

Ultrametric Finite Automata and Their Capabilities

Maksims DIMITRIJEVS, Kristīne CĪPOLA

University of Latvia Faculty of Computing, Raiņa bulvāris 19, Rīga, LV-1586, Latvia
md09032@lu.lv, kristine.cipola@gmail.com

Abstract. Ultrametric automata use p -adic numbers to describe the random branching of the process of computation. Previous research has shown that ultrametric automata can recognize complex languages and can have small number of states when classical automata require much more states. In this paper, we present a survey on ultrametric automata and their language recognition capabilities.

Keywords: Ultrametric automata, p -adic numbers, complexity

1 Introduction

p -adic numbers are used in different sciences, including chemistry and physics (Kozyrev, 2006, Vladimirov et al., 1995). There are numerous efficient applications of p -adic theory to computer science, namely, to numerical analysis, data mining and data analysis, experiment design, theoretical programming, cryptography etc. (Murtagh, 2009, Krishnaswami et al., 2012, Anashin, 2010).

Application of p -adic numbers to automata theory started half a century ago by A. G. Lunts (Lunts). A. G. Lunts paper deals with transducers rather than automata. In the paper it was shown that a letter-to-letter transducer over an alphabet of p^n symbols can be described to a function which is p -adic metric.

p -adic methods have been successfully applied to the theory of formal languages and automata which recognize languages ultrametric too. One of the earlier monographs on this subject is research published by Jean-Eric Pin (Perrin, Pin, 2004). There are several papers that deal with automata which recognize languages by ultrametric (in particular, p -adic) methods. For example, a part of the paper (Grigorchuk, 2000) deals with automata and uses ultrametric methods.

In 2013 Rūsiņš Freivalds introduced the idea of using basic properties of p -adic numbers and of p -adic metric in Turing machines and finite automata to describe the random branching of the process of computation (Freivalds, 2013). He proved that the

use of p -adic numbers exposes new possibilities which are not inherent in deterministic or probabilistic approaches. Moreover, in 1916 Alexander Ostrowski proved that any non-trivial absolute value on the rational numbers \mathbb{Q} is equivalent to either the usual real absolute value or a p -adic absolute value. So using p -adic numbers was the only remaining possibility not yet explored (Freivalds, 2013).

Ultrametric automata are similar to probabilistic automata but research has shown that the capabilities of these types of automata can differ very much. Ultrametric automata are able to recognize nonrecursive languages (Freivalds, 2013) and can have significant state complexity advantages over other types of automata (Balodis et al., 2013, Balodis, 2014). Ultrametric automata can also solve some tasks that have various requirements for computing complexity for Turing machines (Dimitrijevs et al., 2014, Krišlauks et al., 2013).

In this paper, we present a survey on ultrametric automata types and their language recognition capabilities. We begin with the definitions of p -adic numbers, operations with p -adic numbers and definitions of ultrametric automata. Then we list and analyse recent results about state complexity of ultrametric automata. In Section 2 we describe language recognition capabilities of ultrametric automata. Then, in Section 3 we describe other types of ultrametric automata and their capabilities. In summary section we list all results in concentrated form.

2 p -adic Numbers

A p -adic digit is a natural number between 0 and $p - 1$ where p is an arbitrary prime number. A p -adic integer $(a_i)_{i \in \mathbb{N}}$ is an infinite sequence of p -adic digits written from right to left. A p -adic integer can be written as a sequence of digits $\dots a_i \dots a_2 a_1 a_0$.

For each natural number, there exists its p -adic representation and only a finite number of p -adic digits are not zeroes. Negative integers have a different representation in p -adic numbers, namely, they have an infinite sequence of digits $p - 1$ to the left. If all digits of a p -adic integer are $p - 1$ then we have the p -adic number -1 . We can add, subtract and multiply p -adic integers in the same way as natural numbers in base p . The only division that is not possible in p -adic integers is division by p . For example, if we want to have p -adic integer $1/p$, equation $p * x = 1$ should have a solution, but multiplication by p -adic integer p gives zero in the right-most p -adic digit. That being said, p -adic integers can represent any integer and most of the rational numbers, except for those having a positive integral power of p in the denominator.

p -adic non-integers numbers can have a decimal point and are infinite to the left side but finite to the right side. For example, p -adic number $1/p$ can be written as $\dots 0000.1$. The field of p -adic numbers is denoted as \mathbb{Q}_p . For the curious reader, David A. Madore has written extensively about p -adic numbers and further information on the subject can be found in (Madore).

To measure p -adic number we need the absolute value of a p -adic number. If p is a prime number, then the p -adic ordinal of the rational number a , denoted by $ord_p a$, is the largest m such that p^m divides a .

Definition 1. For any rational number x its p -norm (p -adic absolute value) is

$$\|x\|_p = \begin{cases} 1/p^{\text{ord}_p x}, & \text{if } x \neq 0 \\ 0, & \text{if } x = 0. \end{cases}$$

For example, if $p = 7$, then $\|98\|_7 = \|2 * 7^2\|_7 = \|7^2\|_7 = 1/7^2 = 1/49$, if $p = 2$, then $\|98\|_2 = \|2\|_2 = 1/2$, but for any other prime number p , $\|98\|_p = 1$.

3 Definitions of Ultrametric Automata

Ultrametric automata are similar to probabilistic automata. Probabilistic finite automata were introduced by Michael O. Rabin, and the reader can refer to (Rabin) for more details about probabilistic automata.

Definition 2. A one-way p -ultrametric finite automaton is a tuple $(Q, S, \delta, q_0, F, \Lambda)$ where

- Q is a finite set of states,
- Σ is the input alphabet,
- $\delta : Q \times S \times Q \rightarrow \mathbb{Q}_p$ is the transition function,
- $q_0 : Q \rightarrow \mathbb{Q}_p$ is the initial amplitude distribution,
- $F \subseteq Q$ is the set of accepting states,
- $\Lambda = (\lambda, \diamond)$ is the acceptance condition where $\lambda \in \mathbb{R}$ is the acceptance threshold and $\diamond \in \{\geq, \leq\}$.

A probabilistic automaton has transition probabilities that are real numbers. In the case of p -ultrametric automaton the transitions have amplitudes, which are p -adic numbers. Therefore, we can assume that, for a p -ultrametric automaton, prime number p is also a parameter. Probabilistic automata have their initial distribution of probabilities among the states and transitions are performed with probabilities. In ultrametric automata every state has a initial amplitude, and by reading input word, transitions are done with amplitudes. This means that final amplitudes of the states are calculated in the same way as probabilities in probabilistic automata. To get the result after reading the input word, the amplitude of every accepting state is transformed into p -norm and the word is accepted if and only if sum of p -norms of accepting states satisfies the acceptance condition.

To make reader more familiar with denotions on the figures and working principles of ultrametric automata, we will provide an example with explanations. Consider the following language in binary alphabet: $L = \{0^n 1^m | n < m\}$. Ultrametric automaton which recognizes the language L is shown on Figure 1.

The big arrow on Figure 1 that points to state q_1 shows the initial amplitude of the state. Other state has the initial amplitude zero. The arrows between states (including arrows from state to itself) show the amplitudes of transitions. The arrow from q_1 to q_1 denotes, that when the automaton reads zero, the amplitude of the state q_1 is multiplied by two. The other arrow from state q_1 means the transition to state q_2 with amplitude $1/2$. Therefore, when the first letter 1 is read, the state q_2 gets the amplitude of state q_1

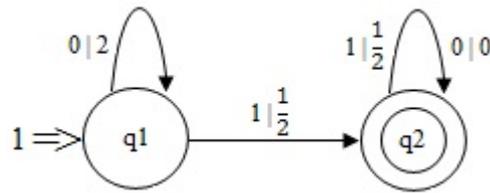


Fig. 1. Ultrametric automaton recognizing L .

multiplied by $1/2$, and amplitude of state $q1$ becomes zero. This automaton is 2-adic which means that all amplitudes are 2-adic numbers. The acceptance condition is ≥ 2 which means that when input is read the 2-norm of the amplitude of the state $q2$ (the accepting state) should be at least 2. It is true only for the words from L . If the input does not have ones or has any zero after any symbol 1, then the amplitude of $q2$ will be zero. Otherwise the amplitude of $q2$ will be 2^k , where $k = n - m$ for input 0^n1^m , 2-norm of 2^{n-m} is $\|2^{n-m}\|_2 = 2^{m-n}$, and this number will be at least 2 only if $m > n$.

It is also possible to represent the structure of ultrametric automaton with the help of transition matrices and vector of the initial amplitudes. This is also true for probabilistic automata, and sometimes it is easier to understand ultrametric automata with the help of these mathematical tools, it also show how ultrametric automata are similar to probabilistic automata. The matrix for the input symbol zero for automaton on Figure 1:

$$\begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}$$

For input symbol 1:

$$\begin{pmatrix} 0 & 1/2 \\ 0 & 1/2 \end{pmatrix}$$

The vector for initial amplitudes:

$$(1 \ 0)$$

Therefore, our automaton is $(Q = \{q1, q2\}, S = \{0, 1\}, \delta, q0 = (1, 0), F = \{q2\}, \Lambda = \{2, \geq\})$, where δ has transition matrices for both input symbols. Example of the action of our automata for input word "011":

$$(1 \ 0) \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1/2 \\ 0 & 1/2 \end{pmatrix} \begin{pmatrix} 0 & 1/2 \\ 0 & 1/2 \end{pmatrix} = (0 \ 1/2)$$

We obtain the amplitude $1/2$, whose 2-norm is 2, therefore, this input satisfies the acceptance condition.

Usage of all possible p -adic numbers in p -ultrametric automata is allowed. This was allowed in the first definition of ultrametric automata because Paavo Turakainen defined probabilistic finite automata where the "probabilities" can be arbitrary real numbers and he has proven that languages recognizable by these probabilistic finite automata are the same as for ordinary probabilistic finite automata. Ultrametric automata defined in this way have great capabilities, for example, they are able to recognize nonrecursive

languages (Freivalds, 2013). This is also the reason why more restricted versions of ultrametric automata were introduced.

Definition 3. Finite p -ultrametric automaton is called integral if all the p -adic numbers in its initial distribution and transition function are p -adic integers.

At the moment no examples of ultrametric integral automata recognizing nonrecursive languages are known. Now we will provide an example of ultrametric integral automaton, which recognizes the following unary language: $C_n = \{1^n\}$. Ultrametric integral automaton, which works like a counter, is shown on Figure 2.

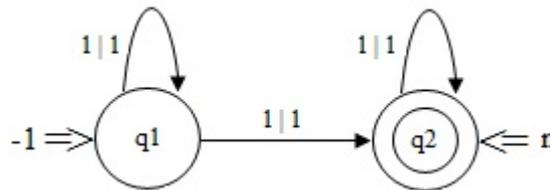


Fig. 2. Ultrametric integral automaton recognizing C_n .

In this case the automaton has the initial amplitude -1 in state $q1$ and n in the accepting state $q2$, and then each symbol read decreases amplitude of $q2$ by one. It is obvious that the amplitude of the accepting state will be zero only in the case of the word that must be accepted. Therefore, the acceptance condition for the p -norm will be ≤ 0 , because any non-zero rational number has positive amplitude.

Formally, the depicted automaton is the following: $(Q = \{q1, q2\}, S = \{1\}, \delta, q0 = (-1, n), F = \{q2\}, \Lambda = \{0, \leq\})$, where δ has the following transition matrix for the only symbol 1:

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

Example of the action of our automata for input word "111":

$$(-1 \ n) \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = (-1 \ n - 3)$$

In (Freivalds, 2013) even more restrictive definition was introduced.

Definition 4. A state of a p -ultrametric automaton is called regulated if there exist constants λ, c such that for every input word the p -norm of amplitude γ of this state is bounded by $\lambda - c < \|\gamma\|_p < \lambda + c$, and the total number of possible different amplitudes is limited. A finite p -ultrametric automaton is called regulated if all of its states are regulated.

Ultrametric regulated automata can recognize only regular languages (Freivalds, 2013). However, ultrametric regulated automata can have

much fewer states than deterministic and even probabilistic automata (Freivalds, 2013, Balodis et al., 2013).

Ultrametric automata with other acceptance conditions were also considered. Two results were published with ultrametric automata that do not have an acceptance threshold, but instead they have an acceptance interval which is represented by two real numbers (Dimitrijevs et al., 2014, Dimitrijevs, Ščeguļnaja, 2015). The results achieved in these papers are also achievable for ultrametric automata that are defined with acceptance threshold. On the other hand, with acceptance interval it is possible to reduce the number of required states. For example, Kaspars Balodis has proven, that ultrametric automaton requires at least 2 states to recognize the language $C_n = \{1^n\}$. In the case of acceptance interval it is possible to have one state, which is used like a counter, initial amplitude one, and multiplication of amplitude by p when an input symbol is read. Then the acceptance interval would be $[p^{-n}; p^{-n}]$, because $\|p^n\|_p = p^{-n}$. This allows to recognize the language C_n .

Rihards Krišlauks and Kaspars Balodis have also considered an other possible acceptance condition for ultrametric automata, defining them with accepting and rejecting states. The formal definition is the following (Krišlauks, Balodis, 2015):

Definition 5. A finite one-way p -ultrametric one-head automaton ($1u_pfa$ or $1u_pfa(1)$) is a sextuple $\langle S, \Sigma, s_0, \delta, Q_A, Q_R \rangle$ where

- S is a finite set—the set of states,
- Σ is a finite set ($\$ \notin \Sigma$)—input alphabet,
- $s_0 : S \rightarrow \mathbb{Q}_p$ is the initial amplitude distribution,
- $\delta : (\Sigma \cup \{\$\}) \times S \times S \rightarrow \mathbb{Q}_p$ is the transition function,
- $Q_A, Q_R \subseteq S$ are the sets of accepting and rejecting states, respectively.

The automaton works as follows: At every timestep, each of its states has an associated p -adic number called its amplitude. The automaton starts with an initial amplitude distribution s_0 . It subsequently proceeds by processing the input word's $w = w_1 \dots w_n$ symbols one at a time. The amplitude distribution after processing the i -th symbol is denoted as s_i , with $s_i(y) = \sum_{x \in S} s_{i-1}(x) \cdot \delta(w_i, x, y)$ for every $y \in S$. After the n -th symbol, the end marker $\$$ is similarly processed, obtaining the final amplitude distribution s_{n+1} . If the sum of the p -norms of final amplitudes over accepting states is greater than the sum of final amplitudes over rejecting states, i.e. if $\sum_{x \in Q_A} \|s_{n+1}(x)\|_p > \sum_{x \in Q_R} \|s_{n+1}(x)\|_p$, then the word w is said to be accepted, otherwise—rejected.

In the following sections we will consider ultrametric automata that are defined with an acceptance threshold. We will tell the reader in the case of considering other types of ultrametric automata.

4 State complexity

In this section we to summarize the results in recent publications about state complexity advantages of ultrametric finite automata. Most of the results show cases when ultrametric automata require much fewer states than deterministic, nondeterministic, alternating and probabilistic automata.

One of the first published results on the state complexity advantages of ultrametric automata was a comparison between deterministic and ultrametric regulated automata. In (Balodis et al., 2013) the following regular language was considered. Let $w = (w_1, w_2, \dots, w_m) \in \{0, 1, \dots, k-1\}^m$, and consider the following two operations:

1. a cyclic shift: $f_a(w_1, w_2, \dots, w_m) = (w_m, w_1, w_2, \dots, w_{m-1})$;
2. increasing the first element: $f_b(w_1, w_2, \dots, w_m) = ((w_1 + 1) \bmod k, w_2, \dots, w_m)$.

Let $x \in \{a, b\}^*$. Define $f_{x_1 x_2 \dots x_n}(w) = f_{x_n}(\dots f_{x_2}(f_{x_1}(w))\dots)$. The considered language is $L_{k,m} = \{x \in \{a, b\}^* \mid f_x(0^m) = 0^m\}$. Balodis et al. proved that a deterministic automaton requires at least k^m states. For all prime numbers p , an ultrametric automaton can recognize $L_{k,m}$ with $k \cdot m$ states. In this case it is possible to construct a regulated ultrametric automaton. Stronger results are achieved for every prime $p > m$: p -ultrametric automata can recognize $L_{p,m}$ with $m + 1$ states (Balodis et al., 2013). In this case the ultrametric automaton is not regulated.

Therefore, exponential state complexity advantages have been achieved. To conclude about the state complexity advantages of ultrametric regulated automata over deterministic automata, we have to mention the following two results. There is a proof that for any arbitrary prime number p there is a constant c_p such that if a language M is recognized by a regulated p -ultrametric finite automaton with k states, then there is a deterministic finite automaton with $(c_p)^{k \cdot \log k}$ states recognizing the language M (Balodis et al., 2013). Second, there is a proof that such a difference in state complexity is obtainable: for any arbitrary prime p there is a language, which is recognized by a p -ultrametric regulated automaton with $p + 2$ states, and this language requires at least $p! = c^{p \cdot \log p}$ states for the deterministic automaton to recognize this language (Freivalds, 2013).

Ultrametric integral automata have better capabilities than regulated ultrametric automata and we can expect greater state complexity advantages. A language $L_n = \{awbwa \mid w \in \{0, 1\}^* \text{ and } |w| = n\}$ was recently considered. One-way deterministic and even nondeterministic automata require at least 2^n states (Damanik, 1996). This language can be recognized with ultrametric integral automaton (for all prime numbers p) with constant state complexity. In Figure 3 such an ultrametric automaton is depicted (Dimitrijevs, 2016b). To construct integral this ultrametric automaton we have to choose a prime number q , which is not equal to p (therefore, $1/q$ remains p -adic integer). The automaton has four logical parts. The amplitude of $q3$ will be equal to zero if and only if the positions of the 1s are the same in both word parts w , the amplitude of state $q7$ will be zero if and only if the lengths of both parts w are equal. States $q8$, $q9$ and $q11$ will have an amplitude of zero if and only if the input word has a correct structure: b after a , the second letter a after b , and no letters present after the second letter a . States $q12$ and $q13$ ensure the check for equality $|w| = n$ (if $|w| = n$, state $q13$ will have amplitude zero).

The result was also improved by enhancing the language L_n . For every positive integer k there exists a language, which consists of words of length $O(n)$, requires at least k^n states to be recognized by a nondeterministic finite automaton, but for every prime number p , an integral p -ultrametric finite automaton can recognize this language with constant state complexity. Increasing the base of the exponent k by one increases the

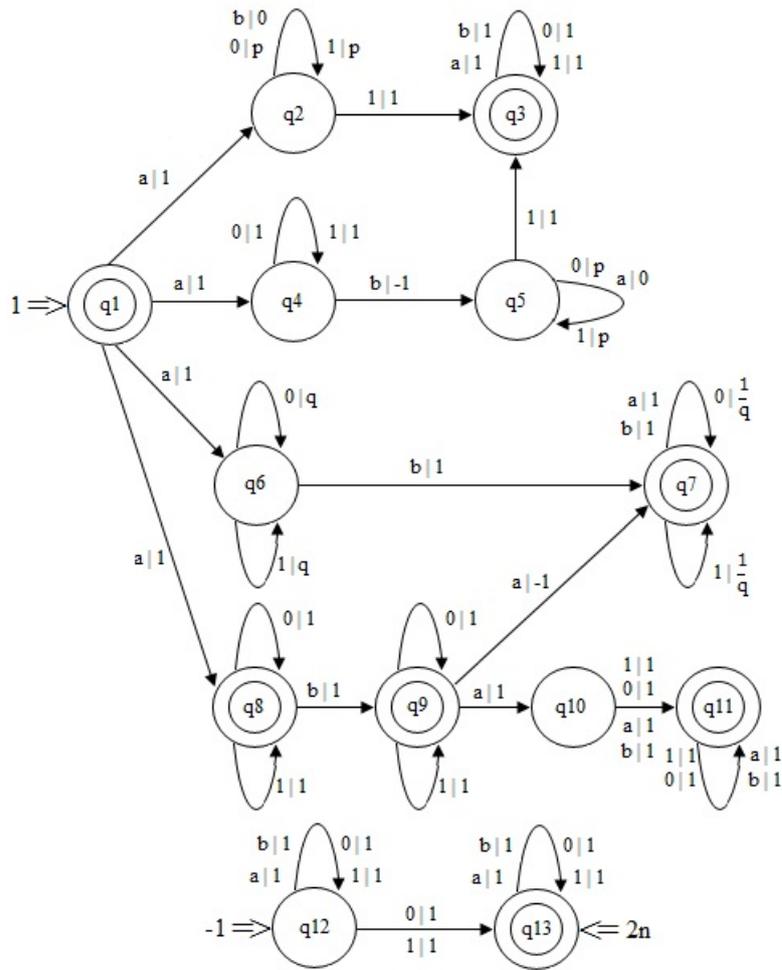


Fig. 3. A p -ultrametric automaton recognizing L_n .

required number of states for ultrametric integral automaton by 4 (Dimitrijevs, 2016b). The state complexity still remains a constant, not depending on the length of input word.

Next results show advantages of ultrametric automata over probabilistic automata. The paper (Balodis, 2014) contains results, where probabilistic automata require more states than p -ultrametric automata. Here the language is quite simple: $C_n = \{1^n\}$. A probabilistic automaton requires 3 states, while a p -ultrametric automaton requires 2 states. There exists a probabilistic automaton with 3 states and a p -ultrametric automaton with 2 states, both recognizing C_n . The strength of this result is in the fact that it is proven for all prime numbers p as parameters of p -ultrametric automaton.

In (Ambainis, 1996) the authors considered a language L_m with the m letter alphabet $\{a_1, a_2, \dots, a_m\}$ consisting of all words that contain each of the letters a_1, a_2, \dots, a_m exactly m times. There exists a probabilistic finite automaton with isolated cutpoint, which accepts L_m and has $O(m * (\log m)^2 / \log \log m)$ states. A deterministic finite automaton requires at least $(m + 1)^m$ states to recognize this language (Ambainis, 1996). For every prime number p , language L_m can be recognized by an integral p -ultrametric automaton with two states (Dimitrijevs, 2016b).

In the case of probabilistic automata with bounded error the difference can be more significant. In (Dimitrijevs, 2016b) the following language was considered, defined for all integers $k > 0$: $EVENODD_{yes}^k = \{a^{j2^k} \mid j \text{ is a nonnegative even integer}\}$. It is known that the language $EVENODD_{yes}^k$ requires at least 2^{k+1} states to be recognized by a one-way probabilistic automaton with bounded error (Ambainis, 2012). A two-way nondeterministic automaton also requires at least 2^{k+1} states (Say, Yakaryilmaz, 2014). It is possible to recognize the language $EVENODD_{yes}^k$ with 2-ultrametric automaton with two states (Dimitrijevs, 2016b).

It is worth to mention that the language $EVENODD_{yes}^k$ requires at least $k + 1$ states to be recognized by an alternating automaton (Geffert, 2014), therefore this gives us also an example of advantages over alternating automata.

5 Recognizable languages

The aim of this section is to show which languages can be recognized with different types of ultrametric automata, including ultrametric automata with limited number of states.

First, we remind the reader that regulated ultrametric automata can recognize only regular languages.

On the other hand, it is enough to have ultrametric automaton with one state to recognize nonregular language. The language $L_1 = \{x \mid x \in \{0, 1\}^* \text{ and } |x|_0 \geq |x|_1\}$, where $|x|_a$ denotes the number of symbols a in x , is mentioned in (Dimitrijevs, 2016a). L_1 can be easily shown to be nonregular based on the argument that the difference between the number of zeroes and the number of ones can increase infinitely. It is possible to construct a p -ultrametric automaton (for every prime number p) with one accepting state with initial amplitude 1. When the automaton reads symbol 1, it multiplies the amplitude by p . When the symbol zero is read, the amplitude is then multiplied by p^{-1} . After reading the input word the amplitude of the state will be $p^{|x|_1 - |x|_0}$. The p -norm of this number is equal to $1/p^{|x|_1 - |x|_0} = p^{|x|_0 - |x|_1}$. The automaton will accept the input

word x if and only if the p -norm of the amplitude is at least 1, and this is possible only when $|x|_0 \geq |x|_1$ (Dimitrijevs, 2016a).

The paper (Dimitrijevs, 2016a) has also a proof for the following theorem.

Theorem 1. *For every prime number p , the languages recognizable by p -ultrametric automata with one state form a proper subset of languages recognizable by one-way deterministic counter automata.*

Therefore, ultrametric automata with one state can surpass deterministic finite automata, but cannot surpass one-way deterministic automata with a counter.

In (Dimitrijevs, 2016a) is one more limitation for ultrametric automata with one state.

Theorem 2. *For every prime number p , a p -ultrametric integral automata with one state can recognize only regular languages.*

Therefore, only unrestricted ultrametric automata with one state can surpass deterministic automata. The next logical step is to consider context-sensitive languages that are not context-free (therefore, are not recognizable by nondeterministic push-down automata). In (Dimitrijevs, 2016a) the following non-context-free language was considered: $L_2 = \{x|x \in \{0, 1, 2\}^* \text{ and } |x|_0 < |x|_1 \text{ and } |x|_1 < |x|_2\}$. This language was recognized with unrestricted ultrametric automaton with two states. The next improvement has shown that non-context-free language can be recognized with integral ultrametric automaton with two states. On Figure 4 ultrametric integral automaton is depicted which recognizes the following non-context-free language: $L_3 = \{x|x = \{a, b, c, d\}^* \text{ and } |x|_a = |x|_b = |x|_c \text{ before the first symbol } d\}$ (Dimitrijevs, 2016a). The languages in both results can be recognized for any prime number p .

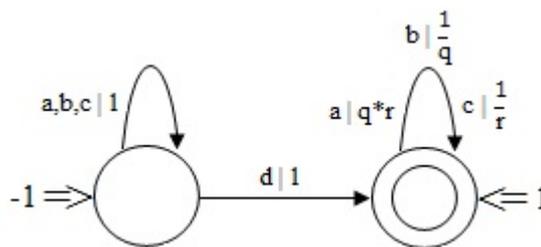


Fig. 4. Ultrametric automaton recognizing L_3 .

It turns out that two states are even enough to recognize nonrecursive languages. This is true for general ultrametric automata, and the question about the possibility to recognize nonrecursive languages with ultrametric integral automata is still open. Let $\beta = \dots a_3 a_2 a_1 a_0$ be an arbitrary p -adic number, which is not a p -adic integer, p be an arbitrary prime number and all $a_i = \{0, 1\}$. We define a language L_β in the following way: a binary sequence belongs to L_β if and only if it is equal to the last digits of β . p -adic numbers can only be finite to the right side from a decimal point. Assume that

number β has k p -adic digits after the decimal point. Ultrametric automaton which recognizes L_β can be seen on Figure 5 (Dimitrijevs, 2016a). This result also was achieved for all prime numbers p .

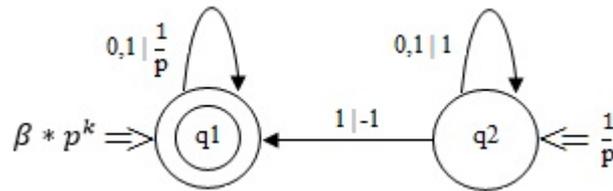


Fig. 5. Ultrametric automaton recognizing L_β .

In 2014 Maksims Dimitrijevs and Irina Ščeguļnaja have considered an other aspect of the languages, recognizable by ultrametric automata (Dimitrijevs et al., 2014). They considered languages that have some requirements on the computing complexity for deterministic Turing machines. Ultrametric automata are able to recognize some languages that require quadratic time complexity, logarithmic space complexity or linear reversal complexity for deterministic Turing machines. These results are also true for integral ultrametric automata and for all prime numbers p (Dimitrijevs, 2015).

6 Other types of automata

In this section two-way automata, multihead automata and pushdown automata are considered. We list recently achieved results for mentioned types of automata in the context of ultrametric automata.

In (Dimitrijevs, 2016b) binary palindromes were considered to show that two-way ultrametric automata in some cases require fewer states than one-way ultrametric automata. Two-way ultrametric integral automaton can recognize binary palindromes with three states, while one-way ultrametric automaton without restrictions requires at least four states. This is true for all prime numbers p . The following two figures show one-way and two-way ultrametric automata that recognize binary palindromes. On Figure 6 q is a prime number, which is not equal to p . This number was used in the paper to show how to construct ultrametric integral automaton.

In his seminal paper Rūsiņš Freivalds has shown that regulated two-way ultrametric automata can recognize non-regular languages (Freivalds, 2013). We follow with the exact text of the theorem.

Theorem 3. *For every prime $p \geq 3$ there exists a regulated 2-way finite integral p -ultrametric automaton recognizing the language $\{0^n 1^n\}$.*

This result is similar to the case of probabilistic automata with bounded error - two-way probabilistic automata are also able to recognize nonregular languages (Freivalds, 1982).

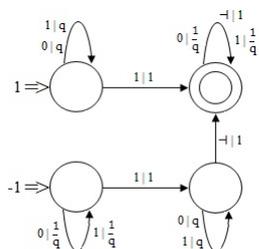


Fig. 6. A one-way p -ultrametric automaton recognizing palindromes.

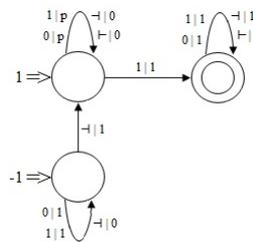


Fig. 7. A two-way p -ultrametric automaton recognizing palindromes.

Rūsiņš Freivalds has also published a result for regulated ultrametric pushdown automata (Freivalds, 2013). There exists language that can be recognized by 3-ultrametric regulated one-way pushdown automaton, but cannot be recognized by deterministic one-way pushdown automata and probabilistic one-way pushdown automata with bounded error.

Rihards Krišlauks and Kaspars Balodis considered the hierarchy of two-way ultrametric multihead automata (Krišlauks, Balodis, 2015). They considered the different definition of ultrametric automata, that have accepting and rejecting states. They have proven for all $k > 1$ that two-way ultrametric automaton with $k + 1$ heads can recognize languages, that cannot be recognized by two-way ultrametric automaton with k heads.

Maksims Dimitrijevs and Irina Ščeguļnaja have compared one-way ultrametric and nondeterministic multihead automata (Dimitrijevs, Ščeguļnaja, 2015). They have shown for all $k \geq 1$ that for all prime numbers p one-way nondeterministic automata with k heads can recognize languages, that form proper subset of languages, recognizable by one-way ultrametric automata with k heads.

7 Summary

Research of the past four years has shown some of the capabilities of ultrametric automata. Most of the research concentrate on state complexity advantages, the recognition power and different models of ultrametric automata. Most of results were proven for all prime numbers p .

Regulated ultrametric automata can have exponential state complexity advantages over deterministic automata. Ultrametric integral automata can have constant state complexity, while nondeterministic automata require exponential state complexity, where base of the exponent can be arbitrary large. Ultrametric automata can also have fewer states than probabilistic automata with unbounded error. For specific prime numbers p there exist languages that require constant number of states for ultrametric automata, but require exponential number of states for two-way nondeterministic and one-way probabilistic automata with bounded error. Ultrametric automata can also have constant state complexity when alternating automata require linear number of states.

Ultrametric automata with one state can recognize nonregular languages, but cannot surpass deterministic counter automata. Ultrametric integral automata with one state

can recognize only regular languages. Ultrametric integral automata with two states can recognize non-context-free languages, while unrestricted ultrametric automata with two states can recognize nonrecursive languages. Ultrametric integral automata are able to recognize some of the languages, that require quadratic time complexity, logarithmic space complexity or linear reversal complexity for deterministic Turing machines.

Two-way ultrametric automata can have fewer states than one-way ultrametric automata. Two-way regulated ultrametric automata can recognize nonregular languages, while one-way regulated ultrametric automata can't. There exists language that can be recognized by 3-ultrametric regulated one-way pushdown automaton, but cannot be recognized by deterministic one-way pushdown automata and probabilistic one-way pushdown automata with bounded error. For all $k > 1$ two-way ultrametric automaton with $k + 1$ heads can recognize languages, that cannot be recognized by two-way ultrametric automaton with k heads. For all $k \geq 1$ one-way nondeterministic automata with k heads can recognize languages, that form proper subset of languages, recognizable by one-way ultrametric automata with k heads.

References

- Ambainis, A. (1996). The complexity of probabilistic versus deterministic finite automata. In: Asano, T., Igarashi, Y., Nagamochi, H., Miyano, S., Suri, S. (eds.) *Algorithms and Computation, Lecture Notes in Computer Science*, vol. 1178, pp. 233-238. Springer Berlin Heidelberg, <http://dx.doi.org/10.1007/BFb0009499>
- Ambainis, A., Yakaryilmaz, A. (2012). Superiority of exact quantum automata for promise problems. *Information Processing Letters*, 112(7):289-291.
- Anashin, V. (2010). Non-Archimedean ergodic theory and pseudorandom generators. *The Computer Journal*, 53(4):370–392. doi: 10.1093/comjnl/bxm101
- Balodis, K., Beriņa, A., Čipola, K., Dimitrijevs, M., Iraids, J., Jēriņš, K., Kacs, V., Kalējs, J., Krišlauks, R., Lukstiņš, K., Raumanis, R., Somova, N., Ščeguļnaja, I., Vanaga, A., Freivalds, R. (2013). On the State Complexity of Ultrametric Finite Automata, 39th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2013), Špindlerův Mlýn, Czech Republic, January 26-31, 2013. *Proceedings, Volume 2*, pp. 1-9.
- Balodis, K. (2014). Counting with Probabilistic and Ultrametric Finite Automata, *Essays Dedicated to Jozef Gruska on the Occasion of His 80th Birthday, LNCS*, pp. 3-16.
- Damanik, D. (1996). Finite automata with restricted two-way motion. Master's thesis, J. W. GoetheUniversitaet Frankfurt (In German).
- Dimitrijevs, M. (2016). Capabilities of Ultrametric Automata with One, Two and Three States, 42nd International Conference on Current Trends in Theory and Practice of Computer Science, Harrachov, Czech Republic, January 23-28, 2016, *Proceedings*, pp. 253-264.
- Dimitrijevs, M. (2016). State Complexity Advantages of Ultrametric Automata, 42nd International Conference on Current Trends in Theory and Practice of Computer Science, Harrachov, Czech Republic, January 23-28, 2016, *Proceedings, Volume 2*, pp. 13-24.
- Dimitrijevs, M., Ščeguļnaja, I., Freivalds, R. (2014). Complexity Advantages of Ultrametric Machines, 40th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2014), Nový Smokovec, High Tatras, Slovakia, January 26-29, 2014. *Proceedings, Volume 2*, pp. 21-29.
- Dimitrijevs, M., Ščeguļnaja, I. (2015). Ultrametric Automata with One Head versus Multihead Nondeterministic Automata, 41st International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2015), Pec pod Sněžkou, Czech Republic, January 24–29, 2015. *Proceedings, Volume 2*, pp. 13-19.

- Dimitrijevs, M. (2015). Possibilities of ultrametric finite automata, Master's thesis, University of Latvia, Riga (In Latvian).
- Freivalds, R. (2013). Ultrametric automata and Turing machines, 17th International Conference on Developments in Language Theory (DLT 2013), June 18-21, 2013, Paris-Est, France, pp. 1-11.
- Freivalds, R. (1982). Probabilistic Two-Way Machines. In: Proceedings of the International Symposium on Mathematical Foundations of Computer Science, pp. 33-45.
- Geffert, V., Yakaryilmaz, A. (2014). Classical Automata on Promise Problems, Descriptive Complexity of Formal Systems (DCFS 2014), LNCS, pp. 126-137.
- Grigorchuk, R. I., Nekrashevich, V. V., Sushchanskii V. I. (2000). Automata, dynamical systems, and groups. Proc. Steklov Institute Math., 231:128-203.
- Kozyrev, S. (2006). Ultrametric analysis and interbasin kinetics, 2nd International Conference on p-Adic Mathematical Physics, American Institute of Physics, pp. 121-128.
- Krishnaswami, N. R., Benton, N., Hoffmann, J. (2012). Higher-order functional reactive programming in bounded space. In POPL'12, January 25-27, pages 45-58, Philadelphia PA.
- Krišlauks, R., Balodis, K. (2015). On the Hierarchy Classes of Finite Ultrametric Automata, 41st International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2015), Pec pod Sněžkou, Czech Republic, January 24-29, 2015. Proceedings, pp. 314-326.
- Krišlauks, R., Rukšāne, I., Balodis, K., Kucevalovs, I., Freivalds, R., Nāgele, I. (2013). Ultrametric Turing Machines with Limited Reversal Complexity, 39th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2013), Spindleruv Mlyn, Czech Republic, January 26-31, 2013. Proceedings, Volume 2, pp. 87-94.
- Lunts, A. G. (1965). The p-adic apparatus in the theory of finite automata. Problemy Kibernetiki, 14:17-30. In Russian.
- Madore, D. A. A first introduction to p-adic numbers. Available online: <http://www.madore.org/~david/math/padics.pdf>
- Murtagh, F. (2009). From data to the p-adic or ultrametric model. p-Adic Numbers, Ultrametric Analysis and Applications, 1:53-63.
- Perrin, D., Pin, J.-E. (2004). Infinite Words. Elsevier.
- Rabin, M. O. (1963). Probabilistic Automata, Information and Control 6(3), pp. 230-245.
- Say, A. C. C., Yakaryilmaz, A. (2014). Quantum finite automata: A modern introduction, Essays Dedicated to Jozef Gruska on the Occasion of His 80th Birthday, LNCS, pp. 208-222.
- Vladimirov, V. S., Volovich, I. V., Zelenov, E. I. (1995). P-Adic Analysis and Mathematical Physics, World Scientific.

Received September 2, 2016 , accepted November 12, 2016