Ultrametric Automata with One Head Versus Multihead Nondeterministic Automata*

Maksims Dimitrijevs and Irina Ščegulnaja

University of Latvia, Faculty of Computing,
Raiņa bulvāris 19, Riga, LV-1586, Latvia
mvdmaks@inbox.lv,irina.scegulnaja@gmail.com

Abstract. The idea of using p-adic numbers in Turing machines and finite automata to describe random branching of the process of computation was recently introduced. In the last two years some advantages of ultrametric algorithms for finite automata and Turing machines were explored. In this paper advantages of ultrametric automata with one head versus multihead deterministic and nondeterministic automata are observed.

1 Introduction

The idea of using p-adic numbers as parameters in finite automata belongs to Rūsiņš Freivalds. 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. So using p-adic numbers was rather the only remaining possibility not yet explored [1]. Rūsiņš Freivalds proved that use of p-adic numbers expose new possibilities which does not inhere in deterministic or probabilistic approaches. He stated that complexity of probabilistic automata and complexity of ultrametric automata can differ very much.

In this paper authors take a look at how can behave ultrametric algorithms in the context of multihead finite automata. The main goal of this paper is to show where ultrametric automata with one head can do better than multihead deterministic and nondeterministic automata. Some advantages of multihead ultrametric automata are also observed.

2 p-Adic Numbers

p-adic digit a_i is a natural number in 0 and $p-1$ where p is an arbitrary prime number. Infinite sequence of p-adic digits to the left side $(a_i)_{i \in N}$ is called *p-adic integer*. p-adic numbers are finite to the right side and infinite to the left side. For each natural x exists p-adic representation and only finite number of p-adic digits are not zeroes. There also exist p-adic float numbers.

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

Addition, subtraction, multiplication and division of p-adic numbers are done the same way like natural numbers. p-adic numbers are widely used in chemistry [2], molecular biology [3] and mathematics [4], and were first described in 1897 by Kurt Hensel. More about p-adic numbers and mathematical operations is written by David A. Madore in [5].

We will also need an *absolute value* or *norm*, which is a distance from zero. It's denoted by $||x||$ and has the following properties:

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

Norm is called ultrametric if $||x + y|| \leq max(||x||, ||y||)$.

If p is a prime number, then the *p-adic ordinal* of a, denoted by $ord_p a$, is the highest power of p which divides a. Accordingly for any rational number x, it's *p-norm* will be: $||x||_p = \begin{cases} 1/p^{ord_p x}, \text{if } x \neq 0 \\ 0, \text{if } x = 0 \end{cases}$ [6].

3 Ultrametric Automata

Ultrametric automata are described in more details by Rūsiņš Freivalds [1]. He had also introduced some specific definitions of ultrametric automata. Here we will describe some basics of ultrametric automata.

Most principles of the work of ultrametric automata are in common with probabilistic automata. Compared to deterministic automaton, probabilistic automaton has a *stochastic vector* (vector of probabilities) instead of one beginning state. Transitions also have probabilities and can be represented with the help of *stochastic matrices*. One matrix shows transitions for one specific letter of the input alphabet. Probabilistic automaton also has a *threshold*, and input word is accepted if and only if the sum of probabilities of every accepting state exceeds threshold after input word is read.

Ultrametric automaton has an additional element - prime number p. Probabilities of transitions are p-adic numbers, and are called *amplitudes*. Ultrametric automaton doesn't have limitation for beginning state vector and transition matrices to be stochastic. While probabilistic automaton has an accepting threshold, ultrametric automaton has an *accepting interval*, which is represented by two real numbers. After reading input word amplitude of every accepting state is transformed into p-norm. If sum of such p-norms belongs to an accepting interval, input word is accepted.

4 Multihead Automata

Multihead automata are similar to finite automata with one head. Transition function will be different - it will be in the following form: $S \times (A \cup \{\vdash, \dashv\})^k \rightarrow$

$S \times \{-1, 0, 1\}^k$ where k is number of heads, S is the set of states, A is an input alphabet, $\vdash$ is left end-marker, $\dashv$ is right end-marker. $\{-1, 0, 1\}^k$ means, that every one of k heads can:

- move one symbol to the left on the input word, if value is -1;
- stay on the same symbol of the input word, if value is 0;
- move one symbol to the right on the input word, if value is 1.

If head is on the left end-marker it can't move to the left, and, respectively, it can't move to the right, if it is on the right end-marker. Here were mentioned main principles of how differs two-way multihead deterministic finite automaton from two-way deterministic finite automaton with one head. In the case of one-way automata head can't move to the left. More precise definition can be found in [7].

In the case of nondeterministic and ultrametric finite automata definition will differ with the fact that we have to deal with multiple heads. So, the only difference will be in the transition function. It will be like transition function of deterministic multihead automata, but transitions will be made with amplitudes.

5 Advantages of Multihead Ultrametric Automata

It is known that one-way deterministic finite automata with k heads are weaker (can recognize fewer languages) than one-way nondeterministic finite automata with k heads for $k \geq 2$ [7]. Using these results we can prove theorem for multihead ultrametric automata.

Theorem 1. *For every prime number p for every $k \geq 1$ set of languages recognized by one-way deterministic finite automata with k heads is a proper subset of the set of languages, that can be recognized by one-way p-ultrametric finite automata with k heads.*

Proof. *For $k = 1$ let's take language $L_1 = \{w2w^{rev}2 | w \in \{0, 1\}^*\}$ where w^{rev} consists of the same letters, as w, but in opposite order. In [8] it was proven, that for every prime p we can construct one-way p-ultrametric automaton with one head that can recognize the language L_1. Language is not regular, so it can't be recognized by one-way deterministic finite automaton with one head. On the other hand, ultrametric automata can recognize all the languages that can be recognized by deterministic automata.*

We can transform any nondeterministic automaton into p-ultrametric. In this case we will keep all states and transitions of nondeterministic automaton, all beginning states will have amplitude 1, all transitions will be made with amplitude 1 and input word will be accepted by ultrametric automaton if p-norm sum of all accepting states will be greater than 0. Resulting ultrametric automaton does not depend on parameter p. This means that one-way ultrametric automata with k heads are at least as powerful as one-way nondeterministic automata with k heads. For every $k \geq 2$ for every prime number p the set of languages recognizable by one-way deterministic finite automata with k heads is a proper subset *of set of languages recognizable by one-way p-ultrametric finite automata with k heads.*

From parts of proof for $k = 1$ and $k \geq 2$ we obtain that for automata with k heads condition holds for every $k \geq 1$.

Now let's take a look at situation where we can reduce the number of heads for ultrametric automata compared to deterministic ones.

Theorem 2. *For every prime number p for every $k \geq 1$ there is a language which can be recognized by one-way p-ultrametric finite automaton with 2 heads and cannot be recognized by one-way deterministic finite automaton with k heads.*

Proof. *In [9] it is shown that there is a language L which can be recognized by one-way nondeterministic finite automaton with two heads and cannot be recognized by one-way deterministic finite automaton with k heads for any k. Like in the proof of the first theorem we can replace nondeterministic automaton with p-ultrametric automaton for any prime number p.*

6 Power of Ultrametric Automata with One Head

There is a proof that for one-way deterministic and nondeterministic finite automata class of languages recognizable by automaton with $k + 1$ head is wider than class of languages recognizable by automaton with k heads, for all natural $k \geq 1$ [9]. This was proven by using language L_b, which is defined for all positive natural numbers b in the following way: $L_b = \{w_1 * w_2 * ... * w_{2b} | (w_i \in \{0,1\}^*) \wedge (w_i = w_{2b+1-i})\ for\ all\ 1 \leq i \leq 2b\}$. This was proven with the help of theorem which says, that one-way deterministic and nondeterministic finite automaton with k heads recognizes language L_b if and only if $b \leq \binom{k}{2}$ [9].

Theorem 3. *For every integer $k \geq 1$ for every prime number p there exist languages that cannot be recognized by one-way deterministic or nondeterministic automaton with k heads, but can be recognized by one-way p-ultrametric finite automaton with one head.*

Proof. *We will construct 2-adic automaton that recognizes language L_b. For each $1 \leq i \leq b$ we will make set of states which will check equality of the corresponding pair of fragments, that is, if $w_i = w_{2b+1-i}$. Fragment for one such pair is shown on fig. 1.*

In the states of bottom row on fig. 1 there is a check for fragment length equality. Beginning amplitude is 1. When automaton is reading fragment w_i every symbol of the fragment multiplies amplitude by 2. Then automaton goes further on the row of states until it will reach fragment w_{2b+1-i}. While reading this fragment every symbol will divide amplitude by 2. Amplitude will be equal to 1 if and only if both fragments had same length. Then after reading whole input word -1 will be added to amplitude, so amplitude of accepting state will be equal to 0 if condition of fragment equality holds.

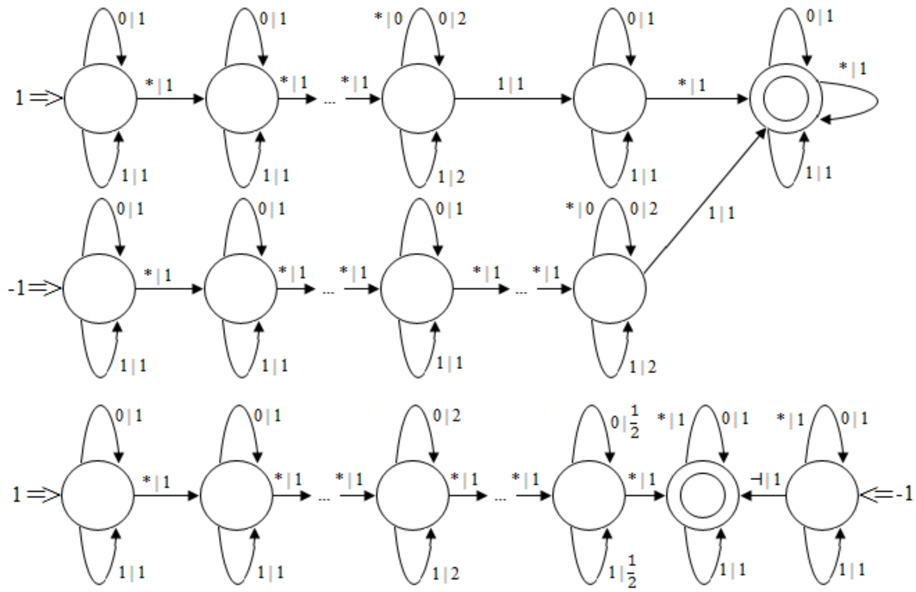

Fig. 1. Fragment of 2-adic automaton for language's L_b recognition

Upper two rows on fig. 1 ensure check for matching of symbol "1" positions in corresponding fragments w_i and w_{2b+1-i}. In the first row automaton walks through states until it reaches fragment w_i. Then in the next step symbols "1" and their positions are counted, and amplitude after reading fragment gets equal to $\sum_{i=0}^{n-1} a_i 2^i$, where n is length of the fragment and a_i equals to 1 if i-th symbol was 1, else a_i equals to 0. After reading fragment sum is stored in the accepting state. Second row of states in the similar way ensures finding of fragment w_{2b+1-i} and then from the accepting state $\sum_{i=0}^{n-1} a_i 2^i$ is being subtracted in the similar way. As a result amplitude of accepting state will be equal to 0 if and only if positions of symbol "1" in both fragments were the same.

If both conditions hold after reading the input word (that means, if fragments w_i and w_{2b+1-i} are equal), then both accepting states will have amplitude 0. So p-norm sum of accepting states will be equal to 0 if and only if $w_i = w_{2b+1-i}$. Else p-norm sum of accepting states will be equal to some positive number.

For every fragment pair w_i and w_{2b+1-i} same checks for equality will be made, but in the sets of states will be different amount of states that ensure finding of the fragment that we are interested in. In the similar way p-norm sum of accepting states will be equal to 0 if and only if fragments of pair are equal. Else mentioned p-norm sum will be equal to some positive number. By counting together p-norms of accepting states of all sets of that kind we will get 0 if and *only if input word belongs to L_b. In this way we can construct 2-adic automaton to recognize language L_b for every positive integer b. By taking b too big ($b > \binom{k}{2}$ respectively) one-way deterministic or nondeterministic finite automaton with k heads won't be able to recognize language L_b.*

To prove a theorem for all prime numbers p we can replace amplitude 2 with p in all places where we were multiplying or dividing by 2. So the theorem holds for every prime number p.

In fact we can easily expand the set of the languages that can be recognized by one-way ultrametric finite automaton with one head and cannot be recognized by one-way deterministic or nondeterministic automaton with fixed arbitrary amount of heads. When defining languages we can take as base language L_b, but this time every fragment w_i will consist of n letter alphabet instead of two letter alphabet $\{0, 1\}$. Situation will be similar to language L_b, but this time p-ultrametric automaton will have to check positions of $n - 1$ symbols of input alphabet instead of one symbol (in the example of language L_b such symbol was "1"). This can result in the increase of number of states.

Results obtained in this chapter and possibility to transform nondeterministic automaton with k heads into p-ultrametric automaton with k heads for any prime number p can give us another result about hierarchy of multihead automata.

Theorem 4. *For every $k \geq 1$ for every prime number p the set of languages recognizable by one-way nondeterministic finite automata with k heads is a proper subset of the set of languages recognizable by one-way p-ultrametric finite automata with k heads.*

Proof. *In the proof of Theorem 1 we can see that for every number of heads $k \geq 1$ one-way ultrametric automata are at least as powerful, as one-way nondeterministic automata. We can use Theorem 3 to say that for every prime number p for every number of heads $k \geq 1$ one-way p-ultrametric automaton will be able to recognize some languages that cannot be recognized by one-way nondeterministic automata.*

7 Summary

Ultrametric automata can have better place in the hierarchy of multihead automata than nondeterministic automata. All begins with the fact that one head of ultrametric automaton can do more than one head of deterministic or nondeterministic automaton. For any fixed number of heads k one-way k-head ultrametric automata are stronger than one-way k-head deterministic and nondeterministic automata.

In some cases ultrametric automata with one head can recognize languages better than multihead deterministic and nondeterministic automata. The research made shows potential of the heads of ultrametric automata and researches will be continued to find the place of ultrametric automata in the hierarchy of multihead automata.

References

1. Freivalds, R.: Ultrametric automata and Turing machines. 17th International Conference on Developments in Language Theory (DLT 2013), Paris, France (June 2013)
2. Khrennikov, A.Y.: NonArchimedean Analysis: Quantum Paradoxes, Dynamical Systems and Biological Models. Kluwer Academic Publishers (1997)
3. Dragovich, B. and Dragovich, A.: A p-adic Model of DNA Sequence and Genetic Code. p-adic Numbers, Ultrametric Analysis, and Applications, vol. 1, No 1, 34-41 (2009)
4. Vladimirov, V.S., Volovich, I.V. and Zelenov, E.I.:p-Adic Analysis and Mathematical Physics. World Scientific (1995)
5. Madore, D.A.: A first introduction to p-adic numbers. `http://www.madore.org/david/math/padics.pdf` (revised 2000)
6. Koblitz, N.: p-adic Numbers, p-adic Analysis, and Zeta-Functions. Springer, 2nd edition (1984)
7. Holzer, M., Kutrib, M., Malcher, A.: Multi-Head Finite Automata: Characterizations, Concepts and Open Problems. Graduate Texts in Mathematics, vol. 58. `http://arxiv.org/pdf/0906.3051v1.pdf` (2008)
8. Dimitrijevs, M., Ščeguļnaja, I., Freivalds, R.: Complexity Advantages of Ultrametric Machines, 40th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2014), Nový Smokovec, Vysoké Tatry, Slovakia (2014)
9. Yao, A.C., Rivest, R.L.: $k + 1$ Heads are better than k. Massachusetts Institute of Technology, Cambridge, Massachusetts (1977)