Ultrametric automata and Turing machines *

Rūsiņš Freivalds

Institute of Mathematics and Computer Science, University of Latvia,
Raiņa bulvāris 29, Riga, LV-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

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*.

String theory [18], chemistry [14] and molecular biology [3, 12] have introduced p-adic numbers to describe measures of indeterminism.

There were no difficulties to implement probabilistic automata and algorithms practically. Quantum computation [10] 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 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

*The research was supported by Project 2009/0216/1DP/1.1.1.2.0/09/IPIA/VIA/044 from the European Social Fund.

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.

Below we consider ultrametric versus deterministic Turing machines with one input tape which can be read only 1-way and a work tape which is empty at the beginning of the work.

2 p-adic numbers

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

$$n = \overline{a_{k-1}a_{k-2}\cdots a_1 a_0}$$

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.

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 $a \equiv 0(\mod p^m)$. 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:*

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

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. 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 [9, 13].

3 First examples

The notion of p-adic numbers widely used in mathematics but not so much in Computer Science. 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 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.

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 3. *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 ultrametric automaton recognizing the language* $\{0^n1^n\}$.

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 4. *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 5. *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 6. *A balanced finite ultrametric automaton is called* **regulated** *if there exist constants* $c > 0$ *and* λ *such that for arbitrary input word* x *the norm* $\lambda - c < \| \gamma \|_p < \lambda + c$. *We say that the word* x *is accepted if* $\| \gamma \|_p > \lambda$ *and it is rejected if* $\| \gamma \|_p \leq \lambda$.

Theorem 5. *(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.

4 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 $n \equiv 0(\ \mathrm{mod}\ m)$ is equivalent to $g^n \equiv d(\ \mathrm{mod}\ p)$. Hence the automaton multiplies the amplitude of the special state to g and and adds $-d$ when reading the end-marker. □

5 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:

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

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

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 $p! = c^{p \cdot \log p}$ 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. Notice that if a prime number p' is such that $p' < p$ then the p'-ultrametric automaton can need more than $p + 2$ states. □

Now we wish to present a sequence of languages such that the size advantages of ultrametric automata can be achieved for all sufficiently large values of p. Unfortunately, we have proved only exponential advantages, and even the base of the exponent is very small.

Linear codes is the simplest class of codes. The alphabet used is a fixed choice of a finite field $GF(q) = F_q$ with q elements. For most of this Section we consider a special case of $GF(2) = F_2$. These codes are binary codes.

A generating matrix G for a linear $[n, k]$ code over F_q is a k-by-n matrix with entries in the finite field F_q, whose rows are linearly independent. The linear code corresponding to the matrix G consists of all the q^k possible linear combinations of rows of G. The requirement of linear independence is equivalent to saying that all the q^k linear combinations are distinct. The linear combinations of the rows in G are called codewords. However we are interested in something more. We need to have the codewords not merely distinct but also as far as possible in terms of Hamming distance. Hamming distance between two vectors $v = (v_1, \ldots, v_n)$ and $w = (w_1, \ldots, w_n)$ in F_{q^k} is the number of indices i such that $v_i \neq w_i$.

The textbook [8] contains

Theorem A. *For any integer $n \geq 4$ there is a $[2n, n]$ binary code with a minimum distance between the codewords at least $n/10$.*

However the proof of the theorem in [8] has a serious defect. It is non-constructive. It means that we cannot find these codes or describe them in a useful manner. This is why P.Garret calls them mirage codes.

If q is a prime number, the set of the codewords with the operation "component-wise addition" is a group. Finite groups have useful properties. We single out Lagrange's Theorem. The order of a finite group is the number of elements in it.

Lagrange's Theorem (see e.g. [8]). *Let GR be a finite group. Let H be a subgroup of GR. Then the order of H divides the order of G.*

Definition 7. *A generating matrix G of a linear code is called* **cyclic** *if along with an arbitrary row $(v_1, v_2, v_3, \ldots, v_n)$ the matrix G contains also a row $(v_2, v_3, \ldots, v_n, v_1)$.*

We would wish to prove a reasonable counterpart of Theorem A for cyclic mirage codes, but this attempt fails. Instead we consider binary generating matrices of a bit different kind. Let p be an odd prime number, and x be a binary word of length p. The generating matrix $G(p, x)$ has p rows and $2p$ columns. Let $x = x_1x_2x_3 \ldots x_p$. The first p columns (and all p rows) make a unit matrix with elements 1 on the main diagonal and 0 in all the other positions. The last p columns (and all p rows) make a cyclic matrix with $x = x_1x_2x_3 \ldots x_p$ as the first row, $x = x_px_1x_2x_3 \ldots x_{p-1}$ as the second row, and so on.

Definition 8. *We say that the numbering $\Psi = \{\Psi_0(x), \Psi_1(x), \Psi_2(x), \ldots\}$ of 1-argument partial recursive functions is* **computable** *if the 2-argument function $U(n, x) = \Psi_n(x)$ is partial recursive.*

Definition 9. *We say that a numbering Ψ is reducible to the numbering η if there exists a total recursive function $f(n)$ such that, for all n and x, $\Psi_n(x) = \eta_{f(n)}(x)$.*

Definition 10. *We say that a computable numbering φ of all 1-argument partial recursive functions is a* **Gödel numbering** *if every computable numbering (of any class of 1-argument partial recursive functions) is reducible to φ.*

Definition 11. *We say that a Gödel numbering ϑ is a* **Kolmogorov numbering** *if for arbitrary computable numbering Ψ (of any class of 1-argument partial recursive functions) there exist constants $c > 0, d > 0$, and a total recursive function $f(n)$ such that:*

1. *for all n and x, $\Psi_n(x) = \vartheta_{f(n)}(x)$,*
2. *for all n, $f(n) \leq c \cdot n + d$.*

Kolmogorov's Theorem [15]. *There exists a Kolmogorov numbering.*

There exist many distinct Kolmogorov numberings. We now fix one of them and denote it by η. Since Kolmogorov numberings give indices for all partial recursive functions, for arbitrary x and p, there is an i such that $\eta_i(p) = x$. Let $i(x, p)$ be the minimal i such that $\eta_i(p) = x$. It is easy to see that if $x_1 \neq x_2$, then $i(x_1, p) \neq i(x_2, p)$. We consider all binary words x of the length p and denote by $x(p)$ the word x such $i(x, p)$ exceed $i(y, p)$ for all binary words y of the length p different from x. It is obvious that $i \geq 2^p - 1$.

Until now we considered generating matrices $G(p, x)$ for independently chosen p and x. From now on we consider only odd primes p such that 2 is a primitive root modulo p and the matrices $G(p, x(p))$. Freivalds [6] proved that if p is **sufficiently large**, then Hamming distances between

arbitrary two codewords in this linear code is at least $\frac{4p}{19}$. (The gap in Theorem 9 below might be superexponential had we proven an explicit bound what this "sufficiently large" means. Unfortunately, usage of Kolmogorov complexity is intrinsically nonconstructive.)

Above we mentioned a binary generating matrix $G(p, p(x))$ for a linear code. Now we use this matrix to construct a probabilistic reversible automaton $R(p)$.

The matrix $G(p, x(p))$ has $2p$ columns and p rows. The automaton $R(p)$ has $4p + 1$ states, $2p$ of them being accepting and $2p+1$ being rejecting. The input alphabet consists of 2 letters.

The states $q_1, q_2, \ldots, g_{4p}$ are related to the columns of $G(p, x(p))$ and should be considered as $2p$ pairs $(q_1, q_2), (q_3, q_4), \ldots, \ldots (q_{4p-1}, q_{4p})$ corresponding to the $2p$ columns of $G(p, x(p))$. The states
$q_1, q_3, q_5, q_7, \ldots, q_{4p-1}$ are accepting and the states $q_2, q_4, q_6, q_8, \ldots, q_{4p}$ are rejecting. The initial probability distribution is as follows:

$$\begin{cases} \frac{1}{2p}, & \text{for each of } q_1, q_3, \ldots, q_{4p-1} \ , \\ 0, & \text{for each of } q_2, q_4, \ldots, q_{4p} \ . \end{cases}$$

The processing of the input symbols a, b is deterministic. Under the input symbol a the states are permuted as follows:

$q_1 \to q_3$	$q_2 \to q_4$	$q_{2p+1} \to q_{2p+3}$	$q_{2p+2} \to q_{2p+4}$
$q_3 \to q_5$	$q_4 \to q_6$	$q_{2p+3} \to q_{2p+5}$	$q_{2p+4} \to q_{2p+6}$
$q_5 \to q_7$	$q_6 \to q_8$	$q_{2p+5} \to q_{2p+7}$	$q_{2p+6} \to q_{2p+8}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$
$q_{2p-3} \to q_{2p-1}$	$q_{2p-2} \to q_{2p}$	$q_{4p-3} \to q_{4p-1}$	$q_{4p-2} \to q_{4p}$
$q_{2p-1} \to q_1$	$q_{2p} \to q_2$	$q_{4p-1} \to q_{2p+1}$	$q_{4p} \to q_{2p+2}$

The permutation of the states under the input symbol b depends on $G(p, x(p))$. Let be

$$G(p, x(p)) = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1\ 2p} \\ g_{21} & g_{22} & \cdots & g_{2\ 2p} \\ \cdots & \cdots & \cdots & \cdots \\ g_{p1} & g_{p2} & \cdots & g_{p\ 2p} \end{pmatrix}$$

For arbitrary $i \in \{1, 2, \ldots, p\}$,

$$\begin{cases} q_{2i-1} \to q_{2i-1} & \text{, if } g_{1i} = 0 \\ q_{2i} \to q_{2i} & \text{, if } g_{1i} = 0 \\ q_{2i-1} \to q_{2i} & \text{, if } g_{1i} = 1 \\ q_{2i} \to q_{2i-1} & \text{, if } g_{1i} = 1. \end{cases}$$

In order to understand the language recognized by the automaton $R(p)$ we consider the following auxiliary mapping CW from the words in $\{a, b\}^*$ into the set of binary $2p$-vectors defined recursively (starting from the empty word Λ):

1. $CW(\Lambda) = g_{11} g_{12} \ldots g_{1\ 2p}$
2. if $CW(w) = h_1 h_2 h_3 \ldots h_p h_{p+1} h_{p+2} h_{p+3} \ldots h_{2p}$ then

$$\begin{cases} CW(wa) & = h_p h_1 h_2 \ldots h_{p-1} h_{2p} h_{p+1} h_{p+2} \ldots h_{2p-1} \quad \text{and} \\ CW(wb) & = (h_1 \oplus g_{11})(h_2 \oplus g_{12})(h_3 \oplus g_{13}) \ldots (h_{2p} \oplus g_{1\ 2p}) \ . \end{cases}$$

The next two lemmas can be proved by induction over the length of w.

Lemma 1. *For arbitrary word $w \in \{a,b\}^*$, $CW(w)$ is a codeword in the linear code corresponding to the generating matrix $G(p,x(p))$.*

Lemma 2. *Let w be an arbitrary word in $\{a,b\}^*$, and $CW(w) = h_1h_2\dots h_{2p}$. Then the amplitude distribution of the states in $R(p)$ is*

$$\begin{cases} \frac{1}{2p} & \text{, for } g_{2i-1} \text{ if } h_i = 0, \\ 0 & \text{, for } g_{2i} \text{ if } h_i = 0, \\ 0 & \text{, for } g_{2i-1} \text{ if } h_i = 1, \\ \frac{1}{2p} & \text{, for } g_{2i} \text{ if } h_i = 1. \end{cases}$$

We introduce a language

$$L_{G(p,x(p))} = \{w | CW(w) = 000\dots0\}.$$

Lemma 3. *If 2 is a primitive root modulo p and p is sufficiently large, then the automaton $R(p)$ recognizes the language $L_{G(p,x(p))}$*

Lemma 4. *For arbitrary p and arbitrary deterministic finite automaton A recognizing $L_{G(p,x(p))}$ the number of states of A is no less than 2^p.*

Proof. First, we show that every codeword $z = h_1h_2\dots h_{2p}$ is a value of $CW(w)$ for a suitable $z \in \{a,b\}^*$. Since the first p columns of $G(p,x(p))$ are columns of the unit $(p \times p)$ matrix, $z = h_1(g_{11}, g_{12}, \dots, g_{1\ 2p}) + h_2(g_{21}, g_{22}, \dots, g_{2\ 2p}) + \dots + h_{2p}(g_{p1}, g_{p2}, \dots, g_{p\ 2p})$. Then $z = CW(b^{h_1}ab^{h_2}ab^{h_3}a\dots ab^{h_{2p}}a)$.

Second, let u and v be two distinct input words such that $CW(u) \neq CW(v)$. A cannot remember u and v by the same state. Hence the number of the states is at least 2^p. □

Lemmas 3 and 4 imply

Theorem 10. *If 2 is a primitive root for infinitely many distinct primes then there exists an infinite sequence of regular languages $L_1, L_2, L_3, \dots$ in a 2-letter alphabet and a sequence of positive integers $p(1), p(2), p(3), \dots$ such that for arbitrary j:*

1. *any deterministic finite automaton recognizing L_j has at least $2^{p(j)}$ states,*
2. *there is a regulated finite ultrametric automaton with $(4p(j)+1)$ states recognizing L_j.*

Emil Artin made in 1927 a famous conjecture the validity of which is still an open problem.

Artin's Conjcture [1]. If a is neither -1 nor a square, then a is a primitive root for infinitely many primes.

Moreover, it is conjectured that density of primes for which a is a primitive root equals $A = 0.373956\dots$. In 1967, C.Hooley [11] proved that Artin's conjecture follows from the Generalized Riemann hypothesis.

The original Riemann hypothesis (RH) [16] was formulated in 1859. It was a statement about the zeroes of the complex Zeta function. Since then RH has become one of the most notorious open problems in mathematics. Several equivalent formulations of RH have been found. We follow [2] to present here RH and its generalizations.

$\pi(x)$ is the number of primes less than or equal to x. $\pi(x,n,a)$ is the number of primes less than or equal to x and congruent to a mod n.

$$li(x) = \int_2^x \frac{dt}{\log t}$$

The Euler phi function, $\phi(n)$, is defined to be the number of positive integers $\leq n$ which are relatively prime to n.

Riemann Hypothesis For arbitrary $\epsilon > 0$, we have

$$\pi(x) = li(x) + O(x^{\frac{1}{2}+\epsilon}).$$

Extended Riemann Hypothesis Let n and a be relatively prime integers. Then for arbitrary $\epsilon > 0$, we have

$$\pi(x, n, a) = \frac{li(x)}{\phi(n)} + O(x^{\frac{1}{2}+\epsilon}).$$

Generalized Riemann Hypothesis Let K be an algebraic number field and let $\pi_k(n)$ denote the number of prime ideals whose norm is $\leq x$. Then for arbitrary $\epsilon > 0$, we have

$$\pi_k(x) = li(x) + O(x^{\frac{1}{2}+\epsilon}).$$

Hooley's Theorem [11] *Assume GRH. Let $a \geq 2$ be a squarefree integer, with $a \not\equiv 1 \pmod 4$. Let $\pi_a(x)$ denote the number of primes less than or equal to x for which a is a primitive root. Then as $x \to \infty$,*

$$\pi_a(x) = A\frac{x}{\log x} + O(\frac{x(\log a + \log\log x)}{(\log x)^2})$$

where A is the Artin's constant $0.373956\ldots$.

On the other hand, D.R.Heath-Brown [4] proved the following theorem without assumption of any unproved conjectures.

Heath-Brown Theorem [4] *If a, b, and c are distinct odd primes, then the number of $p \leq x$ for which at least one of these is a primitive root modulo p is $\Omega(\frac{x}{(\log x)^2})$.*

It follows from this theorem that Artin's conjecture can be wrong no more than for 2 distinct primes a.

Corollary 1. *Assume Artin's Conjecture. Then Theorem 10 holds.*

Corollary 2. *Assume Generalized Riemann Hypothesis. Then Theorem 10 holds.*

6 2-way finite automata

Our Theorem 2 proved that the language $\{0^n1^n\}$ can be recognized by finite integral p-ultrametric automata for all prime $p \geq 3$. Unfortunately, the automaton exhibited there was not regulated. For 2-way automata this theorem can be strengthened. Idea of the proof is similar to that used in [5].

Theorem 11. *For every prime $p \geq 3$ there exists a regulated 2-way finite integral p-ultrametric automaton recognizing the language $\{0^n1^n\}$.*

7 1-way pushdown automata

Let $A = \{a, b, c, d, e, f, g, h, k, l, m, p, q, r, s, t, u, v\}$. Now we consider a language T in the alphabet $A \cup \{\#\}$ for which both the deterministic and probabilistic 1-way pushdown automata cannot recognize the language but there exists an ultrametric 1-way pushdown automaton recognizing it.

The language T is defined as the set of all the words x in the input alphabet such that either x is in all 9 languages T_i described below or in exactly 6 of them or in exactly 3 of them or in none of them where

$$T_1 = \{x\#y \mid x \in A^* \wedge y \in A^* \wedge proj_{ab}(x) = proj_{ab}(y)\},$$

$$T_2 = \{x\#y \mid x \in A^* \wedge y \in A^* \wedge proj_{cd}(x) = proj_{cd}(y)\},$$

$$T_3 = \{x\#y \mid x \in A^* \wedge y \in A^* \wedge proj_{ef}(x) = proj_{ef}(y)\},$$

$$T_4 = \{x\#y \mid x \in A^* \wedge y \in A^* \wedge proj_{gh}(x) = proj_{gh}(y)\},$$

$$T_5 = \{x\#y \mid x \in A^* \wedge y \in A^* \wedge proj_{kl}(x) = proj_{kl}(y)\},$$

$$T_6 = \{x\#y \mid x \in A^* \wedge y \in A^* \wedge proj_{mp}(x) = proj_{mp}(y)\},$$

$$T_7 = \{x\#y \mid x \in A^* \wedge y \in A^* \wedge proj_{qr}(x) = proj_{qr}(y)\},$$

$$T_8 = \{x\#y \mid x \in A^* \wedge y \in A^* \wedge proj_{st}(x) = proj_{st}(y)\},$$

$$T_9 = \{x\#y \mid x \in A^* \wedge y \in A^* \wedge proj_{uv}(x) = proj_{uv}(y)\}.$$

Theorem 12. *For the language T we have the following properties.*

(1) *There is a regulated 3-ultrametric 1-way pushdown automaton recognizing the language T.*

(2) *No deterministic 1-way pushdown automata can recognize the language T.*

(3) *No probabilistic 1-way pushdown automata can recognize the language T can have bounded error.*

8 Query algorithms

Many papers on query algorithms consider computation of Boolean functions. The input of the query algorithm is a black box oracle containing the values of the variables $x_1 = a_1, x_2 = a_2, \cdots, x_n = a_n$ for an explicitly known Boolean function $f(x_1, \cdots, x_n)$. The result of the query algorithm is to be the value $f(a_1, \cdots, a_n)$. The query algorithm can ask for the values of the variables. The queries are asked individually, and the result of any query influences the next query to be asked or the result to be output.

The complexity of the query algorithm is defined as the number of the queries asked to the black box oracle. Deterministic query algorithms prescribe the next query uniquely depending only on the previously received answers from the black box oracle. Probabilistic query algorithms allow randomization of the process of computation. They sometimes allow to reduce the complexity of the algorithm dramatically.

Following [7], we consider in this paper a more general class of functions $f(x_1, \cdots, x_n)$, namely, functions $\{0,1,2,\cdots,n-1\}^n \to \{0,1\}$. The domain $\{0,1,2,\cdots,n\}^n$ is restricted to a particularly interesting case - permutations. For instance,

$$x_0 = 4, x_1 = 3, x_2 = 2, x_3 = 1, x_4 = 0$$

can be considered as a permutation of 5 symbols {0,1,2,3,4} usually described as 43210. Under such a restriction the functions $f : \{0,1,2,\cdots,n-1\}^n \to \{0,1\}$ can be considered as properties of permutations. For instance, the function

$$f(0,1,2) = 1, f(1,2,0) = 1, f(2,0,1) = 1, f(0,2,1) = 0, f(1,0,2) = 0, f(2,1,0) = 0$$

describes the property of 3-permutations to be *even* (as opposed to the property to be *odd*).

Theorem 13. *There is a property P of permutations such that for each natural number n there is a number m that for all primes $p > m$ the property P can be checked by a p-ultrametric query algorithm with 1 query only while any deterministic query algorithm for n-permutations needs at least $n-1$ queries to check the property P.*

Idea of the proof. The property P is whether the given permutation is not the trivial permutation. The ultrametric query algorithm, in parallel processing, asks one value of x_i and gives the accepting state the amplitude 1 if $x_i = i$ and 0, otherwise. The norm of the sum of all these amplitudes exceeds 0 iff the given permutation has the property P. Notice that for large n the norm of the amplitude can become unrestrictedly close to 0. □

Theorem 14. *For arbitrary natural number m, there is a property P_m of n-permutations where $n = 4m^2 + 4m + 1$, such that:*
(1) Every deterministic query algorithm for P_m needs $4m^2 + 6m + 2$ queries.
(2) For all sufficiently large primes p, there is a p-ultrametric query algorithm deciding P_m with $m + 1$ queries.

Idea of the proof. We consider finite projective geometries with $4m^2+4m+1 = (2m)^2 + 2m + 1$ points and $4m^2 + 4m + 1 = (2m)^2 + 2m + 1$ lines such that each linPG_me contains $2m+1$ points, every pair of lines intersect in precisely 1 point, and given any two distinct points, there is exactly one line that contains both points. Let GR_m be the group of permutations preserving the geometry PG_m. The p-ultrametric query algorithm, by parallel processing, for every line of PG_m, where the points are denoted by $a1, a2, \cdots, a(2m+1)$ asks two sets of queries. In the first set there are queries $x_{a1}, \cdots, x_{a(m+1)}$. In the second set there are queries $x_{a(m+1)}, \cdots, x_{a(2m+1)}$. If the permutation is in GR_m then the answers to one set of queries uniquely determine the other set of answers. If p is sufficiently large, it is possible to give amplitudes α for all possible answers such that the amplitudes annihilate iff they correspond. Norm of the amplitude can be only either 0 or 1 even for very long words. □

9 Turing machines

We denote by pUP the class of all languages recognizable by p-ultrametric Turing machines in a polynomial time. This is a large class of languages.

Theorem 15. *If a language M is recognizable by a probabilistic Turing machine in a polynomial time then for arbitrary $p \geq 3$ there is a p-ultrametric Turing machine recognizing M in a polynomial time.*

Proof. The class PP of all languages recognizable in a polynomial time has natural complete problems, for example, $MAJSAT$. $MAJSAT$ is a decision problem in which one is given a Boolean formula F. The answer must be YES if more than half of all assignments $x_1, x_2, \cdots, x_n$ make F true and NO otherwise. Hence M is reducible to $MAJSAT$ in deterministic polynomial time. On the other hand, $MAJSAT$ is recognizable by a p-ultrametric Turing machine in a polynomial time. This machine considers in parallel all possible assignments for $x_1, x_2, \cdots, x_n$ and adds a p-adic number 2^{-n} to the amplitude α of a special state. F is in $MAJSAT$ iff the resulting amplitude α has p-norm 0. □

Riemann surface is a notion useful to study functions of complex variable [19]. We introduce a discrete counterpart of this notion.

Definition 12. *A* discrete Riemann surface *on the rectangle $[a,b]\times[c,d]$ is a map from (x,y,z) (where $x\in[a,b]$, $y\in[c,d]$ and z is a string of symbols from a finite alphabet Σ whose length equals $y-c$) to a finite alphabet Δ. For each triple its* neighbors *are defined as the triples:*
(1) (x,y',z) where either $y'=y+1$ or $y'=y+1$,
(2) (x',y,z') where either $x'=x-1$ and z' is z with the last symbol omitted, or $x'=x+1$ and z' is z with the one symbol attached at its end.

Definition 13. *A* discrete Dirichlet condition *is a 5-tuple consisting of: (1) a map from (x,y) where $y=c$ to Δ, (2) a map from (x,y) where $y=d$ to Δ, (3) (x,y) where $x=a$ to Δ, (4) (x,y) where $x=b$ to Δ, and (5) neighboring conditions that may forbid some simultaneous maps of neighboring triples.*

Definition 14. *The* discrete Dirichlet problem *is whether or not a Riemann surface is possible consistent with the given discrete Dirichlet condition.*

Theorem 16. *For arbitrary prime number $p\geq 3$, there is a pUP-complete language.*

Idea of the proof. The language consists of all discrete Dirichlet conditions such that the discrete Dirichlet problem has a positive answer. The map in the Riemann surface can be used to describe the work of a ultrametric Turing machine. The symbols of Δ for all possible values of x for a fixed y and z describe the configuration of the tape at the moment y with the choices z made before the moment y and the amplitudes accumulated. The discrete Dirichlet problem asks whether the ultrametric machine accepts the input word. The difference $d-c$ represents the computation time allowed. □

References

[1] Emil Artin. Beweis des allgemeinen reziprozitätsgesetzes. *Mat. Sem. Univ. Hamburg*, 5:353–363, 1927.

[2] E. Bach and J. Shallit. *Algorithmic Number Theory.* MIT Press, 1996.

[3] Branko Dragovich and Alexandra Dragovich. A p-adic model of dna sequence and genetic code. *p-Adic Numbers, Ultrametric Analysis, and Applications*, 1(1):34–41, 2009.

[4] D.R.Heath-Brown. Artin's conjecture for primitive roots. *The Quarterly Journal of Mathematics*, 37:27–38, 1986.

[5] Rūsiņš Freivalds. Probabilistic two-way machines. *Lecture Notes in Computer Science*, 118:33–45, 1981.

[6] Rūsiņš Freivalds. Non-constructive methods for finite probabilistic automata. *International Journal of Foundations of Computer Science*, 19(3):565–580, 2008.

[7] Rūsiņš Freivalds and Kazuo Iwama. Quantum queries on permutations with a promise. In *14th International Conference on Implementation and Application of Automata (CIAA 2009)*, volume 5642 of *Lecture Notes in Computer Science*, pages 208–216. Springer, 2009.

[8] Paul Garret. *The Mathematics of Coding Theory*. Pearson Prentice Hall, Upper Saddle River, 2004.

[9] Fernando Quadros Gouvea. *p-adic Numbers: An Introduction*. Springer, 2nd edition, 1983.

[10] Mika Hirvensalo. *Quantum Computing*. Springer-Verlag, Berlin, Heidelberg, 2001.

[11] C. Hooley. On artin's conjecture. *Journal für die reine und angewandte Mathematik*, 225:229–220, 1967.

[12] Andrei Yu. Khrennikov. *Non-Archimedean Analysis: Quantum Paradoxes, Dynamical Systems and Biological Models*. Kluwer Academic Publishers, 1997.

[13] Neal Koblitz. *p-adic Numbers, p-adic Analysis, and Zeta-Functions*. Springer, 2nd edition, 1984. (Graduate Texts in Mathematics) (v. 58).

[14] Sergey V. Kozyrev. Ultrametric analysis and interbasin kinetics. In *p-Adic Mathematical Physics, Proc. of the 2nd International Conference on p-Adic Mathematical Physics*, volume 826 of *American Institute Conference Proceedings*, pages 121–128. American Institute of Physics, 2006.

[15] Andrei N.Kolmogorov. Three approaches to the quantitative definition of information. *Problems in Information Transmission*, 1(1):1–7, 1965. In Russian.

[16] B. Riemann. Ueber die anzahl der primzahlen unter einer gegebenen grösse. *Monatsberichte der Königlichen Preußischen Akademie der Wissenschaften zu Berlin*, (November 1859):671–680, 1859.

[17] Paavo Turakainen. Generalized automata and stochastic languages. *Proceedings of the American Mathematical Society*, 21(2):303–309, 1969.

[18] V. S. Vladimirov, I. V. Volovich, and E. I. Zelenov. *p-Adic Analysis and Mathematical Physics*. World Scientific, Singapore, 1995.

[19] Hermann Weyl. *The concept of a Riemann surface*. Dover Publications, New York, 2009.