Ultrametric Algorithms and Automata

Rūsiņš Freivalds(✉)

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

Abstract. We introduce a notion of ultrametric automata and Turing machines using p-adic numbers to describe random branching of the process of computation. These automata have properties similar to the properties of probabilistic automata but complexity of probabilistic automata and complexity of ultrametric automata can differ very much.

1 Introduction

Irrational numbers can be represented by decimal digits

$$2.718281828\ldots$$

In this representation infinitely many digits are allowed on the right-hand side but only a finite number of them on the left-hand side. Informally, non-terminating decimals are easily understood, because it is clear that a real number can be approximated to any required degree of precision by a terminating decimal. If two decimal expansions differ only after the 10th decimal place, they are quite close to one another; and if they differ only after the 20th decimal place, they are even closer.

10-adic numbers use a similar non-terminating expansion, but with a different concept of "closeness". Whereas two decimal expansions are close to one another if their difference is a large negative power of 10, two 10-adic expansions are close if their difference is a large positive power of 10.

Thus 3333 and 4333, which differ by 10^3, are close in the 10-adic world, and 33333333 and 43333333 are even closer, differing by 10^7.

More precisely, a rational number r can be expressed as $10^a \cdot \frac{p}{q}$, where p and q are positive integers and q is relatively prime to p and to 10. For each $r \neq 0$ there exists the maximal a such that this representation is possible. Let the 10-adic norm of r to be

$$| r |_{10} = \frac{1}{10^a},$$

$$| 0 |_{10} = 0.$$

The research was supported by Grant No. 271/2012 from the Latvian Council of Science and by the project ERAF Nr.2DP/2.1.1.1/13/APIA/VIAA/027. Partially supported by Latvian State Research programme NexIT project No.1 "Technologies of ontologies, semantic web and security".

C.S. Calude and M.J. Dinneen (Eds.): UCNC 2015, LNCS 9252, pp. 35–52, 2015.
DOI: 10.1007/978-3-319-21819-9_2

Closeness in any number system is defined by a metric. Using the 10-adic metric the distance between numbers x and y is given by $\mid x - y \mid_{10}$.

The "closeness" in the 10-adic metric may seem strange. This reminds me an occasion in the seminar "Algebra and Logics" lead by Professor Anatoly Maltsev in Novosibirsk University nearly 50 years ago. A young student was presenting a paper on modifications of the Tarski-Kuratowski algorithm. I do not remember whether the paper was by the student himself or whether he presented some foreign paper. At some moment the presenter proudly announced that the best earlier known version of the algorithm had time complexity $\exp\exp\ldots\exp n$ (9 times exponent) but the version in the paper was much better, namely, $\exp\exp\ldots\exp n$ (7 times exponent). Professor Maltsev immediately commented that he did not see much difference there. Boris Trakhtenbrot came to the presenter's help and noticed that the improvement is not just exponential but exponent-of-exponential. After short but emotional discussion the participants agreed that the improvement is essential but the difference is negligible.

On that day I learned that humans do not distinguish between huge and very huge numbers. Similarly, they not distinguish between small and very small numbers. The Archimedean metric is good in mathematics but not in the everyday life.

An interesting consequence of the 10-adic metric (or of a p-adic metric) is that there is no longer a need for the negative sign. As an example, by examining the following sequence we can see how unsigned 10-adics can get progressively closer and closer to the number 1:

$$\begin{aligned} 9 &= -1 + 10 & \mid 9 - 1 \mid_{10} &= \tfrac{1}{10} \\ 99 &= -1 + 100 & \mid 99 - 1 \mid_{10} &= \tfrac{1}{100} \\ 999 &= -1 + 1000 & \mid 999 - 1 \mid_{10} &= \tfrac{1}{1000} \\ 9999 &= -1 + 10000 & \mid 9999 - 1 \mid_{10} &= \tfrac{1}{10000} \\ 99999 &= -1 + 100000 & \mid 99999 - 1 \mid_{10} &= \tfrac{1}{100000} \end{aligned}$$

$$\ldots 9999999999999999999999999999999999999 = -1$$

10-adic numbers have a major drawback. It is possible to find pairs of non-zero 10-adic numbers (having an infinite number of digits, and thus not rational) whose product is 0.

The reason for this property turns out to be that 10 is a composite number which is not a power of a prime. This problem is simply avoided by using a prime number p as the base of the number system instead of 10 and indeed for this reason p in p-adic is usually taken to be prime.

Let p be an arbitrary prime number. We will call *p-adic digit* a natural number between 0 and $p - 1$ (inclusive). A *p-adic integer* is by definition a sequence $(a_i)_{i \in N}$ of p-adic digits. We write this conventionally as

$$\cdots a_i \cdots a_2 a_1 a_0$$

(that is, the a_i are written from left to right).

If n is a natural number, and

$$n = \overline{a_{k-1}a_{k-2}\cdots a_1a_0}$$

is its p-adic representation (in other words $n = \sum_{i=0}^{k-1} a_i p^i$ with each a_i a p-adic digit) then we identify n with the p-adic integer (a_i) with $a_i = 0$ if $i \geq k$. This means that natural numbers are exactly the same thing as p-adic integer only a finite number of whose digits are not 0. The number 0 is the p-adic integer all of whose digits are 0, and that 1 is the p-adic integer all of whose digits are 0 except the right-most one (digit 0) which is 1.

If $\alpha = (a_i)$ and $\beta = (b_i)$ are two p-adic integers, we will now define their sum. To that effect, we define by induction a sequence (c_i) of p-adic digits and a sequence (ϵ_i) of elements of $\{0, 1\}$ (the "carries") as follows:

- ϵ_0 is 0.
- c_i is $a_i + b_i + \epsilon_i$ or $a_i + b_i + \epsilon_i - p$ according as which of these two is a $p-$ adic digit (in other words, is between 0 and p - 1). In the former case,$\epsilon_i+1 = 0$ and in the latter, $\epsilon_i + 1 = 1$.

Under those circumstances, we let $\alpha + \beta = (c_i)$ and we call $\alpha + \beta$ the sum of α and β. Note that the rules described above are exactly the rules used for adding natural numbers in p-adic representation. In particular, if α and β turn out to be natural numbers, then their sum as a p-adic integer is no different from their sum as a natural number. So $2 + 2 = 4$ remains valid (whatever p is but if $p = 2$ it would be written $\cdots 010 + \cdots 010 = \cdots 100$). Here is an example of a 7-adic addition:

$$\begin{array}{r}
\cdots 2\,5\,1\,4\,1\,3 \\
\cdots 1\,2\,1\,1\,0\,2 \\
\hline
\cdots 4\,0\,2\,5\,1\,5
\end{array}$$

This addition of p-adic integers is associative, commutative, and verifies $\alpha + 0 = \alpha$ for all α (recall that 0 is the p-adic integer all of whose digits are 0). Subtraction of p-adic integers is also performed in exactly the same way as that of natural numbers in p-adic form. Note that this subtraction scheme gives us the negative integers readily: for example, subtract 1 from 0 (in the 7-adics) :

$$\begin{array}{r}
\cdots 0\,0\,0\,0\,0\,0 \\
\cdots 0\,0\,0\,0\,0\,1 \\
\hline
\cdots 6\,6\,6\,6\,6\,6
\end{array}$$

(each column borrows a 1 from the next one on the left). So $-1 = \cdots 666$ as 7-adics. More generally, -1 is the p-adic all of whose digits are $p-1$, -2 has all of its digits equal to $p-1$ except the right-most which is $p-2$, and so on. In fact, (strictly) negative integers correspond exactly to those p-adics all of whose digits except a finite number are equal to $p-1$.

It can then be verified that p-adic integers, under addition, form an abelian group.

We now proceed to describe multiplication. First note that if n is a natural number and α is a p-adic integer, then we have a naturally defined $n \cdot \alpha = \alpha + \alpha + \ldots + \alpha$ (n times, with $0 \cdot \alpha = 0$, of course). This limited multiplication satisfies some obvious equalities, such as $(m + n)\alpha = m\alpha + n\alpha, n(\alpha + \beta) = n\alpha + n\beta, m(n\alpha) = (mn)\alpha$, and so on. Note also that multiplying by $p = \cdots 0010$ is the same as adding a 0 on the right. Multiplying two p-adic integers on the other hand requires some more work. To do that, we note that if $\alpha_0, \alpha_1, \alpha_2, \cdots$ are p-adic integers, with α_1 ending in (at least) one zero, α_2 ending in (at least) two zeros, and so on, then we can define the sum of all the α_i, even though they are not finite in number. Indeed, the last digit of the sum is just the last digit of α_0 (since $\alpha_1, \alpha_2, \cdots$ all end in zero), the second-last is the second-last digit of $\alpha_0 + \alpha_1$ (because $\alpha_2, \alpha_3, \cdots$ all end in 00), and so on: every digit of the (infinite) sum can be calculated with just a finite sum. Now we suppose that we want to multiply α and $\beta = (b_i)$ two p-adic integers. We then let $\alpha_0 = b_0\alpha$ (we know how to define this since b_0 is just a natural number), $\alpha_1 = pb_1\alpha$, and so on: $\alpha_i = p^i b_i \alpha$. Since α_i is a p-adic integer multiplied by p_i, it ends in i zeros, and therefore the sum of all the α_i can be defined. This procedure may sound complicated, but, it is the usual algorithm to multiply two natural numbers. Here is an example of a 7-adic multiplication:

$$\begin{array}{r}
\cdots 2\,5\,1\,4\,1\,3 \\
\cdots 1\,2\,1\,1\,0\,2 \\
\hline
\cdots 5\,3\,3\,1\,2\,6 \\
\cdots 0\,0\,0\,0\,0 \\
\cdots 1\,4\,1\,3 \\
\cdots 4\,1\,3 \\
\cdots 2\,6 \\
\cdots 3 \\
\hline
\cdots 3\,1\,0\,4\,2\,6
\end{array}$$

To have 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 "decimal" point but only a finite number of digits to the right of it.

However, p-adic numbers is not merely one of generalizations 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 pairs of elements (x, y) of X to the nonnegative real numbers such that

1. $d(x, y) = 0$ 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.

The *norm* of an element $x \in X$ is the distance from 0:

1. $\| x \| = 0$ if and only if $x = y$,
2. $\| x.y \| = \| x \| . \| xy \|$,
3. $\| x + y \| \leq \| x \| + \| y \|$.

We know one metric on Q induced by the ordinary absolute value. However, there are other norms as well.

A norm is called *ultrametric* if the third requirement can be replaced by the stronger statement: $\| x + y \| \leq \max\{\| x \|, \| y \|\}$. Otherwise, the norm is called *Archimedean*.

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

Definition 2. *Let $p \in \{2, 3, 5, 7, 11, 13, \cdots\}$ be any prime number. For arbitrary rational number x, its* p-norm *is:*

$$||x||_p = \begin{cases} \frac{1}{p^{\mathrm{ord}_p x}}, \text{if} & x \neq 0, \\ \neg p_i, \text{if} & x = 0 \ ; \end{cases}$$

Rational numbers are p-adic integers for all prime numbers p. The nature of irrational numbers is more complicated. For instance, $\sqrt{2}$ just does not exist as a p-adic number for some prime numbers p. More precisely, $\sqrt{a}$ can be represented as a p-adic number if and only if a is a quadratic residue modulo p, i.e. if the congruence $x^2 = a(modp)$ has a solution. On the other hand, there is a continuum of p-adic numbers not being real numbers. Moreover, there is a continuum of 3-adic numbers not being 5-adic numbers, and vice versa.

p-adic numbers are described in much more detail in [7,11,14].

ascal 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*.

String theory [18], chemistry [12] and molecular biology [2,10] have introduced p-adic numbers to describe measures of indeterminism.

Popularity of usage of p-adic numbers can be explained easily. There is a well-known difficulty to overcome the distinction between *continuous* and *discrete* processes. For instance, according to Rutherford's model of atoms, the electrons can be situated only on specific orbits. When energy of an electron increases, there is a quantum leap. Niels Bohr proposed, in 1913, what is now called the Bohr model of the atom. He suggested that electrons could only have certain classical motions:

1. Electrons in atoms orbit the nucleus.
2. The electrons can only orbit stably, without radiating, in certain orbits (called by Bohr the "stationary orbits"): at a certain discrete set of distances from the nucleus. These orbits are associated with definite energy levels. In these orbits, the electron's acceleration does not result in radiation and energy loss as required by classical electromagnetics.
3. Electrons can only gain and lose energy by jumping from one allowed orbit to another, absorbing or emitting electromagnetic radiation with a frequency determined by the energy difference of the levels according to the Planck relation.

One of the methods to model such quantum leaps is to consider p-adic numbers and there norms. The p-adic numbers can have continuum distinct values but their norms can have only denumerable values. If a variable gradually changes taking p-adic values, its norm performs quantum leaps. Hence usage of p-adic numbers as measures of indeterminism provides a mechanism which is similar to probabilistic model but mathematically different from it.

There were no difficulties to implement probabilistic automata and algorithms practically. Quantum computation [9] has made a considerable theoretical progress but practical implementation has met considerable difficulties. However, prototypes of quantum computers exist, some quantum algorithms are implemented on these prototypes, quantum cryptography is already practically used. Some people are skeptical concerning practicality of the initial spectacular promises of quantum computation but nobody can deny the existence of quantum computation.

We consider a new type of indeterministic algorithms called *ultrametric* algorithms. They 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 the parameters. Slightly simplifying the description of the definitions one can say that ultrametric algorithms are the same probabilistic algorithms, only the interpretation of the probabilities is different.

Our choice of p-adic numbers instead of real numbers is not quite arbitrary. In 1916 Alexander Ostrowski [16] proved that any non-trivial absolute value on the rational numbers Q is equivalent to either the usual real absolute value or a p-adic absolute value. This result shows that using p-adic numbers is not merely one of many possibilities to generalize the definition of deterministic algorithms but rather the only remaining possibility not yet explored.

Moreover, Helmut Hasse's local-global principle states that certain types of equations have a rational solution if and only if they have a solution in the real numbers and in the p-adic numbers for each prime p.

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 have 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 irrational numbers. However, there is an important feature that distinguishes p-adic numbers from real numbers. Real numbers (both rational and irrational) are linearly

ordered. 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. 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.

Ultrametric finite automata and ultrametric Turing machines are reasonably similar to probabilistic finite automata and Turing machines.

2 First Examples

The notion of p-adic numbers widely used in mathematics but not so much in Computer Science. It seems that the first author having proposed to use p-adic numbers to analyze finite automata has been A.G.Lunts [13]. The first papers containing definition of ultrametric finite automata in the sense used in this survey were [5,6].

The aim of our next sections is to show that the notion of ultrametric automata and ultrametric Turing machines is natural.

In mathematics, a stochastic matrix is a matrix used to describe the transitions of a Markov chain. A *right stochastic matrix* is a square matrix each of whose rows consists of nonnegative real numbers, with each row summing to 1. A *stochastic vector* is a vector whose elements consist of nonnegative real numbers which sum to 1. The *finite probabilistic automaton* is defined [3,4] as an extension of a non-deterministic finite automaton $(Q, \Sigma, \delta, q_0, F)$, with the initial state q_0 replaced by a stochastic vector giving the probability of the automaton being in a given initial state, and with stochastic matrices corresponding to each symbol in the input alphabet describing the state transition probabilities. It is important to note that if A is the stochastic matrix corresponding to the input symbol a and B is the stochastic matrix corresponding to the input symbol b, then the product AB describes the state transition probabilities when the automaton reads the input word ab. Additionally, the probabilistic automaton has a threshold λ being a real number between 0 and 1. If the probabilistic automaton has only one *accepting state* then the input word x is said to be accepted if after reading x the probability of the accepting state has a probability exceeding λ. If there are several accepting states, the word x is said to be accepted the total of probabilities of the accepting states exceeds λ.

Ultrametric automata are defined exactly in the same way as probabilistic automata, only the parameters called *probabilities of transition from one state to another one* are real numbers between 0 and 1 in probabilistic automata, and they are p-adic numbers called *amplitudes* in the ultrametric automata. Formulas to calculate the amplitudes after one, two, three, $\cdots$ steps of computation are exactly the same as the formulas to calculate the probabilities in the probabilistic automata. Following the example of finite quantum automata, we demand that the input word x is followed by a special end-marker. At the beginning of the work, the states of the automaton get *initial amplitudes* being p-adic numbers. When reading the current symbol of the input word, the automaton changes the

amplitudes of all the states according to the transition matrix corresponding to this input symbol. When the automaton reads the end-marker, the *measurement* is performed, and the amplitudes of all the states are transformed into the p-norms of these amplitudes. The norms are rational numbers and it is possible to compare whether or not the norm exceeds the threshold λ. If total of the norms for all the accepting states of the automaton exceeds λ, we say that the automaton accepts the input word.

Ultrametric algorithms are described by finite directed acyclic graphs (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 β_p. We refer to β_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 [8] 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 10 has only one accepting state.

Paavo Turakainen considered various generalizations of finite probabilistic automata in 1969 and proved that there is no need to demand in cases of probabilistic branchings that total of probabilities for all possible continuations equal 1. He defined generalized probabilistic finite automata where the "probabilities" can be arbitrary real numbers, and that languages recognizable by these generalized probabilistic finite automata are the same as for ordinary probabilistic finite automata. Hence we also allow usage of all possible p-adic numbers in p-ultrametric machines. Remembering the theorem by P.Turakainen [17] we start with the most general possible definition hoping to restrict it if we below find examples of not so natural behavior of ultrametric automata. (Moreover, we do not specify all the details of the definitions in Theorems 1–4, and make the definition precise only afterwards. The reader may consider such a presentation strange but we need some natural examples of ultrametric automata before we concentrate on one standard definition.)

However, it is needed to note that if there is only one accepting state then the possible probabilities of acceptance are discrete values $0, p^1, p^{-1}, p^2, p^{-2}, p^3, \cdots$. Hence there is no natural counterpart of *isolated cut-point* or *bounded error* for ultrametric machines. On the other hand, a counterpart of Turakainen's theorem for probabilistic automata with isolated cut-point still does not exist. We also did not succeed to prove such a theorem for ultrametric automata. Most probably, there are certain objective difficulties.

Theorem 1. *There is a continuum of languages recognizable by finite ultrametric automata.*

Proof. Let $\beta = \cdots 2a_3 2a_2 2a_1 2a_0 2$ be an arbitrary p-adic number (not p-adic integer) where $p \geq 3$ and all $a_i \in \{0, 1\}$. Denote by B the set of all possible such β. Consider an automaton A_β with 3 states, the initial amplitudes of the states being $(\beta, -1, -1)$. The automaton is constructed to have the following property. If the input word is $2a_0 2a_1 2a_2 2a_3 2 \cdots 2a_n 2$ then the amplitude of the first state becomes $\cdots 2a_{n+4} 2a_{n+3} 2a_{n+2} 2a_{n+1} 2$. To achieve this, the automaton adds -2, multiplies to p, adds $-a_n$ and again multiplies to p.

Now let β_1 and β_2 be two different p-adic numbers. Assume that they have the same first symbols $a_m \cdots 2a_3 2a_2 2a_1 2a_0 2$ but different symbols a_{m+1} and b_{m+1}. Then the automaton accepts one of the words $a_{m+1} 2a_m \cdots 2a_3 2a_2 2a_1 2a_0 2$ and rejects the other one $b_{m+1} 2a_m \cdots 2a_3 2a_2 2a_1 2a_0 2$. Hence the languages are distinct. □

Definition 5. *Finite p-ultrametric automaton is called* **integral** *if all the parameters of it are p-adic integers.*

Automata recognizing nonrecursive languages cannot be considered natural. Hence we are to restrict our definition.

Theorem 2. *There exists a finite integral p-ultrametric automaton recognizing the language $\{0^n 1^n\}$.*

Proof. When the automaton reads 0 it multiplies the amplitude to 2, and when it reads 1 it multiplies it to $\frac{1}{2}$. The norm of the amplitude equals p^0 iff the number of zeros is equal to the number of ones. □

We consider the following language.

$$L = \{w | w \in \{0,1\}^* \text{ and } w = w^{rev}\}$$

Theorem 3. *For every prime number $p \geq 5$, there is an integral p-ultrametric automaton recognizing L.*

Proof. The automaton has two special states. If the input word is

$$a(1)a(2)\cdots a(n)a(n+1)a(n+2)\cdots a(2n+1)$$

then one of these states has amplitude

$$a(1)p^n + \cdots + a(n)p^{+1} + a(n+1)p^0 + a(n+2)p^{-1} + \cdots + a(2n)p^{-n+1} + a(2n+1)p^{-n}$$

and the other one has amplitude

$$-a(1)p^{-n} - \cdots - a(n)p^{-1} - a(n+1)p^0 - a(n+2)p^{+1} - \cdots - a(2n)p^{+n-1} + a(2n+1)p^{+n}$$

If the sum of these two amplitudes equals 0 then the input word is a palindrome. Otherwise, the sum of amplitudes has a norm removed from p^0. □

Definition 6. *A square matrix with elements being p-adic numbers is called* **balanced** *if for arbitrary row of the matrix the product of p-norms of the elements equals 1.*

Definition 7. *A finite ultrametric automaton is called* **balanced** *if all the matrices in its definition are balanced.*

Theorem 4. *If a language M can be recognized by a finite ultrametric automaton then M can be recognized also by a balanced finite ultrametric automaton.*

Proof. For every state of the automaton we add its duplicate. If the given state has an amplitude γ then its duplicate has the amplitude $\frac{1}{\gamma}$. Product of balanced matrices is balanced. □

Definition 8. *A balanced finite ultrametric automaton is called* **regulated** *if there exist constants λ and c such that $0 < c < 1$ and for arbitrary input word x and for arbitrary state of the automaton the norm $c\lambda < \| \gamma \|_p < \frac{\lambda}{c}$. We say that the word x is accepted if $\| \gamma \|_p > \lambda$ and it is rejected if $\| \gamma \|_p \leq \lambda$.*

Theorem 5. *[6] (1) If a language M is recognized by a regulated finite ultrametric automaton then M is regular.*
(2) For arbitrary prime number p there is a constant c_p such that if a language M is recognized by a regulated finite p-ultrametric automaton with k states then there is a deterministic finite automaton with $(c_p)^{k.log k}$ states recognizing the language M.

3 Non-regulated Finite Automata

Since the numbers 1 and 0 are also p-adic numbers, every deterministic finite automaton can be described in terms of matrices for transformation of amplitudes. Hence every regular language is recognizable by a regulated p-ultrametric automaton. There is a natural problem : are there languages for which regulated p-ultrametric automata can have smaller complexity, i.e. smaller number of states.

The following 3 theorems seem to present such an example but there is a catch: these automata are not regulated because the norm of the amplitude to be measured can be arbitrary small (for lengthy input words).

Theorem 6. *For arbitrary prime number $p \geq 3$ the language*

$$L_{p-1} = \{1^n \mid n \equiv p-1(\ mod\ p)\}$$

is recognizable by a p-ultrametric finite automaton with 2 states.

Proof. A primitive root modulo n is any number g with the property that any number coprime to n is congruent to a power of g modulo n. In other words, g is a generator of the multiplicative group of integers modulo n. Existence of primitive roots modulo prime numbers was proved by Gauss. The initial amplitude 1 of a special state in our automaton is multiplied to an arbitrary primitive root modulo p. When the end-marker is read the amplitude -1 of the other state is added to this amplitude. The result has p-norm p^0 iff $n \equiv p-1$. □

Theorem 7. *For arbitrary prime number $p \geq 3$ the language*

$$L_p = \{1^n \mid n \equiv p(\ mod\ p)\}$$

is recognizable by a p-ultrametric finite automaton with 2 states.

Proof. The value 1 of the amplitude of the second state is added to the amplitude of the accepting state at every step of reading the input word. The result has p-norm p^0 iff $n \equiv p$. □

Theorem 8. *For arbitrary natural number m there are infinitely many prime numbers p such that the language*

$$L_m = \{1^n \mid n \equiv 0(\ mod\ m)\}$$

is recognizable by a p-ultrametric finite automaton with 2 states.

Proof. Dirichlet prime number theorem, states that for any two positive coprime integers m and d, there are infinitely many primes of the form $m + nd$, where $n \geq 0$. In other words, there are infinitely many primes which are congruent to m modulo d. The numbers of the form $mn + d$ form an arithmetic progression

$$d,\ m+d,\ 2m+d,\ 3m+d,\ \ldots,$$

and Dirichlet's theorem states that this sequence contains infinitely many prime numbers.

Let p be such a prime and g be a primitive root modulo p. Then the sequence of remainders $g, g^2, g^3, \cdots$ modulo p has period m and $n \equiv 0(\bmod m)$ is equivalent to $g^n \equiv d(\bmod p)$. Hence the automaton multiplies the amplitude of the special state to g and and adds $-d$ when reading the end-marker. □

4 Regulated Finite Automata

We wish to complement Theorem 5 by a proof showing that the gap between the complexity of regulated finite ultrametric automata and the complexity of deterministic finite automata is not overestimated. It turns out that this comparison is related to well-known open problems.

First, we consider a sequence of languages where the advantages of ultrametric automata over deterministic ones are super-exponential but the advantages are achieved only for specific values of the prime number p.

It is known that every p-permutation can be generated as a product of sequence of two individual p-permutations:

$$a = \begin{pmatrix} 1\,2\,3\cdots p-1\,p \\ 2\,3\,4\cdots \quad p \quad 1 \end{pmatrix}$$

$$b = \begin{pmatrix} 1\,2\,3\cdots p-1\,p \\ 2\,1\,3\cdots p-1\,p \end{pmatrix}$$

A string $x \in \{a, b\}^*$ is in the language M_p if the product of these p-permutations equals the trivial permutation.

Theorem 9. *(1) For arbitrary prime p, the language M_p is recognized by a p-ultrametric finite automaton with $p+2$ states.*
(2) If a deterministic finite automaton has less than $p! = c^{p.\log p}$ states then it does not recognize M_p.

Idea of the proof. The ultrametric automaton gives initial amplitudes $0, 1, 2, \cdots, p-1$ to p states of the automaton and after reading any input letter only permutes these amplitudes. After reading the endmarker from the input the automaton subtracts the values $0, 1, 2, \cdots, p-1$ from these amplitudes. □

5 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.

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. □

Theorem 10. *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 $\frac{1}{7}$.

Table 1. Values of the functions

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

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. 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. 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 $f(x_1, x_2, x_3, x_4) = 0$ □

6 Kushilevitz's Function

E. 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 [15].

Kushilevitz's function h of 6 Boolean variables is defined as follows:
$h(z_1, \ldots, z_6) = \Sigma_i z_i - \Sigma_{i \neq j} z_i z_j + z_1 z_3 z_4 + z_1 z_2 z_5 + z_1 z_4 z_5 + z_2 z_3 z_4 + z_2 z_3 z_5 + z_1 z_2 z_6 + z_1 z_3 z_6 + z_2 z_4 z_6 + z_3 z_5 z_6 + z_4 z_5 z_6$.

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

Lemma 3. *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 4. *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 5. *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 6. *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 7. *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 8. *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 9. *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 5. □

Lemma 10. *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 3. □

Lemma 11. *(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 9, $h(z_1, \ldots, z_6) = \Sigma_i z_i - \Sigma_{i \neq j} z_i z_j = 4 - 6 + 2$. If $\Sigma_i z_i = 5$ then, by Lemma 10, $h(z_1, \ldots, z_6) = \Sigma_i z_i - \Sigma_{i \neq j} z_i z_j = 5 - 10 + 5$. 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 12. *(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 $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 0 - 10 \equiv 2 \pmod 3$. If $\Sigma_i z_i = 1$ then, by Lemma 4, $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 0 - 5 \equiv 1 \pmod 3$. If $\Sigma_i z_i = 2$ then, by Lemma 6, $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 0 - 2 \equiv 1 \pmod 3$. 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 7 and 8, $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 1 - 0 \equiv 1 \pmod 3$. 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 7 and 8, $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 0 - 1 \equiv 2 \pmod 3$. If $\Sigma_i z_i = 4$ then, by Lemma 9, $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 2 - 0 \equiv 2 \pmod 3$. If $\Sigma_i z_i = 5$ then, by Lemma 10, $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 5 - 0 \equiv 2 \pmod 3$. If $\Sigma_i z_i = 5$ then $\alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 10 - 0 \equiv 1 \pmod 3$. These results correspond to Lemma 11. □

Theorem 11. *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 Sect. 1, 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 12, $11 + \alpha(z_1, \ldots, z_6) - \beta(z_1, \ldots, z_6) = 1$ 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 $h(z_1, \ldots, z_6) = 0$. □

7 Iterated Kushilevitz's Function

The iterated Kushilevitz's function is defined as follows. Let $f_1(x_1, \ldots, x_6)$ be the Kushilevitz's function.

$$f_{d+1} = f(f_d(x_1, \ldots, x_{6^d}), f_d(x_{6^d+1}, \ldots, x_{2.6^d}), \ldots, f_d(x_{5.6^d+1}, \ldots, x_{6.6^d})$$

Theorem 12. *There exists a* 3*-ultrametric query algorithm computing the function* g_d *using* $\cdot 3^d$ *queries.*

References

1. Ambainis, A.: Polynomial degree vs. quantum query complexity. J. Comput. Syst. Sci. **72**(2), 220–238 (2006)
2. Radu, V.: Application. In: Radu, V. (ed.) Stochastic Modeling of Thermal Fatigue Crack Growth. ACM, vol. 1, pp. 63–70. Springer, Heidelberg (2015)
3. Freivalds, R.: Complexity of probabilistic versus deterministic automata. In: Bārzdinš, J., Bjørner, D. (eds.) Baltic Computer Science. LNCS, vol. 502, pp. 565–613. Springer, Heidelberg (1991)
4. Freivalds, R.: Non-constructive methods for finite probabilistic automata. Int. J. Found. Comput. Sci. **19**(3), 565–580 (2008)
5. Freivalds, R.: Ultrametric automata and Turing machines. In: Voronkov, A. (ed.) Turing-100. EPiC Series, vol. 10, pp. 98–112. EasyChair (2012)
6. 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)
7. Gouvea, F.Q.: p-adic Numbers: An Introduction. Universitext, 2nd edn. Springer, Heidelberg (1983)
8. 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)

9. Hirvensalo, M.: Quantum Computing. Springer, Heidelberg (2001)
10. Khrennikov, A.Yu.: Non-Archimedean Analysis: Quantum Paradoxes, Dynamical Systems and Biological Models. Kluwer Academic Publishers, Dordrecht (1997)
11. Koblitz, N.: p-adic Numbers, p-adic Analysis, and Zeta-Functions. Graduate Texts in Mathematics, vol. 58, 2nd edn. Springer, Heidelberg (1984)
12. Kozyrev, S.V.: Ultrametric analysis and interbasin kinetics. In: Proceedings of the 2nd International Conference on p-Adic Mathematical Physics, American Institute Conference Proceedings, vol. 826, pp. 121–128 (2006)
13. Lunts, A.G.: A method of analysis of finite automata. Sov. Phys. Dokl. **10**, 102–105 (1965)
14. Madore, D.A.: A first introduction to p-adic numbers. http://www.madore.org/david/math/padics.pdf
15. Nisan, N., Wigderson, A.: On rank vs. communication complexity. Combinatorica **15**(4), 557–565 (1995)
16. Ostrowski, A.: Über einige Lösungen der funktionalgleichung $\varphi(x)\varphi(y) = \varphi(xy)$. Acta Math. **41**(1), 271–284 (1916)
17. Turakainen, P.: Generalized automata and stochastic languages. Proc. Am. Math. Soc. **21**(2), 303–309 (1969)
18. Vladimirov, V.S., Volovich, I.V., Zelenov, E.I.: p-Adic Analysis and Mathematical Physics. World Scientific, Singapore (1995)

Springer

http://www.springer.com/978-3-319-21818-2

Unconventional Computation and Natural Computation
14th International Conference, UCNC 2015, Auckland, New Zealand, August 30 -- September 3, 2015, Proceedings

Calude, C.S.; Dinneen, M.J. (Eds.)

2015, X, 301 p. 58 illus., Softcover
ISBN: 978-3-319-21818-2