Advantages of Ultrametric Counter Automata*

Valdis Ādamsons, Kārlis Jēriņš, Rihards Krišlauks, Marta Lapiņa,
Andris Pakulis, and Rūsiņš Freivalds

Faculty of Computing, University of Latvia, Raiņa bulvāris 19, Riga, LV-1586, Latvia
Institute of Mathematics and Computer Science, University of Latvia,
Raiņa bulvāris 29, Riga, LV-1459, Latvia
valdisxp1@gmail.com

Abstract. Ultrametric algorithms are similar to probabilistic algorithms but they describe the degree of indeterminism by p-adic numbers instead of real numbers. No wonder that only very few examples of advantages for ultrametric algorithms over probabilistic ones have been published up to now, and all they are slightly artificial. This paper considers ultrametric and probabilistic one-counter automata with two one-way input tapes. A language is found which is recognizable by ultrametric but not by probabilistic automata of this type.

1 Introduction

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. A query algorithm is an algorithm for computing $f(x_1, \ldots, x_n)$ that accesses $x_1, \ldots, x_n$ by asking questions about the values of x_i. The complexity of a query algorithm is the maximum number of questions that it asks. The query complexity of a function f is the minimum complexity of a query algorithm correctly computing f. The theory of computation studies various models of computation: deterministic, non-deterministic, and probabilistic and quantum (see [20] on traditional models of computation and [17] on quantum computation).

Deterministic, nondeterministic, probabilistic and quantum algorithms are widely considered in literature (e.g., see survey [4]). We consider a relatively new type of algorithms, namely, ultrametric algorithms introduced in [6]. The definition of ultrametric algorithms rather closely follow the example of the corresponding probabilistic and quantum algorithms.

Our model of the computing device is an automaton with two input tapes and one counter for memory. There is exactly one reading head on each input tape. These heads can move only one direction (or to stay at the same square of the tape). The input words are limited by special markers showing that the word has been completely read. The counter is empty at the beginning of the work.

* The research was supported by Project 271/2012 from the Latvian Council of Science.

2 Ultrametric Algorithms

A new type of indeterministic algorithms called *ultrametric* algorithms was introduced in [6]. An extensive research on ultrametric algorithms of various kinds has been performed by several authors (cf. [2, 7, 16, 22]). So, ultrametric algorithms is a very new concept and their potential still has to be explored. This is the first paper showing a problem where ultrametric algorithms have advantages over quantum algorithms.

Ultrametric algorithms are very similar to probabilistic algorithms but while probabilistic algorithms use *real* numbers r with $0 \leq r \leq 1$ as parameters, ultrametric algorithms use *p-adic* numbers as parameters. The usage of p-adic numbers as *amplitudes* and the ability to perform *measurements* to transform amplitudes into real numbers are inspired by quantum computations and allow for algorithms not possible in classical computations. Slightly simplifying the description of the definitions, one can say that ultrametric algorithms are the same as probabilistic algorithms, only the *interpretation* of the probabilities is *different*.

The choice of p-adic numbers instead of real numbers is not quite arbitrary. Ostrowski [19] proved that any non-trivial absolute value on the rational numbers $\mathbb{Q}$ is equivalent to either the usual real absolute value or a p-adic absolute value. This result shows that using p-adic numbers was not merely one of many possibilities to generalize the definition of deterministic algorithms but rather the only remaining possibility not yet explored.

The notion of p-adic numbers is widely used in science. String theory [21], chemistry [15] and molecular biology [5, 12] have introduced p-adic numbers to describe measures of indeterminism. Indeed, research on indeterminism in nature has a long history. Pascal and Fermat believed that every event of indeterminism can be described by a real number between 0 and 1 called *probability*. Quantum physics introduced a description in terms of complex numbers called *amplitude of probabilities* and later in terms of probabilistic combinations of amplitudes most conveniently described by *density matrices*. Using p-adic numbers to describe indeterminism allows to explore some aspects of indeterminism but, of course, does not exhaust all the aspects of it.

There are many distinct p-adic absolute values corresponding to the many prime numbers p. These absolute values are traditionally called *ultrametric*. Absolute values are needed to consider *distances* among objects. We are used to rational and irrational numbers as measures for distances, and there is a psychological difficulty to imagine that something else can be used instead of rational and irrational numbers, respectively. However, there is an important feature that distinguishes p-adic numbers from real numbers. Real numbers (both rational and irrational) are linearly ordered, while p-adic numbers *cannot* be linearly ordered. This is why *valuations* and *norms* of p-adic numbers are considered.

The situation is similar in Quantum Computation (see [17]). Quantum amplitudes are complex numbers which also cannot be linearly ordered. The counterpart of valuation for quantum algorithms is *measurement* translating a complex

number $a+bi$ into a real number a^2+b^2. Norms of p-adic numbers are rational numbers. We continue with a short description of p-adic numbers.

Advantages of ultrametric algorithms over deterministic one have been proved in [2, 6, 10, 16, 22]. It is much more hard to prove advatages of ultrametric algorithms over probabilistic and nondeterministic ones. The only published examples are in computational learning theory [7].

3 p-Adic Numbers and p-Ultrametric Algorithms

Let p be an arbitrary prime number. A number $a \in \mathbb{N}$ with $0 \leq a \leq p-1$ is called a *p-adic digit*. A *p-adic integer* is by definition a sequence $(a_i)_{i\in\mathbb{N}}$ of p-adic digits. We write this conventionally as $\cdots a_i \cdots a_2 a_1 a_0$, i.e., the a_i are written from left to right.

If n is a natural number, and $n = \overline{a_{k-1}a_{k-2}\cdots a_1 a_0}$ is its p-adic representation, i.e., $n = \sum_{i=0}^{k-1} a_i p^i$, where each a_i is a p-adic digit, then we identify n with the p-adic integer (a_i), where $a_i = 0$ for all $i \geq k$. This means that the natural numbers can be identified with the p-adic integers $(a_i)_{i\in\mathbb{N}}$ for which all but finitely many digits are 0. In particular, the number 0 is the p-adic integer all of whose digits are 0, and 1 is the p-adic integer all of whose digits are 0 except the right-most digit a_0 which is 1.

To obtain p-adic representations of all rational numbers, $\frac{1}{p}$ is represented as $\cdots 00.1$, the number $\frac{1}{p^2}$ as $\cdots 00.01$, and so on. For any p-adic number it is allowed to have infinitely many (!) digits to the left of the "p-adic" point but only a finite number of digits to the right of it.

However, p-adic numbers are not merely a generalization of rational numbers. They are related to the notion of *absolute value* of numbers. If X is a nonempty set, a distance, or metric, on X is a function d from $X \times X$ to the nonnegative real numbers such that for all $(x, y) \in X \times X$ the following conditions are satisfied.

(1) $d(x, y) \geq 0$, and $d(x, y) = 0$ if and only if $x = y$,
(2) $d(x, y) = d(y, x)$,
(3) $d(x, y) \leq d(x, z) + d(z, y)$ for all $z \in X$.

A set X together with a metric d is called a *metric space*. The same set X can give rise to many different metric spaces. If X is a linear space over the real numbers then the *norm* of an element $x \in X$ is its distance from 0, i.e., for all $x, y \in X$ and α any real number we have:

(1) $\|x\| \geq 0$, and $\|x\| = 0$ if and only if $x = 0$,
(2) $\|\alpha \cdot y\| = |\alpha| \cdot \|y\|$,
(3) $\|x + y\| \leq \|x\| + \|y\|$.

Note that every norm induces a metric d, i.e., $d(x, y) = \|x - y\|$. A well-known example is the metric over $\mathbb{Q}$ induced by the ordinary absolute value. However, there are other norms as well. A norm is called *ultrametric* if Requirement (3) can be replaced by the stronger statement: $\|x+y\| \leq \max\{\|x\|, \|y\|\}$. Otherwise, the norm is called *Archimedean*.

Definition 1. *Let $p \in \{2,3,5,7,11,13,\ldots\}$ be any prime number. For any nonzero integer a, let the p-adic* ordinal (*or* valuation) *of a, denoted $ord_p\, a$, be the highest power of p which divides a, i.e., the greatest number $m \in \mathbb{N}$ such that $a \equiv 0 \pmod{p^m}$. For any rational number $x = a/b$ we define $ord_p\, x =_{df} ord_p\, a - ord_p\, b$. Additionally, $ord_p\, x =_{df} \infty$ if and only if $x = 0$.*

For example, let $x = 63/550 = 2^{-1} \cdot 3^2 \cdot 5^{-2} \cdot 7^1 \cdot 11^{-1}$. Thus, we have

$$\begin{aligned} \operatorname{ord}_2 x &= -1 & \operatorname{ord}_7 x &= +1 \\ \operatorname{ord}_3 x &= +2 & \operatorname{ord}_{11} x &= -1 \\ \operatorname{ord}_5 x &= -2 & \operatorname{ord}_p x &= 0 \quad \text{for every prime } p \notin \{2,3,5,7,11\}\ . \end{aligned}$$

Definition 2. *Let $p \in \{2,3,5,7,11,13,\ldots\}$ be any prime number. For any rational number x, we define its p-*norm *as $p^{-ord_p\, x}$, and we set $\|0\|_p =_{df} 0$.*

For example, with $x = 63/550 = 2^{-1}3^2 5^{-2} 7^1 11^{-1}$ we obtain:

$$\begin{aligned} \|x\|_2 &= 2 & \|x\|_7 &= 1/7 \\ \|x\|_3 &= 1/9 & \|x\|_{11} &= 11 \\ \|x\|_5 &= 25 & \|x\|_p &= 1 \quad \text{for every prime } p \notin \{2,3,5,7,11\}\ . \end{aligned}$$

Rational numbers are p-adic integers for all prime numbers p. Since the definitions given above are all we need, we finish our exposition of p-adic numbers here. For a more detailed description of p-adic numbers we refer to [8, 13].

We continue with *ultrametric algorithms*. In the following, p always denotes a prime number. Ultrametric algorithms are described by finite directed acyclic graphs (abbr. DAG), where exactly one node is marked as root. As usual, the root does not have any incoming edge. Furthermore, every node having outdegree zero is said to be a *leaf*. The leaves are the output nodes of the DAG.

Let v be a node in such a graph. Then each outgoing edge is labeled by a p-adic number which we call *amplitude*. We require that the sum of all amplitudes that correspond to v is 1. In order to determine the *total amplitude* along a computation path, we need the following definition.

Definition 3. *The total amplitude of the root is defined to be* 1*. Furthermore, let v be a node at depth d in the DAG, let α be its total amplitude, and let $\beta_1, \beta_2, \cdots, \beta_k$ be the amplitudes corresponding to the outgoing edges $e_1, \ldots, e_k$ of v. Let $v_1, \ldots, v_k$ be the nodes where the edges $e_1, \ldots, e_k$ point to. Then the total amplitude of v_ℓ, $\ell \in \{1, \ldots, k\}$, is defined as follows.*

(1) *If the indegree of v_ℓ is one, then its total amplitude is $\alpha\beta_\ell$.*
(2) *If the indegree of v_ℓ is bigger than one, i.e., if two or more computation paths are joined, say m paths, then let $\alpha, \gamma_2, \ldots, \gamma_m$ be the corresponding total amplitudes of the predecessors of v_ℓ and let $\beta_\ell, \delta_2, \ldots, \delta_m$ be the amplitudes of the incoming edges The total amplitude of the node v_ℓ is then defined to be $\alpha\beta_\ell + \gamma_2\delta_2 + \cdots + \delta_m\gamma_m$.*

Note that the total amplitude is a p-adic integer.

It remains to define what is meant by saying that a p-ultrametric algorithm produces a result with a certain probability. This is specified by performing a so-called *measurement* at the leaves of the corresponding DAG. Here by measurement we mean that we transform the total amplitude β of each leaf to $\|\beta\|_p$. We refer to $\|\beta\|_p$ as the *p-probability* of the corresponding computation path.

Definition 4. *We say that a p-ultrametric algorithm* produces a result m with a probability q *if the sum of the p-probabilities of all leaves which correctly produce the result m is no less than q.*

Comment. Just as in Quantum Computation, there is something counterintuitive in ultrametric algorithms. The notion of probability which is the result of measurement not always correspond to our expectations. It was not easy to accept that L. Grover's query algorithm [9] does not read all the input on any computation path. There is a similar situation in ultrametric algorithms. It is more easy to accept the definition of ultrametric algorithms in the case when there is only one accepting state in the algorithm. The 2-ultrametric algorithm in Theorem 1 has only one accepting state.

4 Ambainis' Function

A. Ambainis exhibited a function f that provides the first superlinear separation between polynomial degree and quantum query complexity [1].

Ambainis' function f of 4 Boolean variables is defined as follows:

$$f(x_1, x_2, x_3, x_4) = x_1 + x_2 + x_3x_4 - x_1x_4 - x_2x_3 - x_1x_2.$$

It is easy to check that for arbitrary 4-tuple (x_1, x_2, x_3, x_4), if $(x_1, x_2, x_3, x_4) \in \{0,1\}^4$ then $f(x_1, x_2, x_3, x_4) \in \{0,1\}$. To explore properties of the Ambainis' function we introduce 6 auxiliary sets of variables.

$$\begin{array}{l|l} S_1 = \{x_1, x_2\} & T_1 = \{x_1\} \\ S_2 = \{x_2, x_3\} & T_2 = \{x_2\} \\ S_3 = \{x_1, x_4\} & T_3 = \{x_3, x_4\} \end{array}$$

By S we denote the class (S_1, S_2, S_3) and by T we denote the class (T_1, T_2, T_3).

By $\alpha(x_1, x_2, x_3, x_4)$ we denote the cardinality of those $S_i = (x_j, x_k)$ such that $x_j = x_k = 1$. By $\beta(x_1, x_2, x_3, x_4)$ we denote the cardinality of those T_i such that it contains at least one element x_j which equals 0.

Table 1.

x_1	x_2	x_3	x_4	$\alpha(x_1,x_2,x_3,x_4)$	$\beta(x_1,x_2,x_3,x_4)$	$f(x_1,x_2,x_3,x_4)$
0	0	0	0	0	3	0
0	0	0	1	0	3	0
0	0	1	0	0	3	0
0	0	1	1	0	2	1
0	1	0	0	0	2	1
0	1	0	1	0	2	1
0	1	1	0	1	2	0
0	1	1	1	1	1	1
1	0	0	0	0	2	1
1	0	0	1	1	2	0
1	0	1	0	0	2	1
1	0	1	1	1	1	1
1	1	0	0	1	1	1
1	1	0	1	2	1	0
1	1	1	0	2	1	0
1	1	1	1	3	0	0

Lemma 1. *For arbitrary 4-tuple* $(x_1,x_2,x_3,x_4) \in \{0,1\}^4$, $f(x_1,x_2,x_3,x_4) = 0$ *iff* $\alpha(x_1,x_2,x_3,x_4) + \beta(x_1,x_2,x_3,x_4)$ *is congruent to 1 modulo 2.*

Proof. Immediately from Table 1. □

5 A Language for Advantages

Definition 5. *Let* X *and* Y *be two finite sets such that* $X \subseteq Y$. *Let* y *be a word in the alphabet* Y. *Projection* $proj_X(y)$ *is the word obtained from* y *by removing all the symbols not in* X.

To define the language L of pairs of words (v,w) where $v \in \{a_1,a_2,a_3,a_4\}^*$, $w \in \{a_1,a_2,a_3,a_4\}^*$ we first define four auxiliary languages L_1, L_2, L_3, L_4.

$$L_1 = \{(v,w) \mid proj_{\{a_1\}}v = proj_{\{a_1\}}w\}$$

$$L_2 = \{(v,w) \mid proj_{\{a_2\}}v = proj_{\{a_2\}}w\}$$

$$L_3 = \{(v,w) \mid proj_{\{a_3\}}v = proj_{\{a_3\}}w\}$$

$$L_4 = \{(v,w) \mid proj_{\{a_4\}}v = proj_{\{a_4\}}w\}$$

By $c_i(v,w)$ where $v \in \{a_1,a_2,a_3,a_4\}^*, w \in \{a_1,a_2,a_3,a_4\}^*$, $i \in \{1,2,3,4\}$ we denote the function

$$c_i(v,w) = \begin{cases} 1 \text{ , if } (v,w) \in L_i \\ 0 \text{ , if } (v,w) \notin L_i. \end{cases}$$

$$L = \{(v,w) \mid f(c_1(v,w), c_2(v,w), c_3(v,w), c_4(v,w)) = 1\}.$$

Lemma 2. *For each $i \in \{1,2,3,4\}$ there exists a deterministic one-counter automaton with two one-way input tapes recognizing the language L_i.*

Proof. Immediate. □

Lemma 3. *For each pair (i,j) such that $i \in \{1,2,3,4\}$ and $j \in \{1,2,3,4\}$, there exists a deterministic one-counter algorithm with two one-way input tapes transforming (v,w) into $(c_i(v,w), c_j(v,w))$.*

Proof. The needed deterministic automaton does not use counter to check whether $proj_{\{a_i\}}v = proj_{\{a_j\}}w\}$ and synchronizes the reading of the input tapes to keep the difference between the number of symbols a_i already read from the two input tapes minimal. At every moment the content of the counter equals the difference between the number of symbols a_j already read from the two input tapes. $c_j(v,w) = 1$ if the counter is empty after reading all the symbols. □

Theorem 1. *There exists a 2-ultrametric automaton with two one-way input tapes recognizing the language L.*

Proof. The desired algorithm branches its computation path into 6 branches at the root. We assign to each starting edge of the computation path the amplitude $\frac{1}{7}$.

The first 3 branches (labeled with numbers $1,2,3$) correspond to exactly one set S_i.

Let S_i consist of elements x_j, x_k. Then the algorithm computes the pair $(c_j(v,w), c_k(v,w))$ as decribed in the proof of Lemma 3. If the two computed values equal 1 then the algorithm goes to the state q_3. If at least one of the computed values equals 0 then the algorithm goes to the state q_4.

The next 3 branches (labeled with numbers $4,5,6$) correspond to exactly one set T_i. Let T_i consist of elements x_j, x_k. Then the algorithm computes the pair $(c_j(v,w), c_k(v,w))$ as described in the proof of Lemma 3. If at least one of the results equals 0 then the algorithm goes to the state q_3. If all the results equal 1 then the algorithm goes to the state q_4.

1 branch (labeled with number 7) asks no query and the algorithm goes to the state q_3.

In result of this computation the amplitude A_3 of the states q_3 has become

$$A_3 = \frac{1}{7}(1 + \alpha(x_1,x_2,x_3,x_4) + \alpha(x_1,x_2,x_3,x_4)),$$

The 2-ultrametric query algorithm performs measurement of the state q_3. The amplitude A_3 is transformed into a rational number $\|A_3\|$. 2-adic notation for the

number 7 is $\ldots 000111$ and 2-adic notation for the number $\frac{1}{7}$ is $\ldots 110110110111$. Hence, for every 2-adic integer γ, $\|\gamma\| = \|\frac{1}{7}\gamma\|$.

By Lemma 1, $\|1 + \alpha(x_1, x_2, x_3, x_4) + \alpha(x_1, x_2, x_3, x_4)\|_2$ equals 1, if $f(x_1, x_2, x_3, x_4) = 0$ and it equals $\frac{1}{2}$, if $f(x_1, x_2, x_3, x_4) = 0$ □

We wish to prove that there is no error bounded probabilistic one-counter automaton with two one-way input tapes recognizing the language L. Had our automaton had two-way input tapes, even deterministic automaton could do the job. The real problem is whether it is possible to combine four deterministic one-counter automata recognizing L_1, L_2, L_3, L_4 in a single probabilistic automaton. Our Theorem 2 below shows that it is "almost possible".

We define the language

$$M = L_1 \cap L_2 \cap L_3 \cap L_4 = \{(v, w) \mid c_1(v, w) \cdot c_2(v, w) \cdot c_3(v, w) \cdot c_4(v, w) = 1\}$$

which seems to be quite similar to L but probabilistic automata can recognize it. Advantages of ultrametric algorithms can be seen comparing the languages L and M.

Theorem 2. *For arbitrary $\epsilon > 0$, there exists a probabilistic one-counter automaton with two one-way input tapes recognizing the language M with probability $1 - \epsilon$.*

We wish to prove that there exists no error bounded probabilistic one-counter automaton with two one-way input tapes recognizing the language L. In the proof we heavily employ the property of Ambainis' function to be non-monotonic. Being non-monotonic is the property that distinguishes Ambainis' function from conjunction used in definition of the language M.

$$\begin{cases} f(1,1,0,1) = 0 \\ f(0,1,0,0) = 1 \\ f(0,1,0,1) = 1 \\ f(0,0,0,0) = 0 \end{cases}$$

We need main notions of the theory of finite Markov chains for this proof. (For more details see the classical book [11].)

A finite Markov chain is a stochastic process which moves through a finite number of states, and for which the probability of entering a certain state depends only on the last state occupied.

We describe a Markov chain as follows: We have a set of states,

$$S = \{s_1, s_2, \ldots, s_k\}.$$

The process starts in one of these states and moves successively from one state to another. Each move is called a step. If the chain is currently in state s_i, then it moves to state s_j at the next step with a probability denoted by p_{ij}, and this probability does not depend upon which states the chain was in before

the current state. The probabilities p_{ij} are called transition probabilities. The process can remain in the state it is in, and this occurs with probability p_{ii}. An initial probability distribution, defined on S, specifies the starting state. Usually this is done by specifying a particular state as the starting state.

In particular, the states are divided into equivalence classes. Two states are in the same equivalence class if they "communicate," i.e. if one can go from either state to the other one. The resulting partial ordering shows us the possible directions in which the process can proceed. The minimal elements of the partial ordering are of particular interest.

Definition 6. *The minimal elements of partial ordering of equivalence classes are called ergodic sets. The remaining elements are called transient sets. The elements of a transient set are called transient states. The elements of an ergodic set are called ergodic (or non-transient) states.*

An ergodic set of states is a set in which every state can be reached from every other state, and which cannot be left once it is entered. A transient set of states is a set in which every state can be reached from every other state, and which can be left.

Since every finite partial ordering must have at least one minimal element, there must be at least one ergodic set for every Markov chain. In particular, if an ergodic set contains only one element, then we have a state which once entered cannot be left. Such a state is called absorbing. However, there need be no transient set. The latter will occur if the entire chain consists of a single ergodic set, or if there are several ergodic sets which do not communicate with others.

If a chain has more than one ergodic set, then there is absolutely no interaction between these sets. Hence we have two or more unrelated Markov chains lumped together. A chain consisting of a single ergodic set is called an ergodic chain. A particular case of an ergodic set is cyclic. A cyclic chain is an ergodic chain in which each state can only be entered at certain periodic intervals. Such a chain has a period d, and its states are subdivided into d cyclic sets ($d > 1$).

In any finite Markov chain, no matter where the process starts, the probability after n steps that the process is in an ergodic state tends to 1 as n tends to infinity.

An absorbing chain is one all of whose ergodic states are absorbing; or, equivalently, which has at least one absorbing state, and such that an absorbing state can be reached from every state.

In a chain with transient sets the process moves towards the ergodic sets. The probability that the process is in an ergodic set tends to 1, and it cannot escape from an ergodic set once it enters it.

Theorem 3. *There exists no error-bounded probabilistic one-counter automaton with two one-way input tapes recognizing the language L.*

Proof. Assume from the contrary that A is such an automaton recognizing L with a probability $\frac{1}{2} + \delta$. We introduce a notion of *approximating Markov chain*

(shortly: AMC) for the automaton A. AMC gives only partial information about the automaton A but it allows to classify several possibilities of properties of the assumed automaton in order to get to contradiction using these properties explicitly.

AMC for the automaton A depends on four parameters: n being a natural number (contradiction is obtained by considering $n \to \infty$), parameters

$$\{(r_1, r_2) \mid r_1 \in \{1, 2, 3, 4\}, r_2 \in \{1, 2, 3, 4\}\},$$

and a word $u \in \{a_1, a_2, a_3, a_4\}^*$. The work of A cannot be described as a Markov chain. The AMC is the Markov chain obtained by simulating the work of the probabilistic one-counter automaton A on infinitely long input words, namely, the word on the first input tape being u continued by ω-word $a_{r_1}a_{r_1}a_{r_1}\ldots$ and the word on the second input tape being the ω-word $a_{r_2}a_{r_2}a_{r_2}\ldots$

We consider the work of A starting from the moment when the reading of the first input tape has reached the last symbol of u. In the process of the simulation the counter is always neglected—assuming that A always gets information "the counter is not empty". The symbols read from the two input tapes do not change anymore. The state of the automaton at each moment is completely described by a Markov chain. This Markov chain is our $AMC(n, u, r_1, r_2)$.

Hence the simulated process may differ from the work of A. Rather the AMC describes a part of the work of A.

We start our analysis of A by considering $AMC(1, u_0, 1, 3)$ where u_0 is an empty word. This means that A does not receive any information from the input tapes but the length of the words read. Moreover, AMC has only a finite number of states, and does not have ability to remember these lengths.

By Theorem 3.1.1 in [11] the probability after n steps that the process is in an ergodic state tends to 1 as n tends to infinity. By Theorem 4.1.4 in [11] each ergodic set of states tends to a stable distribution of probabilities. For arbitrary state s in arbitrary ergodic set, by $\xi(s)$ we denote the minimum recurring time, i.e. the minimum positive number of steps when the probability to go from the state s to the same state s is positive.

Let $s_i, \ldots, s_j$ be all distinct states in the same ergodic set. Let ξ be the least common multiple of $\xi(s_i), \ldots, \xi(s_j)$. The Markov chain $AMC(1, u_0, 1, 3)$ cannot remember how many fragments of the length ξ have been read form the input. Hence, since A is to remember this, A remembers this by adding some number to the counter. If the states $s_i, \ldots, s_j$ allow different numbers to be added to the counter, then reading n fragments of the length ξ where n tends to infinity, produces a Gauss distribution of the numbers added to the counter. This implies that A is not able to distinguish between close values of the lengths. Hence each ergodic set has to be cyclic. Moreover, for each ergodic set there is a constant c such that, independent of which state the automaton A is in, reading ξ symbols from the first input adds c to the counter.

Note that there may be several ergodic sets of states. However, if the total number of the states is k then $k!$ is an upper bound for the least common multiple of the lengths of all these cycles.

Now we consider another AMC, namely, $AMC(n, u, 2, 3)$, where u is a word in a single-letter alphabet $\{a_1\}$ of the length n (now only letters a_2 are read from the first input). In a similar way we prove that for each ergodic set of this AMC there is a constant d such that, independent of which state the automaton A is in, reading ξ symbols from the first input adds d to the counter. This implies that if we are interested only in the content of the counter, the cut-and-paste arguments can be used exactly as in deterministic automata. We can add a fragment consisting of a constant number of symbols to the first input word and the only change in the perfomance of A is adding another d to the counter.

Let n be a natural number large enough to have three properties:
1) the probability after reading the first n symbols a_1 from the first input that the process is in an ergodic state exceeds $\frac{3+2\delta}{4}$.
2) $n \geq k!$.

Now we consider the work of A on the pair of words (w_1, w_2) such that w_1 has a prefix $(a_1)^n(a_2)^{n+k!}(a_4)^{n!+k!}$, w_2 has a prefix $(a_3)^m$ where m is large enough to assert that with probability exceeding $\frac{3+2\delta}{8}$ the automaton A does not leave the prefix $(a_3)^m$ of w_2 on the second input tape while reading $(a_1)^n(a_2)^{n+k!}(a_4)^{n!+k!}$ on the first input tape. Additionally, we complement (w_1, w_2) by symbols a_1, a_2, a_4 to get $(w_1, w_2) \in L_1 \cap L_2 \cap \overline{L_3} \cap L_4$.

Note that the prefix $(a_1)^n(a_2)^{n+k!}(a_4)^{n!+k!}$ is long enough to ensure that each of 3 fragments $(a_1)^n$, $(a_2)^{n+k!}$, $(a_4)^{n!+k!}$ allows to use the cut-and-paste arguments exactly as in deterministic automata. We can add a fragment consisting of a constant number of symbols to the first input word and the only change in the perfomance of A is adding a constant number to the counter. It is important to note that A has no possibility to remember how many times this adding has been performed.

If this addition is performed in no fragment of the prefix, then $(w_1, w_2) \notin L$ because Ambainis' function $f(1, 1, 0, 1) = 0$. If this addition is performed in the fragment $(a_1)^n$, then $(w_1, w_2) \in L$ because Ambainis' function $f(0, 1, 0, 1) = 1$. If this addition is performed in the fragments $(a_1)^n$ and $(a_4)^{n!+k!}$, then $(w_1, w_2) \in L$ because Ambainis' function $f(0, 1, 0, 0) = 1$. If this addition is performed in the fragments $(a_1)^n$ and $(a_2)^{n+k!}$ and $(a_4)^{n!+k!}$, then $(w_1, w_2) \notin L$ because Ambainis' function $f(0, 0, 0, 0) = 1$. However, the automaton A does not distinguish between these cases. Contradiction. □

References

1. Ambainis, A.: Polynomial degree vs. quantum query complexity. Journal of Computer and System Sciences, vol. 72, No. 2, pp. 220–238 (2006)
2. Balodis, K., Beriņa, A., Cīpola, K., Dimitrijevs, M., Iraids, J. et al.: On the state complexity of ultrametric finite automata. Proceedings of SOFSEM 2013, vol. 2, 1–9 (2013)
3. Bērziņa, A., Freivalds, R.: On Quantum Query Complexity of Kushilevitz Function. Proceedings of Baltic DB&IS, vol. 2, Riga, Latvia, pp. 57-65 (2004)
4. Buhrman, H. and De Wolf, R.: Complexity measures and decision tree complexity: a survey. Theoretical Computer Science, vol. 288, No. 1, pp. 21–43 (2002)

5. Dragovich, B. and Dragovich, A.: A p-adic model of DNA sequence and genetic code. p-Adic Numbers, Ultrametric Analysis, and Applications, 1, 1, 34–41 (2009)
6. Freivalds, R.: Ultrametric finite automata and Turing machines. Lecture Notes in Computer Science, 7907, 1–11 (2013)
7. Freivalds, R. and Zeugmann, T.: Active Learning of Recursive Functions by Ultrametric Algorithms. Lecture Notes in Computer Science, Springer, vol. 8327, pp.246–257 (2014)
8. Gouvea, F.Q.: p-adic Numbers: An Introduction (Universitext), Springer, 2nd edition (1983)
9. Grover, L.K.: A fast quantum mechanical algorithm for database search. Proceedings of the 28th ACM symposium on Theory of Computing, pp. 212–219 (1996)
10. Jēriņš, K., Balodis, K., Krišlauks, R., Cīpola, K. and Freivalds, R.: Ultrametric Query Algorithms. Proceedings of SOFSEM 2014, vol. 2, 87–94 (2014)
11. Kemeny, J.G., Snell, J.L.: Finite Markov Chains: With a New Appendix "Generalization of a Fundamental Matrix", Springer, 2nd edition (1983)
12. Khrennikov, A.Y.: Non-Archimedean Analysis: Quantum Paradoxes, Dynamical Systems and Biological Models, Kluwer Academic Publishers (1997)
13. Koblitz, N.: P-adic Numbers, p-adic Analysis, and Zeta-Functions (Graduate Texts in Mathematics), vol. 58, Springer, 2nd edition (1984)
14. Kondacs, A. and Watrous, J.: On the power of quantum finite state automata. Proc. IEEE FOCS'97, pp. 66–75 (1997)
15. Kozyrev, S.V.: Ultrametric analysis and interbasin kinetics. p-Adic Mathematical Physics, Proc. of the 2nd International Conference on p-Adic Mathematical Physics, American Institute Conference Proceedings, vol. 826, 121–128 (2006)
16. Krišlauks, R., Rukšāne, I., Balodis, K., Kucevalovs, I., Freivalds, R., Nāgele, I.: Ultrametric Turing machines with limited reversal complexity. Proceedings of SOFSEM 2013, vol. 2, 87–94 (2013)
17. Nielsen, M.A. and Chuang, I.L.: Quantum computation and quantum information. Cambridge University Press (2010)
18. Nisan, N. and Wigderson, A.: On rank vs. communication complexity. Combinatorica, vol. 15, No. 4, pp. 557–565 (1995)
19. Ostrowski, A.: Über einige Lösungen der Funktionalgleichung $\varphi(x)\varphi(y) = \varphi(xy)$. Acta Mathematica, 41, 1, 271–284 (1916)
20. Papadimitriou, Ch.H.: Computational complexity. John Wiley and Sons Ltd. (2003)
21. Vladimirov, V.S., Volovich, I.V., Zelenov, E.I.: p-Adic Analysis and Mathematical Physics. World Scientific, Singapore (1995)
22. Zariņa, S. and Freivalds, R.: Visualisation and ultrametric analysis of Koch fractals. Proc. 16th Japan Conference on Discrete and Computational Geometry and Graphs, Tokyo, 84–85 (2013)