Ultrametric Vs. Quantum Query Algorithms

Rūsiņš Freivalds

Institute of Mathematics and Computer Science, University of Latvia
Raiņa bulvāris 29, Riga, LV-1459, Latvia*
Rusins.Freivalds@mii.lu.lv

Abstract. Ultrametric algorithms are similar to probabilistic algorithms but they describe the degree of indeterminism by p-adic numbers instead of real numbers. This paper introduces the notion of ultrametric query algorithms and shows an example of advantages of ultrametric query algorithms over deterministic, probabilistic and quantum query algorithms.

Keywords: Nature-inspired models of computation, ultrametric algorithms, probabilistic algorithms, quantum algorithms.

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 [27,1,9,10,11,12,13,14] on traditional models of computation and [24,3,21,7] on quantum computation). Similarly, there are query algorithms of all those types.

Deterministic, nondeterministic, probabilistic and quantum query algorithms are widely considered in literature (e.g., see survey [6]). We introduce a new type of query algorithms, namely, ultrametric query algorithms. All ultrametric algorithms and particularly ultrametric query algorithms rather closely follow the example of the corresponding probabilistic and quantum algorithms.

A quantum computation with t queries is just a sequence of unitary transformations

$$U_0 \to O \to U_1 \to O \to \ldots \to U_{t-1} \to O \to U_t.$$

The U_j 's can be arbitrary unitary transformations that do not depend on the input bits $x_1, \ldots, x_n$. The O's are query (oracle) transformations which depend on $x_1, \ldots, x_n$. To define O, we represent basis states as $\mid i, z >$ where i consists of $\lceil \log(N+1) \rceil$ bits and z consists of all other bits. Then, O_x maps $\mid 0, z >$ to itself and $\mid i, z >$ to $(-1)^{x_i} \mid i, z >$ for $i \in \{1, \ldots, n\}$ (i.e., we change phase depending on x_i, unless $i = 0$ in which case we do nothing). The computation starts with

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

A.-H. Dediu et al. (Eds.): TPNC 2014, LNCS 8890, pp. 1–10, 2014.

a state $\mid 0 >$. Then, we apply $U_0, O_x, \ldots, O_x, U_t$ and measure the final state. The result of the computation is the rightmost bit of the state obtained by the measurement.

The quantum computation computes f exactly if, for every $x = (x_1, \ldots., x_n)$, the rightmost bit of $U_T O_x \ldots O_x U_0 \mid 0 >$ equals $f(x_1, \ldots., x_n)$ with certainty.

The quantum computation computes f with bounded error if, for every $x = (x_1, \ldots., x_n)$, the probability that the rightmost bit of $U_T O_x \ldots O_x U_0 \mid 0 >$ equals $f(x_1, \ldots., x_n)$ is at least $1 - \epsilon$ for some fixed . $\epsilon < \frac{1}{2}$.

2 Ultrametric Algorithms

A new type of indeterministic algorithms called *ultrametric* algorithms was introduced in [15]. An extensive research on ultrametric algorithms of various kinds has been performed by several authors (cf. [4,16,23,29]). 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 [26] 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 [28], chemistry [22] and molecular biology [8,19] 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 [24]). 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.

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

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

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:

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

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 [17,20].

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 algorithm [18] does not read all the input on any computation path. There is a similar situation in ultrametric query algorithms. It is more easy to accept the definition of ultrametric query algorithms in the case when there is only one accepting state in the algorithm. The 3-ultrametric query algorithm in Theorem 16 has only one accepting state.

4 Kushilevitz's Function

Kushilevitz exhibited a function f that provides the largest gap in the exponent of a polynomial in $deg(f)$ that gives an upper bound on $bs(f)$. Never published by Kushilevitz, the function appears in footnote 1 of the Nisan-Wigderson paper [25].

Kushilevitz's function h of 6 Boolean variables is defined as follows:
$h(z_1, \ldots, z_6) = \Sigma_i z_i - \Sigma_{i \neq j} z_i z_j + z_1z_3z_4 + z_1z_2z_5 + z_1z_4z_5 + z_2z_3z_4 + z_2z_3z_5 + z_1z_2z_6 + z_1z_3z_6 + z_2z_4z_6 + z_3z_5z_6 + z_4z_5z_6$.

To explore properties of the Kushilevitz's function we introduce 10 auxiliary sets of variables.

$$
\begin{array}{r|l}
S_1 = \{z_1, z_3, z_4\} & T_1 = \{z_2, z_5, z_6\} \\
S_2 = \{z_1, z_2, z_5\} & T_2 = \{z_3, z_4, z_6\} \\
S_3 = \{z_1, z_4, z_5\} & T_3 = \{z_2, z_3, z_6\} \\
S_4 = \{z_2, z_3, z_4\} & T_4 = \{z_1, z_5, z_6\} \\
S_5 = \{z_2, z_3, z_5\} & T_5 = \{z_1, z_4, z_6\} \\
S_6 = \{z_1, z_2, z_6\} & T_6 = \{z_3, z_4, z_5\} \\
S_7 = \{z_1, z_3, z_6\} & T_7 = \{z_2, z_4, z_5\} \\
S_8 = \{z_2, z_4, z_6\} & T_8 = \{z_1, z_3, z_5\} \\
S_9 = \{z_3, z_5, z_6\} & T_9 = \{z_1, z_2, z_4\} \\
S_{10} = \{z_4, z_5, z_6\} & T_{10} = \{z_1, z_2, z_3\}
\end{array}
$$

By S we denote the class $(S_1, \ldots, S_{10})$ and by T we denote the class $(T_1, \ldots, T_{10})$.

Lemma 5. *For every $i \in \{1, \ldots, 6\}$, the union $S_i \cup T_i$ equals $\{1, \ldots, 6\}$.*

Lemma 6. *For every $i \in \{1, \ldots, 6\}$, the variable z_i is a member of exactly 5 sets in S and a member of exactly 5 sets in T.*

Lemma 7. *For every $i \in \{1, \ldots, 6\}$, the variable z_i has an empty intersection with exactly 5 sets in S and with exactly 5 sets in T.*

Lemma 8. *For every pair (i, j) such that $i \neq j$ and $i \in \{1, \ldots, 6\}, j \in \{1, \ldots, 6\}$, the pair of variables (z_i, z_j) is a member of exactly 2 sets in S and a member of exactly 2 sets in T.*

Lemma 9. *For every pair (i, j) such that $i \neq j$ and $i \in \{1, \ldots, 6\}, j \in \{1, \ldots, 6\}$, the pair of variables (z_i, z_j) has an empty intersection with exactly 2 sets in S and with exactly 2 sets in T.*

Lemma 10. *For every triple (i, j, k) of pairwise distinct elements of $\{1, \ldots, 6\}$, the triple of variables (z_i, z_j, z_k) coincides either with some set $S_i \in S$ or with some set T_j.*

Lemma 11. *No triple (i, j, k) of pairwise distinct elements of $\{1, \ldots, 6\}$ is such that the triple of variables (z_i, z_j, z_k) is a member of both S and T.*

Lemma 12. *For every quadruple (i, j, k, l) of pairwise distinct elements of $\{1, \ldots, 6\}$, the quadruple of variables (z_i, z_j, z_k, z_l) contains exactly 2 sets $S_i \in S$ and exactly 2 sets $T_i \in T$.*

Proof. Immediately from Lemma 8. □

Lemma 13. *For every quintuple (i, j, k, l, m) of pairwise distinct elements of $\{1, \ldots, 6\}$, the quintuple of variables $(z_i, z_j, z_k, z_l, z_m)$ contains exactly 5 sets $S_i \in S$ and exactly 5 sets $T_i \in T$.*

Proof. Immediately from Lemma 6. □

Lemma 14. *1) If $\Sigma_i z_i = 0$ then $h(z_1, \ldots, z_6) = 0$.*
2) If $\Sigma_i z_i = 1$ then $h(z_1, \ldots, z_6) = 1$,
3) If $\Sigma_i z_i = 2$ then $h(z_1, \ldots, z_6) = 1$,
4) If $\Sigma_i z_i = 4$ then $h(z_1, \ldots, z_6) = 0$,
5) If $\Sigma_i z_i = 5$ then $h(z_1, \ldots, z_6) = 0$,
6) If $\Sigma_i z_i = 6$ then $h(z_1, \ldots, z_6) = 1$,
7) If $\Sigma_i z_i = 3$ and there exist 3 pairwise distinct (j, k, l) such that $(z_j = z_k = z_l = 1)$ and $(z_j, z_k, z_l) \in S$ then $h(z_1, \ldots, z_6) = 1$,
8) If $\Sigma_i z_i = 3$ and there exist 3 pairwise distinct (j, k, l) such that $(z_j = z_k = z_l = 1)$ and $(z_j, z_k, z_l) \in T$ then $h(z_1, \ldots, z_6) = 0$.

Proof. If $\Sigma_i z_i = 0$ then all monomials in the definition of $h(z_1, \ldots, z_6)$ equal zero. If $\Sigma_i z_i = 1$ then $\Sigma_i z_i = 1$ but all the other monomials in the definition of $h(z_1, \ldots, z_6)$ equal zero. If $\Sigma_i z_i = 2$ then $h(z_1, \ldots, z_6) = \Sigma_i z_i - \Sigma_{i \neq j} z_i z_j = 2-1$. If $\Sigma_i z_i = 3$ and $(z_j, z_k, z_l) \in S$ then $h(z_1, \ldots, z_6) = \Sigma_i z_i - \Sigma_{i \neq j} z_i z_j = 3-3+1$. If $\Sigma_i z_i = 3$ and $(z_j, z_k, z_l) \in T$ then $h(z_1, \ldots, z_6) = \Sigma_i z_i - \Sigma_{i \neq j} z_i z_j = 3-3+0$. If $\Sigma_i z_i = 4$ then, by Lemma 12, $h(z_1, \ldots, z_6) = \Sigma_i z_i - \Sigma_{i \neq j} z_i z_j = 4-6+2$. If $\Sigma_i z_i = 5$ then, by Lemma 13, $h(z_1, \ldots, z_6) = \Sigma_i z_i - \Sigma_{i \neq j} z_i z_j = 5-10+5$. If $\Sigma_i z_i = 6$ then $h(z_1, \ldots, z_6) = \Sigma_i z_i - \Sigma_{i \neq j} z_i z_j = 6-15+10$. □

By $\alpha(z_1, \ldots, z_6)$ we denote the cardinality of those $S_i = (z_j, z_k, z_l)$ such that $z_j = z_k = z_l = 1$. By $\beta(z_1, \ldots, z_6)$ we denote the cardinality of those $S_i = (z_j, z_k, z_l)$ such that $z_j = z_k = z_l = 0$.

Lemma 15. *1) For arbitrary 6-tuple* $(z_1, \ldots, z_6) \in \{0,1\}^6$, $h(z_1, \ldots, z_6) = 1$ *iff* $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6)$ *is congruent to 1 modulo 3.*
2) For arbitrary 6-tuple $(z_1, \ldots, z_6) \in \{0,1\}^6$, $h(z_1, \ldots, z_6) = 0$ *iff* $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6)$ *is congruent to 2 modulo 3.*

Proof. If $\Sigma_i z_i = 0$ then $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 0 - 10 \equiv 2(\mod 3)$. If $\Sigma_i z_i = 1$ then, by Lemma 7, $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 0 - 5 \equiv 1(\mod 3)$. If $\Sigma_i z_i = 2$ then, by Lemma 9, $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 0 - 2 \equiv 1(\mod 3)$. If $\Sigma_i z_i = 3$ and there exist 3 pairwise distinct (j,k,l) such that $(z_j = z_k = z_l = 1)$ and $(z_j, z_k, z_l) \in S$ then, by Lemmas 10 and 11, $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 1 - 0 \equiv 1(\mod 3)$. If $\Sigma_i z_i = 3$ and there exist 3 pairwise distinct (j,k,l) such that $(z_j = z_k = z_l = 1)$ and $(z_j, z_k, z_l) \in T$ then, by Lemmas 10 and 11, $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 0 - 1 \equiv 2(\mod 3)$. If $\Sigma_i z_i = 4$ then, by Lemma 12, $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 2 - 0 \equiv 2(\mod 3)$. If $\Sigma_i z_i = 5$ then, by Lemma 13, $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 5 - 0 \equiv 2(\mod 3)$. If $\Sigma_i z_i = 5$ then $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 10 - 0 \equiv 1(\mod 3)$. These results correspond to Lemma 14. □

Theorem 16. *There exists a 3-ultrametric query algorithm computing the Kushilevitz's function using 3 queries.*

Proof. The desired algorithm branches its computation path into 31 branches at the root. We assign to each starting edge of the computation path the amplitude $\frac{1}{61}$.

The first 10 branches (labeled with numbers $1, \ldots, 10$)correspond to exactly one set S_i.

Let S_i consist of elements z_j, z_k, z_l. Then the algorithm queries z_j, z_k, z_l. If all the queried values equal 1 then the algorithm goes to the state q_3. If all the queried values equal 0 then the algorithm goes to the state q_3 but multiplies the amplitude to (-1). (For the proof it is important that for every 3-adic number a the norm $\|-a\| = \|a\|$.) If the queried values are not all equal then the algorithm goes to the state q_4.

The next 10 branches (labeled with numbers $11, \ldots, 20$) also correspond to exactly one set S_i. Let S_i consist of elements z_j, z_k, z_l. Then the algorithm queries

z_j, z_k, z_l. If all the queried values equal 1 then the algorithm goes to the state q_5. If all the queried values equal 0 then the algorithm goes to the state q_3. If the queried values are not all equal then the algorithm goes to the state q_4 but multiplies the amplitude to (-1).

11 branches (labeled with numbers $21, \ldots, 31$) ask 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

$$A_3 = \frac{1}{31}(11 + \alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6)),$$

The 3-ultrametric query algorithm performs measurement of the state q_3. The amplitude A_3 is transformed into a rational number $\|A_3\|$. As it was noted in Section 3, 3-adic notation for the number 31 is $\ldots 000112$ and 3-adic notation for the number $\frac{1}{31}$ is $\ldots 0212111221021$. Hence, for every 3-adic integer γ, $\|\gamma\| = \|\frac{1}{31}\gamma\|$.

By Lemma 15, $\|11+\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6)\| = 1$ if $h(z_1, \ldots, z_6) = 1$ and $\|11 + \alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6)\| = \frac{1}{3}$ if $h(z_1, \ldots, z_6) = 0$. □

5 Conclusions

Theorem 16 shows that there exists a bounded error 3-ultrametric query algorithm for the Kushilevitz's function whose complexity is much smaller than complexity of any *known* deterministic, nondeterministic, probabilistic and quantum query algorithm for this function. Moreover, Lemma 15 heavily exploits advantages of ultrametric algorithms, and this invites to conjecture that Kushilevitz's function is specific for advantages of ultrametric algorithms.

More difficult problem is to compare theorem 16 with the provable lower bounds of complexity. It is known that deterministic and nondeterministic query complexity of the Kushilevitz's function is 6. There exists an exact quantum query algorithm for the Kushilevitz's function with complexity 5 (see paper [5]) but nobody can prove that exact quantum query complexity for this function exceeds 3. There is an indirect proof of this conjecture.

Iterated functions are defined as follows.

Define a sequence $h_1, h_2, \ldots$ with h_d being a function of 6^d variables by: $h_1 = h$, $h_{d+1} = h(h_d(x_1, \ldots, x_{6d}), h_d(x_{6d+1}, \ldots, x_{2\cdot 6d})), h_d(x_{2\cdot 6d+1}, \ldots, x_{3\cdot 6d})$, $h_d(x_{2\cdot 6d+1}, \ldots, x_{3\cdot 6d}), h_d(x_{3\cdot 6d+1}, \ldots, x_{4\cdot 6d}), h_d(x_{4\cdot 6d+1}, \ldots, x_{5\cdot 6d})$, $h_d(x_{5\cdot 6d+1}, \ldots, x_{6\cdot 6d}))$

A. Ambainis proved in [2] that even bounded error query complexity for the iterated Kushilevitz's function exceeds $\Omega((\frac{\sqrt{39}}{2})^d) = \Omega((3.12\ldots)^d)$. Had this proof been valid for $d = 1$, we would have that error bounded quantum query complexity for Kushilevitz's function exceeds 3. Unfortunately, Ambainis proof works for *large* values of d.

References

1. Ablayev, F.M., Freivalds, R.: Why sometimes probabilistic algorithms can be more effective. In: Wiedermann, J., Gruska, J., Rovan, B. (eds.) MFCS 1986. LNCS, vol. 233, pp. 1–14. Springer, Heidelberg (1986)
2. Ambainis, A.: Polynomial degree vs. quantum query complexity. Journal of Computer and System Sciences 72(2), 220–238 (2006)
3. Ambainis, A., Freivalds, R.: 1-way quantum finite automata: strengths, weaknesses and generalizations. In: Proc. IEEE FOCS 1998, pp. 332–341 (1998)
4. Balodis, K., Beriņa, A., Cīpola, K., Dimitrijevs, M., Iraids, J., Jēriņš, K., Kacs, V., Kalājs, J., Krišlauks, R., Lukstiņš, K., Raumanis, R., Scegulnaja, I., Somova, N., Vanaga, A., Freivalds, R.: On the state complexity of ultrametric finite automata. In: Proceedings of SOFSEM, vol. 2, pp. 1–9 (2013)
5. Bērziņa, A., Freivalds, R.: On quantum query complexity of kushilevitz function. In: Proceedings of Baltic DB&IS 2004, vol. 2, pp. 57–65 (2004)
6. Buhrman, H., Wolf, R.D.: Complexity measures and decision tree complexity: a survey. Theoretical Computer Science 288(1), 21–43 (2002)
7. Moore, C., Quantum, J.C.: automata and quantum grammars. Theoretical Computer Science 237(1-2), 275–306 (2000)
8. Dragovich, B., Dragovich, A.: A p-adic model of dna sequence and genetic code. p-Adic Numbers, Ultrametric Analysis, and Applications 1(1), 34–41 (2009)
9. Freivalds, R.: Recognition of languages with high probability on different classes of automata. Doklady Akademii Nauk SSSR 239(1), 60–62 (1978)
10. Freivalds, R.: Projections of languages recognizable by probabilistic and alternating finite multi-tape automata. Information Processing Letters 13(4-5), 195–198 (1981)
11. Freivalds, R.: On the growth of the number of states in result of the determinization of probabilistic finite automata. Avtomatika i Vichislitel'naya Tekhnika (3), 39–42 (1982)
12. Freivalds, R.: Complexity of probabilistic versus deterministic automata. In: Barzdins, J., Bjorner, D. (eds.) Baltic Computer Science. LNCS, vol. 502, pp. 565–613. Springer, Heidelberg (1991)
13. Freivalds, R.: Languages recognizable by quantum finite automata. In: Farré, J., Litovsky, I., Schmitz, S. (eds.) CIAA 2005. LNCS, vol. 3845, pp. 1–14. Springer, Heidelberg (2006)
14. Freivalds, R.: Non-constructive methods for finite probabilistic automata. International Journal of Foundations of Computer Science 19, 565–580 (2008)
15. Freivalds, R.: Ultrametric finite automata and turing machines. In: Béal, M.-P., Carton, O. (eds.) DLT 2013. LNCS, vol. 7907, pp. 1–11. Springer, Heidelberg (2013)
16. Freivalds, R., Zeugmann, T.: Active learning of recursive functions by ultrametric algorithms. In: Geffert, V., Preneel, B., Rovan, B., Štuller, J., Tjoa, A.M. (eds.) SOFSEM 2014. LNCS, vol. 8327, pp. 246–257. Springer, Heidelberg (2014)
17. Gouvea, F.Q.: p-adic numbers: An introduction, universitext (1983)
18. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the 28th ACM Symposium on Theory of Computing, pp. 212–219 (1996)
19. Khrennikov, A.Y.: Non-Archimedean Analysis: Quantum Paradoxes, Dynamical Systems and Biological Models. Kluwer Academic Publishers (1997)
20. Koblitz, N.: P-adic Numbers, p-adic Analysis, and Zeta-Functions, 2nd edn. Graduate Texts in Mathematics, vol. 58. Springer (1984)
21. Kondacs, A., Watrous, J.: On the power of quantum finite state automata. In: Proc. IEEE FOCS 1997, pp. 66–75 (1997)

22. Kozyrev, S.V.: Ultrametric analysis and interbasin kinetics. In: Proc. of the 2nd International Conference on p-Adic Mathematical Physics, vol. 826, pp. 121–128. American Institute Conference Proceedings (2006)
23. Krišlauks, R., Rukšāne, I., Balodis, K., Kucevalovs, I., Freivalds, R., Agele, I.N.: Ultrametric turing machines with limited reversal complexity. In: Proceedings of SOFSEM, vol. 2, pp. 87–94 (2013)
24. Nielsen, M.A., Chuang, I.L.: Quantum computation and quantum information. Cambridge University Press (2000)
25. Nisan, N., Wigderson, A.: On rank vs. communication complexity. Combinatorica 15(4), 557–565 (1995)
26. Ostrowski, A.: Über einige Lösungen der Funktionalgleichung $\varphi(x)\varphi(y) = \varphi(xy)$. Acta Mathematica 41(1), 271–284 (1916)
27. Papadimitriou, C.H.: Computational complexity. John Wiley and Sons Ltd, Chichester (2003)
28. Vladimirov, V.S., Volovich, I.V., Zelenov, E.I.: p-Adic Analysis and Mathematical Physics. World Scientific, Singapore (1995)
29. Zariņa, S., Freivalds, R.: Visualisation and and ultrametric analysis of koch fractals. In: Proc. 16th Japan Conference on Discrete and Computational Geometry and Graphs, pp. 84–85. Tokyo (2013)