sup}(x) = \{v \in S \mid x \text{ supports } v\}$. We define an equivalence relation E on Σ^* by $xEy \iff \mathrm{Sup}(x) = \mathrm{Sup}(y)$. Then, using crossing sequence argument, we see that the conditions (i) and (ii) in Lemma 2.5 hold. Hence, L is regular. □

Theorem 2.7. *Let $T\colon \mathbb{N}\to\mathbb{N}$. Suppose that $n \le T(n)$ and $T(n) = o(n\log n)$. Then*

$$\mathbf{REG} = 1\text{-}\mathbf{NTime}(T(n)) = \text{co-}1\text{-}\mathbf{NTime}(T(n)). \tag{3}$$

Proof. $\mathbf{REG} \subseteq 1\text{-}\mathbf{NTime}(T(n))$ is obvious. It follows from Theorem 2.4 and Theorem 2.6 that $1\text{-}\mathbf{NTime}(T(n)) \subseteq \mathbf{REG}$. Thus, we have $1\text{-}\mathbf{NTime}(T(n)) = \mathbf{REG}$. By noting $\text{co-}1\text{-}\mathbf{NTime}(T(n)) = \text{co-}\mathbf{REG} = \mathbf{REG}$, the result is obtained. □

Remark 2.8. *Let L_{pal} be the set of palindromes, i.e., $L_{pal} \equiv \{x \in \{0,1\}^* \mid x = x^R\}$ where x^R is the reverse of x. We can construct a one-tape $O(n\log n)$-time NTM which recognizes $\overline{L_{pal}}$, so $\overline{L_{pal}} \in 1\text{-}\mathbf{NTime}(n\log n)$. On the other hand, it is obvious that $L_{pal} \notin \mathbf{REG}$, so $\overline{L_{pal}} \notin \mathbf{REG}$. Thus, by Theorem 2.7, if $n \le T(n)$ and $T(n) = o(n\log n)$, then*

$$\mathbf{REG} = 1\text{-}\mathbf{NTime}(T(n)) \subsetneq 1\text{-}\mathbf{NTime}(n\log n) \tag{4}$$

and therefore

$$\mathbf{REG} = \text{co-}1\text{-}\mathbf{NTime}(T(n)) \subsetneq \text{co-}1\text{-}\mathbf{NTime}(n\log n). \tag{5}$$

Moreover, it is shown using crossing sequence argument that $L_{pal} \notin 1\text{-}\mathbf{NTime}(n\log n)$. Thus,

$$1\text{-}\mathbf{NTime}(n\log n) \nsubseteq \text{co-}1\text{-}\mathbf{NTime}(n\log n) \tag{6}$$

and therefore

$$\text{co-}1\text{-}\mathbf{NTime}(n\log n) \nsubseteq 1\text{-}\mathbf{NTime}(n\log n). \tag{7}$$

Definition 2.9 (one-tape $T(n)$-time DTM). *Let $T\colon \mathbb{N}\to\mathbb{N}$. A one-tape $T(n)$-time deterministic Turing machine (abbreviated $T(n)$-1DTM) is a $T(n)$-1NTM which has at most one nondeterministic choice at each step.*

Definition 2.10. *Let $T\colon \mathbb{N}\to\mathbb{N}$. The class $1\text{-}\mathbf{DTime}(T(n))$ is defined by*

$$1\text{-}\mathbf{DTime}(T(n)) \equiv \{\, L \mid \exists M : T(n)\text{-1DTM } M \text{ recognizes } L \,\}. \tag{8}$$

Then the class $1\text{-}\mathbf{DLIN}$ is defined by

$$1\text{-}\mathbf{DLIN} \equiv \bigcup_{c=1}^{\infty} 1\text{-}\mathbf{DTime}(cn + c). \tag{9}$$

Definition 2.11 (one-tape $T(n)$-time reversible DTM). *Let $T\colon \mathbb{N}\to\mathbb{N}$. A one-tape $T(n)$-time reversible deterministic Turing machine (abbreviated $T(n)$-1revDTM) is a $T(n)$-1DTM for which each configuration has at most one predecessor configuration.*

Definition 2.12. *Let $T\colon \mathbb{N}\to\mathbb{N}$. The class $1\text{-rev}\mathbf{DTime}(T(n))$ is defined by*

$$1\text{-rev}\mathbf{DTime}(T(n)) \equiv \{\, L \mid \exists M : T(n)\text{-1revDTM } M \text{ recognizes } L \,\}. \tag{10}$$

Then the class $1\text{-rev}\mathbf{DLIN}$ is defined by

$$1\text{-rev}\mathbf{DLIN} \equiv \bigcup_{c=1}^{\infty} 1\text{-rev}\mathbf{DTime}(cn + c). \tag{11}$$

Theorem 2.13. **REG** $\subseteq$ 1-rev**DLIN**.

Proof. The result is obtained using the method used in the simulation of any deterministic finite automaton by some two-way reversible finite automaton given in [12]. □

Theorem 2.14. **REG** $=$ 1-rev**DLIN** $=$ 1-**DLIN** $=$ 1-**NLIN** $=$ co-1-**NLIN**.

Proof. It follows from Theorem 2.13 that **REG** $\subseteq$ 1-rev**DLIN** $\subseteq$ 1-**DLIN** $\subseteq$ 1-**NLIN**. We have, by Theorem 2.7, 1-**NLIN** $=$ **REG**; therefore co-1-**NLIN** $=$ **REG**. Thus, the result follows. □

3 One-Tape Probabilistic Turing Machines

Definition 3.1 (one-tape $T(n)$-time PTM). *Let $T\colon \mathbb{N} \to \mathbb{N}$. A one-tape $T(n)$-time probabilistic Turing machine (abbreviated $T(n)$-1PTM) is a $T(n)$-1NTM which has exactly two nondeterministic choices at each step in a non-final configuration.*

In the above definition, we do not require that for every x, all of computations of a $T(n)$-1PTM on input x halt after the same number of steps.

Definition 3.2 (accepting probability). *Let $T\colon \mathbb{N} \to \mathbb{N}$, and let M be a $T(n)$-1PTM. $AP(M,x)$ is defined as the set of all accepting computation paths of M on input x. The length of a computation path s is denoted by $l(s)$. Here the length of a computation path is the number of applications of the transition function along the path. The accepting probability of M on input x is denoted by $P_a^M(x)$ and is defined as*

$$P_a^M(x) \equiv \sum_{s \in AP(M,x)} \left(\frac{1}{2}\right)^{l(s)}. \tag{12}$$

Let L be a language, and let $0 \le \varepsilon < 1/2$. We say that M recognizes L with error probability ε if

(1) $x \in L \Longrightarrow P_a^M(x) \ge 1 - \varepsilon$, and

(2) $x \notin L \Longrightarrow P_a^M(x) \le \varepsilon$.

Definition 3.3. *Let $T\colon \mathbb{N} \to \mathbb{N}$. The class 1-**BPTime**$(T(n))$ is defined as the set*

$$\{\, L \mid \exists\, M : T(n)\text{-1PTM } \exists\, \varepsilon \in [0, 1/2) \ \ M \text{ recognizes } L \text{ with error probability } \varepsilon \,\}. \tag{13}$$

*Then the class 1-**BPLIN** is defined by*

$$1\text{-}\mathbf{BPLIN} \equiv \bigcup_{c=1}^{\infty} 1\text{-}\mathbf{BPTime}(cn + c). \tag{14}$$

By modifying the proof of Theorem 2.6, "$T(n)$-1NTM" in Theorem 2.6 can be replaced by "$T(n)$-1PTM." Thus, using Theorem 2.4, we can prove the following theorem in stead of Theorem 2.7.

Theorem 3.4. *Let $T: \mathbb{N} \to \mathbb{N}$. Suppose that $n+1 \leq T(n)$ and $T(n) = o(n \log n)$. Then*

$$\mathbf{REG} = 1\text{-}\mathbf{BPTime}(T(n)). \tag{15}$$

We have the following corollary from Theorem 3.4.

Corollary 3.5. *Let $T: \mathbb{N} \to \mathbb{N}$, and let M be a $T(n)$-1PTM. Suppose that L is a non-regular language and M recognizes L with error probability ε for some $\varepsilon \in [0, 1/2)$. Then there exists $c > 0$ such that for infinitely many n, $T(n) \geq cn \log n$.*

Theorem 3.6. $\mathbf{REG} = 1\text{-}\mathbf{BPLIN}$.

Proof. The result follows immediately from Theorem 3.4. □

4 One-Tape Quantum Turing Machines

We adopt Bernstein-Vazirani's model of quantum Turing machine [3]. This model is already a one-tape quantum Turing machine (with multitrack), which we abbreviate to QTM. See [1] and [3] for the definition and the property of QTM.

Definition 4.1. *The class* 1-**BQLIN** *is defined as the set of languages L such that there exist a stationary QTM M, a $c \in \mathbb{N}^+$, and an $\varepsilon > 0$ which have the following properties:*

(1) On every input x, M halts in time $c|x| + c$.

(2) $x \in L \Longrightarrow M$ accepts input x with probability greater than $1/2 + \varepsilon$.

(3) $x \notin L \Longrightarrow M$ accepts input x with probability less than $1/2 - \varepsilon$.

Theorem 4.2. $\mathbf{REG} \subseteq 1\text{-}\mathbf{BQLIN}$.

Proof. From $\mathbf{REG} = 1\text{-rev}\mathbf{DLIN}$ and the fact that every reversible deterministic Turing machine is a well-formed QTM, the result follows. □

Remark 4.3. *It is an open problem whether or not* $\mathbf{REG} = 1\text{-}\mathbf{BQLIN}$ *holds.*

Definition 4.4. *The class* 1-**NQLIN** *is defined as the set of languages L such that there exist a stationary QTM M and a $c \in \mathbb{N}^+$ which have the following properties:*

(1) On every input x, M halts in time $c|x| + c$.

(2) $x \in L \Longleftrightarrow M$ accepts input x with positive probability.

Theorem 4.5. $\mathbf{REG} \subsetneq 1\text{-}\mathbf{NQLIN}$.

Proof. Since $\mathbf{REG} = 1\text{-rev}\mathbf{DLIN}$, we see that $\mathbf{REG} \subseteq 1\text{-}\mathbf{NQLIN}$. Let $L_{ab} = \{a^n b^n | n \geq 0\}$. Then $\overline{L_{ab}} \notin \mathbf{REG}$. Using in essence the method for 2qcfa to recognize L_{ab} given in [2], we can show that $\overline{L_{ab}} \in 1\text{-}\mathbf{NQLIN}$. This completes the proof. □

Acknowledgments

The second author would like to thank Masahiro Hachimori for a valuable suggestion, Hiroshi Imai and Keiji Matsumoto for support.

References

[1] L.M. Adleman, J. DeMarrais, and M.A. Huang, Quantum computability, *SIAM J. Comput.*, **26**, pp.1524–1540, 1997.

[2] A. Ambainis and J. Watrous, Two-way automata with quantum and classical states, LANL cs.CC/9911009, 1999.

[3] E. Bernstein and U. Vazirani, Quantum complexity theory, *SIAM J. Comput.*, **26**, pp.1411–1473, 1997.

[4] A. Condon, Bounded Error Probabilistic Finite State Automata, in *Handbook of Randomized Computing* (edited by Sanguthevar Rajasekaran, Panos M. Pardalos, John H. Reif, and Jose D.P. Rolim), Kluwer, 2001.

[5] C. Dwork and L. Stockmeyer, A time complexity gap for 2-way probabilistic finite state automata, *SIAM J. Comput.*, **19**, pp.1011–1023, 1990.

[6] C. Dwork and L. Stockmeyer, Finite state verifiers I: the power of interaction, *J. ACM*, **39** (1992), pp.800–828.

[7] R. Freivalds, Probabilistic two-way machines, in *Proc. the International Symposium on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, Vol.188, pp.33–45, Springer-Verlag, 1981.

[8] F. C. Hennie, One-tape, off-line Turing machine computations, *Inform. and Control*, **8** (1965), pp.553–578.

[9] J. Kaneps, Stochasticity of the languages recognizable by 2-way finite probabilistic automata, *Diskret. Mat.*, **1**, pp.63–77.

[10] J. Kaņeps and R. Freivalds, Running time to recognize nonregular languages by 2-way probabilistic automata, in *Proc. 18th International Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science, Vol.510, pp.174–185, 1991.

[11] K. Kobayashi, On the structure of one-tape nondeterministic Turing machine time hierarchy, *Theor. Comput. Sci.*, **40** (1985), pp.175–193.

[12] A. Kondacs and J. Watrous, On the power of quantum finite state automata, in *Proc. 38th Annual Symposium on Foundations of Computer Science*, pp.66–75, 1997.

[13] M. Li and P. M. B.Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications, 2nd ed.* (Springer, New York, 1997).

[14] P. Michel, An NP-complete language accepted in linear time by a one-tape Turing machine, *Theor. Comput. Sci.*, **85** (1991), pp.205–212.

[15] M. O. Rabin, Probabilistic automata, *Inform. and Control*, **6** (1963), pp.230–245.