TCS-TR-A-**13-68**

TCS Technical Report

Active Learning of Classes of Recursive Functions by Ultrametric Algorithms

by

Rūsiņš Freivalds and Thomas Zeugmann

Division of Computer Science

Report Series A

October 24, 2013

Hokkaido University
Graduate School of
Information Science and Technology

Email: thomas@ist.hokudai.ac.jp

Phone: +81-011-706-7684
Fax: +81-011-706-7684

Active Learning of Classes of Recursive Functions by Ultrametric Algorithms*

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

Thomas Zeugmann
Division of Computer Science
Hokkaido University
N-14, W-9
Sapporo 060-0814, Japan
thomas@ist.hokudai.ac.jp

Abstract

We study active learning of classes of recursive functions by asking value queries about the target function f, where f is from the target class. That is, the query is a natural number x, and the answer to the query is $f(x)$. The complexity measure in this paper is the worst-case number of queries asked. We prove that for some classes of recursive functions *ultrametric active learning* algorithms can achieve the learning goal by asking *significantly fewer* queries than deterministic, probabilistic, and even nondeterministic active learning algorithms. This is the first ever example of a problem, where ultrametric algorithms have advantages over nondeterministic algorithms.

1. Introduction

In traditional studies of learning of recursive functions [10, 13, 42, 43], concepts are modeled as a set of stimulus and response pairs. Assuming that any concept associates only one response with each possible stimulus, it can be viewed as a function from stimuli to responses. Every string of ASCII symbols can be encoded in the natural numbers. These strings include arbitrarily long finite texts and are certainly sufficient to express both stimuli and responses. So for a mathematical treatment of learning, it suffices to consider only the functions from natural numbers to natural numbers. Via suitable encodings, these functions can represent a wide range of phenomena. Since we shall be concerned only with effective learning procedures, the recursive functions are used to model concepts.

*The research was supported by Grant No. 09.1570 from the Latvian Council of Science and the Invitation Fellowship for Research in Japan S12052 by Japan Society for the Promotion of Science

Inductive inference, the abstract study of generalization, has been studied intensively. In Gold's [18] model of *identification in the limit*, the learner is a deterministic algorithm called *inductive inference machine* (abbr. IIM), and the objects to be learned are recursive functions. The source of information are growing initial segments $(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))$ of ordered pairs of the graph of the target function f, where it is assumed that every pair $(x, f(x))$ appears eventually. As a hypothesis space one can choose any *Gödel numbering* $\varphi_0, \varphi_1, \varphi_2, \ldots$ of the set of all partial recursive functions over the natural numbers (cf. Rogers [38]). If a number i is such that $\varphi_i = f$ then we call i a *φ-program* of f. An IIM, on input an initial segment $(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))$, has to output a natural number i_n which is interpreted as φ-program. An IIM *identifies* f if the sequence $(i_n)_{n\in\mathbb{N}}$ of all computed φ-programs converges to a program i such that $\varphi_i = f$. Here $\mathbb{N} = \{0, 1, 2, \ldots\}$ denotes the set of all natural numbers. If an IIM identifies some function f, then some form of learning must have taken place, since, by the properties of convergence, only a finite part of the graph of f was known by the IIM at the (unknown) point of convergence. The terms *infer* and *learn* will be used as synonyms for *identify*.

Every IIM M learns some set of recursive functions which we denote by $EX(M)$. The family of all such sets, over the universe of effective algorithms viewed as IIMs, serves as a characterization of the learning power inherent in the Gold model. This family is symbolically denoted by EX (short for explanatory) and it is defined rigorously by $EX = \{\mathcal{U} \mid \exists M(\mathcal{U} \subseteq EX(M)\}$. In many studies of inductive inference the family EX is set-theoretically compared with the families that arise from considering other models. In this way, one parameter of the learning process can be examined in isolation (cf., e.g., [43] and the references therein). There are several well studied derivatives of the Gold [18] model. One such derivative is *finite learning*, where the IIM either requests a new input and outputs nothing, or it outputs a program i, and stops. Again we require that program i is correct for f, i.e., $\varphi_i = f$.

The models described so far are models of *passive* learning, since the IIM has no influence on the order in which examples are presented. In contrast, the learning model considered in the present paper is an *active* one. This model goes back to Angluin [3] and is called *query learning*. In the query learning model the learner has access to a teacher that truthfully answers queries of a prespecified type. In this paper we only consider *value queries*. That is, the query is a natural number x, and the answer to the query is $f(x)$. A query learner is an algorithmic device that, depending on the answers already received, either computes a new value query or it returns a hypothesis i and stops. As above, the hypothesis is interpreted with respect to a fixed Gödel numbering φ and it is required that the hypothesis returned satisfies $\varphi_i = f$. So active learning is *finite learning*.

As in the Gold [18] model, we are interested in active learners that can infer whole classes of recursive functions. The complexity measure is then the *worst-case number of queries asked* to identify all the functions from the target class $\mathcal{U}$. We refer to any query learner as *query inference machine* (abbr. QIM).

Note that value queries are just the simplest type of query. Various authors studied many different query languages extending this simplest type of query (cf., e.g., [15, 17, 16, 26]).

Automata theory and complexity theory have considered several natural generalizations of deterministic algorithms, namely, nondeterministic and probabilistic algorithms. In many cases these generalized algorithms allow for computations having a complexity that is strictly less than their deterministic counterpart. Such generalized algorithms attracted considerable attention in learning theory, too. Many papers studied learnability by nondeterministic algorithms [1, 8, 14, 41] and probabilistic algorithms [20, 25, 31, 32, 33, 36, 37].

In the following we use $\mathcal{P}$, $\mathcal{R}$ and $\mathcal{P}^2$, $\mathcal{R}^2$ to denote the set of all partial recursive functions and of all recursive functions of one respectively two variables over $\mathbb{N}$, respectively. Furthermore, we formally define what a Gödel numbering is. Any function $\psi \in \mathcal{P}^2$ is called a *numbering*. Moreover, let $\psi \in \mathcal{P}^2$, then we write ψ_i instead of $\lambda x.\psi(i,x)$ and set $\mathcal{P}_\psi = \{\psi_i \mid i \in \mathbb{N}\}$ as well as $\mathcal{R}_\psi = \mathcal{P}_\psi \cap \mathcal{R}$. A numbering $\varphi \in \mathcal{P}^2$ is called a *Gödel numbering* (cf. Rogers [38]) iff $\mathcal{P}_\varphi = \mathcal{P}$, and for any numbering $\psi \in \mathcal{P}^2$, there is a *compiler* $c \in \mathcal{R}$ such that $\psi_i = \varphi_{c(i)}$ for all $i \in \mathbb{N}$.

Definition 1. *We say that a nondeterministic QIM* learns a function f *if*

(1) *there is at least one computation path such that the QIM produces a correct result on f, i.e., program j such that $\varphi_j = f$;*

(2) *at no computation path the QIM produces an incorrect result on f.*

Probabilistic algorithms can be represented by rooted trees (cf. Hromkovič [21]). The leaves of the tree are the output nodes and the probability to reach a leaf is computed in the usual way.

Definition 2. *We say that a probabilistic QIM* produces a result m *with a* probability p *if the sum of the probabilities of all leaves which correctly produce the result m is no less than p.*

Definition 3. *We say that a probabilistic QIM* learns a function f with a probability p *if*

(1) *the sum of all probabilities of all leaves which produce a correct result on f, i.e., a number j such that $\varphi_j = f$, is no less than p.*

(2) *at no computation path the QIM produces an incorrect result on f.*

Recently, Freivalds [11] introduced a new type of indeterministic algorithms called *ultrametric* algorithms. An extensive research on ultrametric algorithms of various kinds is currently performed by him and his co-authors (cf. [4, 23]). So, ultrametric algorithms are a very new concept and their potential still has to be explored. This is the first paper showing a problem, where ultrametric algorithms *have advantages* over nondeterministic algorithms. Ultrametric algorithms are very similar to probabilistic algorithms but while probabilistic algorithms use *real* numbers r with $0 \leq r \leq 1$ as parameters, ultrametric algorithms use *p-adic* numbers as the parameters. As it turns out the usage of p-adic numbers as *amplitudes* and the ability to perform *measurements* to transform amplitudes into real numbers are inspired by quantum computations and allow for algorithms not possible in classical computations. Slightly simplifying the description of the definitions, one can say that ultrametric algorithms are the same as probabilistic algorithms, only the *interpretation* of the probabilities is *different*.

The choice of p-adic numbers instead of real numbers is not quite arbitrary. Ostrowski [35] proved that any non-trivial absolute value on the rational numbers $\mathbb{Q}$ is equivalent to either the usual real absolute value or a p-adic absolute value. This result shows that using p-adic numbers was not merely one of many possibilities to generalize the definition of deterministic algorithms but rather the only remaining possibility not yet explored.

The notion of p-adic numbers is widely used in science. String theory [40], chemistry [28] and molecular biology [9, 24] have introduced p-adic numbers to describe measures of indeterminism. Indeed, research on indeterminism in nature has a long history. Pascal and Fermat believed that every event of indeterminism can be described by a real number between 0 and 1 called *probability*. Quantum physics introduced a description in terms of complex numbers called *amplitude of probabilities* and later in terms of probabilistic combinations of amplitudes most conveniently described by *density matrices*. Using p-adic numbers to describe indeterminism allows to explore some aspects of indeterminism but, of course, does not exhaust all the aspects of it.

There are many distinct p-adic absolute values corresponding to the many prime numbers p. These absolute values are traditionally called *ultrametric*. Absolute values are needed to consider *distances* among objects. We are used to rational and irrational numbers as measures for distances, and there is a psychological difficulty to imagine that something else can be used instead of rational and irrational numbers, respectively. However, there is an important feature that distinguishes p-adic numbers from real numbers. Real numbers (both rational and irrational) are linearly ordered, while p-adic numbers *cannot* be linearly ordered. This is why *valuations* and *norms* of p-adic numbers are considered.

The situation is similar in Quantum Computation (see [34]). 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. To make the paper as self-contained as possible, we continue with a short description of p-adic numbers.

2. p-adic Numbers and Ultrametric Algorithms

Let p be an arbitrary prime number. A number $a \in \mathbb{N}$ with $0 \le a \le p-1$ is called a *p-adic digit.* A *p-adic integer* is by definition a sequence $(a_i)_{i\in\mathbb{N}}$ of p-adic digits. We write this conventionally as $\cdots a_i \cdots a_2 a_1 a_0$, i.e., the a_i are written from left to right.

If n is a natural number, and $n = \overline{a_{k-1}a_{k-2}\cdots a_1 a_0}$ is its p-adic representation, i.e., $n = \sum_{i=0}^{k-1} a_i p^i$, where each a_i is a p-adic digit, then we identify n with the p-adic integer (a_i), where $a_i = 0$ for all $i \ge k$. This means that the natural numbers can be identified with the p-adic integers $(a_i)_{i\in\mathbb{N}}$ for which all but finitely many digits are 0. In particular, the number 0 is the p-adic integer all of whose digits are 0, and 1 is the p-adic integer all of whose digits are 0 except the right-most digit a_0 which is 1.

To obtain p-adic representations of all rational numbers, $\frac{1}{p}$ is represented as $\cdots 00.1$, the number $\frac{1}{p^2}$ as $\cdots 00.01$, and so on. For any p-adic number it is allowed to have infinitely many (!) digits to the left of the "p-adic" point but only a finite number of digits to the right of it.

However, p-adic numbers are not merely a generalization of rational numbers. They are related to the notion of *absolute value* of numbers.

If X is a nonempty set, a distance, or metric, on X is a function d from $X \times X$ to the nonnegative real numbers such that for all $(x, y) \in X \times X$ the following conditions are satisfied.

(1) $d(x, y) \ge 0$, and $d(x, y) = 0$ if and only if $x = y$,

(2) $d(x, y) = d(y, x)$,

(3) $d(x, y) \le d(x, z) + d(z, y)$ for all $z \in X$.

A set X together with a metric d is called a *metric space*. The same set X can give rise to many different metric spaces. If X is a linear space over the real numbers then the *norm* of an element $x \in X$ is its distance from 0, i.e., for all $x, y \in X$ and α any real number we have:

(1) $\|x\| \ge 0$, and $\|x\| = 0$ if and only if $x = 0$,

(2) $\|\alpha \cdot y\| = |\alpha| \cdot \|y\|$,

(3) $\|x + y\| \le \|x\| + \|y\|$.

Note that every norm induces a metric d, i.e., $d(x, y) = \|x - y\|$. A well-known example is the metric over $\mathbb{Q}$ induced by the ordinary absolute value. However, there are other norms as well.

A norm is called *ultrametric* if Requirement (3) can be replaced by the stronger statement: $\|x + y\| \le \max\{\|x\|, \|y\|\}$. Otherwise, the norm is called *Archimedean*.

Definition 4. *Let $p \in \{2, 3, 5, 7, 11, 13, \ldots\}$ be any prime number. For any nonzero integer a, let the p-adic* ordinal *(or* valuation*) of a, denoted $ord_p\, a$, be the highest power of p which divides a, i.e., the greatest number $m \in \mathbb{N}$ such that $a \equiv 0 \pmod{p^m}$. For any rational number $x = a/b$ we define $ord_p\, x =_{df} ord_p\, a - ord_p\, b$. Additionally, $ord_p\, x =_{df} \infty$ if and only if $x = 0$.*

For example, let $x = 63/550 = 2^{-1} \cdot 3^2 \cdot 5^{-2} \cdot 7^1 \cdot 11^{-1}$. Thus, we have

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

Definition 5. *Let $p \in \{2, 3, 5, 7, 11, 13, \ldots\}$ be any prime number. For any rational number x, we define its p-*norm *as $p^{-ord_p\, x}$, and we set $\|0\|_p =_{df} 0$.*

For example, with $x = 63/550 = 2^{-1}3^2 5^{-2} 7^1 11^{-1}$ we obtain:

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

Rational numbers are p-adic integers for all prime numbers p. Since the definitions given above are all we need, we finish our exposition of p-adic numbers here. For a more detailed description of p-adic numbers we refer to [19, 27].

We continue with *ultrametric algorithms*. In the following, p always denotes a prime number. Ultrametric algorithms are described by finite directed acyclic graphs (abbr. DAG), where exactly one node is marked as root. As usual, the root does not have any incoming edge. Furthermore, every node having outdegree zero is said to be a *leaf*. The leaves are the output nodes of the DAG.

Let v be a node in such a graph. Then each outgoing edge is labeled by a p-adic number which we call *amplitude*. We require that the sum of all amplitudes that correspond to v sums up to 1. In order to determine the *total amplitude* along a computation path, we need the following definition.

Definition 6. *The total amplitude of the root is defined to be 1. Furthermore, let v be a node at depth d in the DAG, let α be its total amplitude, and let $\beta_1, \beta_2, \cdots, \beta_k$ be the amplitudes corresponding to the outgoing edges $e_1, \ldots, e_k$ of v. Let $v_1, \ldots, v_k$ be the nodes where the edges $e_1, \ldots, e_k$ point to. Then the total amplitude of v_ℓ, $\ell \in \{1, \ldots, k\}$, is defined as follows.*

(1) *If the indegree of v_ℓ is one, then its total amplitude is $\alpha\beta_\ell$.*

(2) *If the indegree of v_ℓ is bigger than one, i.e., if two or more computation paths are joined, say m paths, then let $\alpha, \gamma_2, \ldots, \gamma_m$ be the corresponding total amplitudes of the predecessors of v_ℓ and let $\beta_\ell, \delta_2, \ldots, \delta_m$ be the amplitudes of the incoming edges The total amplitude of the node v_ℓ is then defined to be $\alpha\beta_\ell + \gamma_2\delta_2 + \cdots + \delta_m\gamma_m$.*

Note that the total amplitude is a p-adic integer.

We refer the reader to the proof of Theorem 9 for an example.

It remains to define what is meant by saying that a p-ultrametric algorithm produces a result with a certain probability. This is specified by performing a so-called *measurement* at the leaves of the corresponding DAG. Here by measurement we mean that we transform the total amplitude β of each leaf to $\|\beta\|_p$. We refer to $\|\beta\|_p$ as the *p-probability* of the corresponding computation path.

Definition 7. *We say that a p-ultrametric algorithm* produces a result m with a probability q *if the sum of the p-probabilities of all leaves which correctly produce the result m is no less than q.*

Definition 8. *We say that a p-ultrametric QIM* learns a function f with a probability q *if*

(1) *the sum of the* p-probabilities *of all leaves which produce a correct result on f, i.e., a number j such that $\varphi_j = f$ is no less than q,*

(2) *at no computation path the QIM produces an incorrect result on f.*

3. Results

As explained in the Introduction we are interested in the number of queries a QIM has to ask in the worst-case in order to infer all recursive functions from a prespecified class $\mathcal{U}$. In order to study this problem we shall always assume the hypothesis space to be a Gödel numbering φ (cf. [38]). This is no restriction of generality since all natural programming languages provide Gödel numberings of recursive functions.

The complexity of learning recursive functions has been an important topic for several decades [2, 12, 13, 29, 42, 43]. Furthermore, the problem to learn functions from value queries has been intensively studied in various domains. We refer the reader to Bshouty [7] for an overview. In the present paper we compare the query complexity of deterministic, nondeterministic, probabilistic, and ultrametric QIMs to one another.

Our results are somewhat unexpected. Usually, for various classes of problems, nondeterministic algorithms provide the smallest complexity, deterministic algorithms provide the largest complexity and probabilistic algorithms provide some medium complexity. In [4, 11, 23] ultrametric algorithms also gave medium complexity sometimes better and sometimes worse than probabilistic algorithms. Our results in this paper show that, for learning recursive functions from value queries, there are classes $\mathcal{U}$ of recursive functions such that ultrametric QIMs have a smaller query complexity than even nondeterministic QIMs. This is the first ever example of a problem, where ultrametric algorithms have advantages over nondeterministic algorithms.

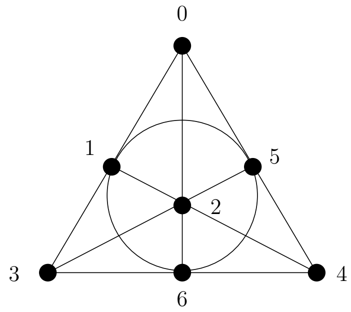

Figure 1: The Fano Plane

To show these results we use a combinatorial structure called the *Fano plane*. It is one of *finite geometries* (cf. Meserve [30]). The Fano plane consists of seven *points* $0, 1, 2, 3, 4, 5, 6$ and seven *lines* $(0, 1, 3)$, $(1, 2, 4)$, $(2, 3, 5)$, $(3, 4, 6)$, $(4, 5, 0)$, $(5, 6, 1)$, $(6, 0, 2)$. For any two points i, j with $i \neq j$, in this geometry there is exactly one line that contains these points (cf. Figure 1). For any two different lines in this geometry there is exactly one point contained in these two lines. In our construction the points $0, 1, 2, 3, 4, 5, 6$ are interpreted as colored in two different colors RED and BLUE, respectively.

Lemma 1 (Meserve [30]). *For an arbitrary coloring of the Fano plane there is at least one line the 3 points of which are colored by the same color.*

Lemma 2 (Meserve [30]). *For any coloring of the Fano plane there cannot exist two different lines colored in opposite colors.*

Proof. As explained above, for any two different lines in the Fano plane there is exactly one point contained in these two lines. So it is impossible to have two different lines colored in opposite colors. ∎

Let φ be a Gödel numbering of $\mathcal{P}$. We consider the following class $\mathcal{U}_7$ of recursive functions. Each function $f \in \mathcal{U}_7$ is such that $f \in \mathcal{R}$ and:

(1) every $f(x)$ where $0 \leq x \leq 6$ equals either 2^s or 3^t, where $s, t \in \mathbb{N}$, $s, t \geq 1$,

(2) if $0 \leq x_1 < x_2 \leq 6, f(x_1) = 2^s$ and $f(x_2) = 2^t$, then $f(x_1) = f(x_2)$,

(3) if $0 \leq x_1 < x_2 \leq 6, f(x_1) = 3^s$ and $f(x_2) = 3^t$, then $f(x_1) = f(x_2)$,

(4) if $f(0) = 2^s$ and $f(1) = 2^s$ and $f(3) = 2^s$, then s is a correct φ-program for f,

(5) if $f(1) = 2^s$ and $f(2) = 2^s$ and $f(4) = 2^s$, then s is a correct φ-program for f,

(6) if $f(2) = 2^s$ and $f(3) = 2^s$ and $f(5) = 2^s$, then s is a correct φ-program for f,

(7) if $f(3) = 2^s$ and $f(4) = 2^s$ and $f(6) = 2^s$, then s is a correct φ-program for f,

(8) if $f(4) = 2^s$ and $f(5) = 2^s$ and $f(0) = 2^s$, then s is a correct φ-program for f,

(9) if $f(5) = 2^s$ and $f(6) = 2^s$ and $f(1) = 2^s$, then s is a correct φ-program for f,

(10) if $f(6) = 2^s$ and $f(0) = 2^s$ and $f(2) = 2^s$, then s is a correct φ-program for f,

(11) if $f(0) = 3^t$ and $f(1) = 3^t$ and $f(3) = 3^t$, then s is a correct φ-program for f,

(12) if $f(1) = 3^t$ and $f(2) = 3^t$ and $f(4) = 3^t$, then t is a correct φ-program for f,

(13) if $f(2) = 3^t$ and $f(3) = 3^t$ and $f(5) = 3^t$, then t is a correct φ-program for f,

(14) if $f(3) = 3^t$ and $f(4) = 3^t$ and $f(6) = 3^t$, then t is a correct φ-program for f,

(15) if $f(4) = 3^t$ and $f(5) = 3^t$ and $f(0) = 3^t$, then t is a correct φ-program for f,

(16) if $f(5) = 3^t$ and $f(6) = 3^t$ and $f(1) = 3^t$, then t is a correct φ-program for f,

(17) if $f(6) = 3^t$ and $f(0) = 3^t$ and $f(2) = 3^t$, then t is a correct φ-program for f.

Comment. In our construction of the class $\mathcal{U}_7$ the points $0, 1, 2, 3, 4, 5, 6$ can be interpreted as colored in two colors. Some points $f(i)$ are such that $f(i) = 2^s$ (these points are described below as RED) while some other points j are such that $f(j) = 3^t$ (these points are described below as BLUE). The properties of the Fano plane ensure that for every such coloring in two colors there exists a line such that the three points on this line are colored in the same color, and there cannot exist two lines colored in opposite colors.

Definition 9. *A* partial coloring *C of a Fano plane is an assignment of colors RED, BLUE, NONE to the points of the Fano plane.*

A partial coloring C_2 is an extension *of a partial coloring C_1 if every point colored RED or BLUE in C_1 is colored in the same color in C_2.*

A partial coloring *C of a Fano plane is called* complete *if every point is colored RED or BLUE.*

Lemma 3. *Given any partial coloring C of the points in the Fano plane assigning colors RED and BLUE to some but not all points such that no line contains three points in the same color, there exists*

(1) *a complete extension of the given coloring C such that it contains a line with three RED points, and*

(2) *a complete extension of the given coloring C such that it contains a line with three BLUE points.*

Proof. Color all the not colored points RED for the first function, and BLUE for the second function. ∎

Lemma 4. *Given any partial coloring C of points in the Fano plane assigning colors RED and BLUE to some but not all points such that no line contains three points in the same color, there exist numbers $k_*, \ell_* \in \mathbb{N}$ and*

(1) *a function $f_{RED} \in \mathcal{U}_7$ defined as $f(x) = 2^{\ell_*}$ for all x colored RED in C such that f_{RED} contains a line with three RED points, and all the points $0, 1, 2, 3, 4, 5, 6$ are colored RED or BLUE,*

(2) *a function $f_{BLUE} \in \mathcal{U}_7$ defined as $f(x) = 3^{k_*}$ for all x colored BLUE in C such that f_{BLUE} contains a line with three BLUE points, and all the points $0, 1, 2, 3, 4, 5, 6$ are colored RED or BLUE.*

Proof. Let *red* and *blue* be the subset of $\{0, 1, 2, 3, 4, 5, 6\}$ which are colored RED and BLUE, respectively. By the assumptions of the lemma we then have $red \cap blue = \emptyset$ and $red \cup blue \subset \{0, 1, 2, 3, 4, 5, 6\}$. Furthermore, so far there is no line in the Fano plane that has all points colored in the same color.

To show Assertion (1), let a function $h \in \mathcal{R}$ be chosen such that for all $\ell, x \in \mathbb{N}$

$$\varphi_{h(\ell)}(x) \quad = \quad \begin{cases} 3^\ell \,, & \text{if } x \in blue \,; \\ 2^\ell \,, & \text{otherwise} \,. \end{cases}$$

The construction directly implies that $\varphi_{h(\ell)} \in \mathcal{R}$ for all $\ell \in \mathbb{N}$. By the fixed point theorem (cf. Rogers [38]) there is a number $\ell_* \in \mathbb{N}$ such that the equality $\varphi_{\ell_*} = \varphi_{h(\ell_*)}$ is satisfied. Now, by Lemma 3 we know that the Fano plane has a line colored in RED, say (i, j, k) (cf. the comment made after the definition of the class $\mathcal{U}_7$). In accordance with our construction we conclude that

$$\varphi_{\ell_*}(i) \quad = \quad \varphi_{\ell_*}(j) \quad = \quad \varphi_{\ell_*}(k) \quad = \quad 2^{\ell_*} \,.$$

Consequently, $\varphi_{\ell_*} \in \mathcal{U}_7$ and Assertion (1) is shown by setting $f_{\mathrm{RED}} =_{df} \varphi_{\ell_*}$.

Assertion (2) can be shown *mutatis mutandis.* ∎

Theorem 1. *There is a deterministic QIM M that learns the class $\mathcal{U}_7$ with 7 queries.*

Proof. The desired QIM M queries the points $0, 1, \ldots, 6$. After having received $f(0), f(1), f(2), \ldots, f(6)$, it checks at which line all points have the same color, and outputs the φ-program corresponding to this line. Note that by Lemmata 1 and 2 there is at least one such line and there cannot be two lines colored in different colors. By the definition of the class $\mathcal{U}_7$ one can directly output a correct φ-program for the target function f. ∎

In order to show Theorem 3 below we need Smullyan's double fixed point theorem (cf. [38, 39]). So let us recall it here.

Theorem 2 (Smullyan [39])**.** *Let φ be any Gödel numbering of $\mathcal{P}$, and let $h, s \in \mathcal{R}^2$. Then there are $k_*, \ell_* \in \mathbb{N}$ such that simultaneously the equations $\varphi_{k_*} = \varphi_{h(k_*, \ell_*)}$ and $\varphi_{\ell_*} = \varphi_{s(k_*, \ell_*)}$ are satisfied.*

Theorem 3. *There exists no deterministic QIM learning $\mathcal{U}_7$ with 6 queries.*

Proof. The proof is by contradiction. Using Smullyan's double fixed point theorem [38] one can construct two functions f and $\tilde{f}$ such that both are in $\mathcal{U}_7$ but at least one of them is not correctly learned by the QIM M. This is done as follows.

Suppose that such a QIM M exists. Of course, in order to learn functions from the class $\mathcal{U}_7$ the QIM M should ask value queries for values in $\{0, 1, \ldots, 6\}$. If any other value query is made we define the corresponding function values to be 0. Furthermore, we assume without loss of generality that every query is made only one time.

By symmetry we can assume without loss of generality that the first value query of M is 0. To answer it, let functions $h, s \in \mathcal{R}^2$ be chosen such that for all $k, \ell \in \mathbb{N}$

$$\varphi_{h(k,\ell)}(0) \;=\; 2^k \quad \text{and} \quad \varphi_{s(k,\ell)}(0) \;=\; 2^k \;.$$

We return 2^k to the QIM M and set $Q_1 = \{0\}$.

Next, let the second query of M be x_2 for some $x_2 \in \{0, 1, \ldots, 6\} \setminus Q_1$. Now we define

$$\varphi_{h(k,\ell)}(x_2) \;=\; 2^k \quad \text{and} \quad \varphi_{s(k,\ell)}(x_2) \;=\; 2^k \;,$$

return 2^k to the QIM M and set $Q_2 = \{0, x_2\}$.

If the third query of the QIM M is x_3, where $x_3 \in \{0, 1, 2, 3, 4, 5, 6\} \setminus Q_2$ is such that $(0, x_2, x_3)$ form a line in the Fano plane then we define

$$\varphi_{h(k,\ell)}(x_3) \;=\; 3^\ell \quad \text{and} \quad \varphi_{s(k,\ell)}(x_3) \;=\; 3^\ell$$

to avoid that this line gets three equally colored points. In this case we return 3^ℓ.

Otherwise, we define

$$\varphi_{h(k,\ell)}(x_3) \;=\; 2^k \quad \text{and} \quad \varphi_{s(k,\ell)}(x_3) \;=\; 2^k \;,$$

return 2^k and set in both cases $Q_3 = \{0, x_2, x_3\}$.

The remaining three queries of the QIM M are answered in the following way (cf. Figure 1). If the next queried point x is such that no new line gets 3 equally colored points, we define

$$\varphi_{h(k,\ell)}(x) \;=\; 2^k \quad \text{and} \quad \varphi_{s(k,\ell)}(x) \;=\; 2^k \;,$$

return 2^k to the QIM M, and add x to the set of queried points.

If the queried point is such that some line can get 3 equally colored points, we define the value of the functions as

$$\varphi_{h(k,\ell)}(x) \;=\; 2^k \quad \text{or} \quad \varphi_{h(k,\ell)}(x) \;=\; 3^\ell$$

as well as

$$\varphi_{s(k,\ell)}(x) \;=\; 2^k \quad \text{or} \quad \varphi_{s(k,\ell)}(x) \;=\; 3^\ell$$

to avoid this. Then we return the defined value and add the queried point to the set of all already queried points.

Note that after these 6 queries the functions $\varphi_{h(k,\ell)}$ and $\varphi_{s(k,\ell)}$ are defined in the same way for all $\ell, k \in \mathbb{N}$ and all 6 queried points in Q_6.

To complete the proof we define the functions $\varphi_{h(k,\ell)}$ and $\varphi_{s(k,\ell)}$ for the remaining point $x \in \{0,1,2,3,4,5,6\} \setminus Q_6$ to be

$$\varphi_{h(k,\ell)}(x) = 2^k \quad \text{and } \varphi_{s(k,\ell)}(x) = 3^\ell$$

and set $\varphi_{h(k,\ell)}(y) = \varphi_{s(k,\ell)}(y) = 0$ for all $\ell, k \in \mathbb{N}$ and all $y \in \mathbb{N}$, where $y > 6$.

By construction, $\varphi_{h(k,\ell)}$, $\varphi_{s(k,\ell)} \in \mathcal{R}$ for all $\ell, k \in \mathbb{N}$. Now we apply Theorem 2. Hence there are $k_*, \ell_* \in \mathbb{N}$ such that simultaneously the equations $\varphi_{k_*} = \varphi_{h(k_*,\ell_*)}$ and $\varphi_{\ell_*} = \varphi_{s(k_*,\ell_*)}$ are satisfied. So we have

$$\begin{aligned}
\varphi_{k_*}(0) &= \varphi_{h(k_*,\ell_*)}(0) = 2^{k_*} \\
\varphi_{\ell_*}(0) &= \varphi_{s(k_*,\ell_*)}(0) = 2^{k_*} \\
\varphi_{k_*}(x_1) &= \varphi_{h(k_*,\ell_*)}(x_1) = 2^{k_*} \\
\varphi_{\ell_*}(x_1) &= \varphi_{s(k_*,\ell_*)}(x_1) = 2^{k_*} ,
\end{aligned}$$

and so on. But by construction the function φ_{k_*} must have a line $(x_{i_1}, x_{i_2}, x_{i_3})$ in the Fano plane such that

$$\varphi_{k_*}(x_{i_j}) = \varphi_{h(k_*,\ell_*)}(x_{i_j}) = 2^{k_*} \quad \text{for } j = 1,2,3 .$$

Consequently, we know that $\varphi_{k_*} \in \mathcal{U}_7$.

Analogously, the function φ_{ℓ_*} has a line $(z_{i_1}, z_{i_2}, z_{i_3})$ in the Fano plane such that

$$\varphi_{\ell_*}(z_{i_j}) = \varphi_{s(k_*,\ell_*)}(z_{i_j}) = 3^{\ell_*} \quad \text{for } j = 1,2,3 .$$

Therefore, we conclude that $\varphi_{\ell_*} \in \mathcal{U}_7$. So, we set $f =_{df} \varphi_{k_*}$ and $\tilde{f} =_{df} \varphi_{\ell_*}$. As shown above, the functions f and $\tilde{f}$ belong to $\mathcal{U}_7$ and coincide in their values on all points queried by the QIM M. Hence, M must fail to learn f or $\tilde{f}$, a contradiction. ∎

Theorem 4. *There is a nondeterministic QIM M learning $\mathcal{U}_7$ with 3 queries.*

Proof. The QIM M starts with a nondeterministic branching of the computation into 7 possibilities corresponding to the 7 lines in the Fano plane. In each case, all 3 points i, j, k are queried. If $f(i), f(j), f(k)$ are not of the same color then the computation path is aborted. If they are of the same color, e.g., $f(i) = 2^{s_i}, f(j) = 2^{s_j}, f(k) = 2^{s_k}$, then the definition of the class $\mathcal{U}_7$ ensures that $s_i = s_j = s_k$ and the QIM M outputs s_i which is a correct program computing the function f. ∎

Theorem 5. *There is no nondeterministic QIM learning $\mathcal{U}_7$ with 2 queries.*

Proof. By Lemma 4, there are two distinct functions in the class $\mathcal{U}_7$ with the same values queried by the nondeterministic algorithm. The output is not correct for at least one of them. ∎

In contrast to Theorem 3 we can show that 6 queries are already enough for a probabilistic QIM to achieve successful learning via value queries with a probability bigger than 1/2.

Theorem 6. *There is a probabilistic QIM M learning $\mathcal{U}_7$ with probability $\frac{4}{7}$ with 6 queries.*

Proof. There are 7 possibilities to choose a subset of 6 points from the 7 points of the Fano plane. So the desired probabilistic QIM M branches its computation in 7 branches corresponding to the 7 different subset of size 6 of the points of the Fano plane. In each branch all 6 points are queried. If the QIM M finds a line (i, j, j) in the Fano plane such that all points of it have the same color, e.g., $f(i) = f(j) = f(k) = 3^s$, it outputs s. By definition of the class $\mathcal{U}_7$ the result is a correct program computing the function f. Otherwise, the QIM M aborts the computation path.

It remains to analyze the probability of success. In the worst-case there is only a single monochromatic line (i, j, j) in the Fano plane. So, there are 4 subsets of size 6 containing the points of this line and there are 3 subsets of size 6 which miss at least one point of this line. Hence, the probability of success is 4/7. ∎

Next, we deal with the minimum number of queries needed by a probabilistic QIM M to achieve a probability greater than 0.

Theorem 7. *There is a probabilistic QIM M learning $\mathcal{U}_7$ with probability $\frac{1}{7}$ with 3 queries.*

Proof. The algorithm starts with branching its computation into 7 possibilities corresponding to the 7 lines in Fano plane. Each branch is reached with probability 1/7. In each branch, all 3 points i, j, k are queried. If $f(i), f(j), f(k)$ are not of the same color then the computation path is aborted. If they are of the same color, e.g., $f(i) = f(j) = f(k) = 2^s$, then s is output. By definition of the class $\mathcal{U}_7$ the result is a correct program computing the function f. ∎

However, the situation for the remaining number of queries is not yet completely analyzed. We may be interested to know what probability of success can be achieved by asking 4 or 5 queries. So far, we have only the following theorem.

Theorem 8. *There is a probabilistic QIM M learning $\mathcal{U}_7$ with probability $\frac{2}{7}$ with 5 queries.*

Proof. The desired QIM M will use its 5 queries to query in each branch two lines of the Fano plane. Since any two lines of the Fano plane share one point, it is clear that 5 queries are sufficient to query two lines. As above, if a line (i, j, k) of the Fano plane is found such that all points of it have the same color, e.g., $f(i) = f(j) = f(k) = 2^s$, then s is output. Otherwise the branch is aborted.

There are 21 possibilities to choose any two lines of the Fano plane. Again, in the worst-case there is only one line in the Fano plane such that all its points have the same color. So we may choose this line and any other line of the remaining 6 lines. This gives 6 successful branches, while the remaining 15 branches are aborted. Hence, the probability of success is $6/21 = 2/7$. ∎

Theorem 9. *For every prime number p, there is a p-ultrametric QIM M learning $\mathcal{U}_7$ with p-probability 1 with 2 queries.*

Proof. The desired QIM M branches its computation path into 7 branches at the root, where each branch corresponds to exactly one line of the Fano plane. The corresponding nodes are denoted by $\ell_1, \ldots, \ell_7$, respectively, in Figure 2 below.

We assign to each edge the amplitude $1/7$. At the second level, each of these branches is branched into 3 subbranches each of which is assigned the amplitude $1/3$. So far we have at level three 21 nodes denoted by $v_1, \ldots, v_{21}$ (cf. Figure 2).

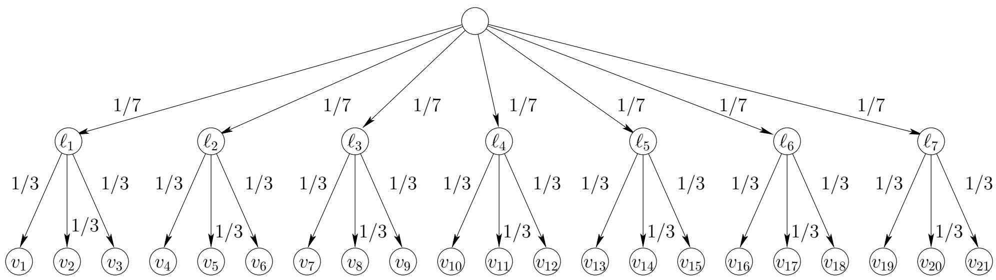

Figure 2: The first three levels of the DAG representing the computation of the QIM

For each of these nodes we formulate two queries. Let v be such that its father node corresponds to the line containing the point i, j, k of the Fano plane, where we order these points such that $i < j < k$. If v is the leftmost node then we query (i, j), if v is the middle node then we query (j, k) and if v is the rightmost node then we query (i, k). Every triple of nodes having the same father share a register, say r_{ijk}. Initially, the register contains the value $\uparrow$ which stands for "no output." The node activated when reached in the computation path sends the following value to r_{ijk}. After having received the answer to its queries, e.g., $f(i) = 2^s$ and $f(j) = 3^t$ then it writes 0 in r_{ijk}, and if the values coincide, e.g., $f(i) = 3^t$ and $f(j) = 3^t$, then it writes t in r_{ijk}.

Looking at any triple of nodes having a common father at the third level, then we note that the following 8 cases may occur as answer. We use again the corresponding colors, where R and B are used as shortcuts for RED and BLUE, respectively.

Thus, we need for each node at the third level 8 outgoing edges as the table above shows. If the edge corresponds to a pair (R, R) or (B, B) then we assign the amplitude $1/2$ and otherwise the amplitude $-1/4$. Note that sum of these amplitudes is again 1.

Finally, we join each triple as shown in table above into one node, e.g., the edges corresponding to (B, B), (B, R), and (B, R) are joined. If the total amplitude of such

(i,j)	(j,k)	(i,k)	(i,j)	(j,k)	(i,k)
(B,B)	(B,R)	(B,R)	(R,R)	(R,B)	(R,B)
(B,B)	(B,B)	(B,B)	(R,R)	(R,R)	(R,R)
(B,R)	(R,R)	(B,R)	(R,B)	(B,B)	(R,B)
(B,R)	(R,B)	(B,B)	(R,B)	(B,R)	(R,R)

Figure 3: The 8 cases of possible answers for any triple of nodes having a common father at the third level.

a node at the third level is different from zero, then the node produces as output the value stored in register r_{ijk}. Figure 4 shows the part of the DAG for the queries performed for the first line of the Fano plane, i.e., for the line $(0,1,3)$. So this part starts at the nodes v_1, v_2 and v_3 shown in Figure 2. For the sake of readability, we show the queries asked at each node, i.e., $(0,1)$ at node v_1, $(1,3)$ at node v_2, and $(0,3)$ at node v_3. A blue edge denotes the case that both answers to the queries asked at the corresponding vertex returned a value of f indicating that the related nodes of the first line of the Fano plane are blue. This result is then propagated along the blue edges. Analogously, a red edge indicates that both answers corresponded to a red node of the first line of the Fano plane. If the answers returned function values indicating that the colors of the queried nodes of the first line of the Fano plane have different colors then the edge is drawn in black. Blue and red edges have the amplitude $1/2$ and the black edges have the the amplitude $-1/4$.

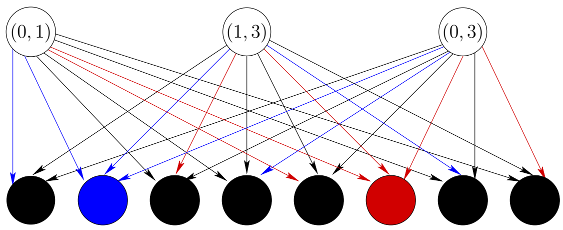

Figure 4: The part of the DAG representing the computation of the QIM M for the line $(0,1,3)$ starting at the nodes of the third level

It remains to show that the QIM M has the desired properties. By construction, at every computation path exactly two queries are asked.

Next, by Definition 6 it is obvious that the total amplitude of each node at the second level is $1/21$. Next, we consider any node at the third level. If a triple (B,B), (B,B), and (B,B) is joined then the total amplitude is

$$\frac{1}{21}\cdot\frac{1}{2}+\frac{1}{21}\cdot\frac{1}{2}+\frac{1}{21}\cdot\frac{1}{2}=\frac{1}{2\cdot 7}\ .$$

The same holds for (R,R), (R,R), and (R,R) (cf. Definition 6). Figure 4 shows the corresponding leaves in blue and red, respectively.

If a triple has a different form than considered above, e.g., (B, B), (B, R), and (B, R) then, again by Definition 6, we have for the total amplitude

$$\frac{1}{21} \cdot \frac{1}{2} - \frac{1}{21} \cdot \frac{1}{4} - \frac{1}{21} \cdot \frac{1}{4} = 0 \; .$$

One easily verifies that all remaining total amplitudes are also 0. The corresponding leaves are drawn in black in Figure 4. Finally, we perform the measurement. Clearly, for each leaf which has a total amplitude 0 the measurement results in $\|0\|_p = 0$. For the remaining nodes we obtain $\|\frac{1}{2\cdot 7}\|_p$ which is 1 for every prime p such that $p \notin \{2, 7\}$. If $p = 2$ then we have $\|\frac{1}{2\cdot 7}\|_2 = 2$ and for $p = 7$ we directly get $\|\frac{1}{2\cdot 7}\|_7 = 7$.

By Lemma 1 there must be at least one line such that all nodes have the same color, and by Lemma 2 it is not possible to have a line colored in RED and a line colored in BLUE simultaneously. So at least one node has p-probability at least 1, and the result output is correct in accordance with the definition of the class $\mathcal{U}_7$.

If there are several lines colored in the same color then distinct but correct results may be produced, since any two lines share exactly one point. Thus, the resulting p-probability is always no less than 1. ∎

The idea of this paper can be extended to obtain even more spectacular advantages of ultrametric algorithms over nondeterministic ones. In order to show this let us recall the general definition of a *finite projective plane.*

Definition 10. *A* finite projective plane *of* order *n, where $n \in \mathbb{N}$, $n > 0$, is a collection of $n^2 + n + 1$ lines and $n^2 + n + 1$ points such that*

(1) *every line contains $n + 1$ points,*

(2) *every point is on $n + 1$ lines,*

(3) *any two distinct lines intersect at exactly one point, and*

(4) *any two distinct points lie on exactly one line.*

So, the interesting question is whether or not there are infinitely many numbers n such that there is a projective plane of order n. It is known that for every prime power q there exists a projective plane of order q (cf. [6]).

This allows us to construct a class $\mathcal{U}_m$ of recursive functions similar to the class $\mathcal{U}_7$ above. The counterpart of Lemma 1 does not hold (cf. Figure 5) but this demands only an additional requirement for the function in the class to have a line colored in one color but not simultaneously two lines colored in opposite colors. Formally we define the class $\mathcal{U}_m$ as follows. Let q be a prime power, let $m =_{df} q^2 + q + 1$, let P_m be the finite projective plane of order q obtained from the proof given in [6], and let φ be any fixed Gödel numbering of $\mathcal{P}$.

The class $\mathcal{U}_m$ is the set of all functions $f \in \mathcal{R}$ satisfying the following properties:

(1) for every $x \in \{0, 1, \dots, m-1\}$ the value $f(x)$ equals either 2^s or 3^t, where $s, t \in \mathbb{N}$ and $s, t \geq 1$,

(2) if $0 \leq x_1 < x_2 \leq m-1$, $f(x_1) = 2^s$ and $f(x_2) = 2^t$ then $f(x_1) = f(x_2)$,

(3) if $0 \leq x_1 < x_2 \leq m-1$, $f(x_1) = 3^s$ and $f(x_2) = 3^t$ then $f(x_1) = f(x_2)$,

(4) either there is a line $(x_1, \dots, x_{q+1})$ in the projective plane P_m such that $f(x_1) = f(x_2) = \cdots = f(x_{q+1}) = 2^k$ and $\varphi_k = f$ or there is a line $(x_1, \dots, x_{q+1})$ in the projective plane P_m such that $f(x_1) = f(x_2) = \cdots = f(x_{q+1}) = 3^k$ and $\varphi_k = f$.

Next, let us provide an example showing that Lemma 1 does not generalize to all prime powers q and the corresponding finite projective planes P_m, where $m = q^2+q+1$. Instead of drawing a geometrical figure, we shall represent the finite projective plane by an $(m \times m)$-incidence matrix A. The rows of this matrix represent the points and the columns represent the points. Let $A = (a_{ij})_{\substack{i=1,\dots,m\\ j=1,\dots,m}}$; then entry $a_{ij} = 1$ if point i is at line j and $a_{ij} = 0$ otherwise.

	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}	L_{11}	L_{12}	L_{13}
P_1	1	1	1	1	0	0	0	0	0	0	0	0	0
P_2	1	0	0	0	1	1	1	0	0	0	0	0	0
P_3	1	0	0	0	0	0	0	1	1	1	0	0	0
P_4	1	0	0	0	0	0	0	0	0	0	1	1	1
P_5	0	1	0	0	1	0	0	1	0	0	1	0	0
P_6	0	1	0	0	0	1	0	0	1	0	0	1	0
P_7	0	1	0	0	0	0	1	0	0	1	0	0	1
P_8	0	0	1	0	1	0	0	0	1	0	0	0	1
P_9	0	0	1	0	0	1	0	0	0	1	1	0	0
P_{10}	0	0	1	0	0	0	1	1	0	0	0	1	0
P_{11}	0	0	0	1	1	0	0	0	0	1	0	1	0
P_{12}	0	0	0	1	0	1	0	1	0	0	0	0	1
P_{13}	0	0	0	1	0	0	1	0	1	0	1	0	0

Figure 5: A coloring of a finite projective plane of order 3 which does not have a monochromatic line

Using an incidence matrix A the conditions of Definition 10 directly translate into:

(1) A has constant row sum q,

(2) A has constant column sum q,

(3) the scalar product of any two distinct row vectors of A is 1,

(4) the scalar product of any two distinct column vectors of A is 1.

So the additional problem arising for all prime powers $q > 2$ is to find a suitable coloring on which the definition of the corresponding functions from $\mathcal{U}_m$ can be based.

It is straightforward to generalize Theorem 1 and Theorem 4.

Theorem 10. *Let q be any prime power, let $m = q^2 + q + 1$, and let $\mathcal{U}_m \subseteq \mathcal{R}$ be the corresponding class of recursive functions. Then there is a deterministic QIM M that learns the class $\mathcal{U}_m$ with m queries.*

Proof. The desired QIM M queries the points $0, 1, \ldots, m-1$. After having received $f(0), f(1), f(2), \ldots, f(m-1)$, it checks at which line all points have the same color, and outputs the φ-program corresponding to this line. Note that by the definition of the class $\mathcal{U}_m$ there is at least one such line and there cannot be two lines colored in different colors. If there is more than one line at which all points have the same color then the function values for the points on these monochromatic lines must coincide (cf. Conditions (2) and (3) of the definition of the class $\mathcal{U}_m$). So, by the definition of the class $\mathcal{U}_m$ one can directly output a correct φ-program for the target function f. ▮

Unfortunately, so far we cannot show that every deterministic QIM M needs m queries to learn the class $\mathcal{U}_m$. However, we conjecture that there is no deterministic QIM M that learns the class $\mathcal{U}_m$ with $m - 1$ queries.

Theorem 11. *Let q be any prime power, let $m = q^2 + q + 1$, and let $\mathcal{U}_m \subseteq \mathcal{R}$ be the corresponding class of recursive functions. Then there is a nondeterministic QIM M learning $\mathcal{U}_m$ with $q + 1$ queries.*

Proof. The QIM M starts with a nondeterministic branching of the computation into m possibilities corresponding to the m lines in the finite projective plane P_m. In each case, all $q + 1$ points $i_1, \ldots, i_{q+1}$ of the line corresponding to the branch are queried. If $f(i_1), \ldots, f(i_{q+1})$ are not of the same color then the computation path is aborted. If they are of the same color, e.g., $f(i_1) = 2^{k_1}, \ldots, f(i_{q+1}) = 2^{k_{q+1}}$, then the definition of the class $\mathcal{U}_m$ ensures that $k_{i_1} = \cdots = k_{q+1}$ and the QIM M outputs k_{i_1} which is a correct program computing the target function f. ▮

Note that the nondeterministic QIM M learning $\mathcal{U}_m$ needs q^2 less queries than the deterministic QIM from Theorem 10.

Theorem 12. *Let q be any prime power, let $m = q^2 + q + 1$, and let $\mathcal{U}_m \subseteq \mathcal{R}$ be the corresponding class of recursive functions. Then there is no nondeterministic QIM M learning $\mathcal{U}_m$ with q queries.*

Proof. We show the theorem indirectly. Suppose to the contrary that there is a nondeterministic QIM M learning $\mathcal{U}_m$ with q queries. For the sake of presentation, we illustrate the construction for the case that $q = 3$.

First, we construct a suitable coloring of P_m. Here we use an idea from Jepsen [22]. We fix the point P_1 and color it RED. Let $L_1, \ldots, L_{q+1}$ be the lines containing P_1. Now we color all points contained in $L_2, \ldots, L_{q+1}$ except P_1 by BLUE, and the points contained in L_1 with RED. The definition of a finite projective plane of order q ensures that L_1 and L_2 have exactly one point in common. By our construction this is P_1. Thus L_2 must contain q points not contained in L_1. The same holds for L_2 and L_3, i.e., L_3 must have q points neither contained in L_1 nor contained in L_2. Iterating this argument shows that L_{q+1} must contain q points not contained in any of the lines $L_1, \ldots, L_q$. Thus gives a total of $q+1$ points (contained in L_1) and q times q points for $L_2, \ldots, L_{q+1}$, i.e., $q+1+q^2$ points. Hence, we must have colored all points. Figure 6 shows the constructed coloring for the case $q = 3$. By construction, this coloring has

	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}	L_{11}	L_{12}	L_{13}
P_1	1	1	1	1	0	0	0	0	0	0	0	0	0
P_2	1	0	0	0	1	1	1	0	0	0	0	0	0
P_3	1	0	0	0	0	0	0	1	1	1	0	0	0
P_4	1	0	0	0	0	0	0	0	0	0	1	1	1
P_5	0	1	0	0	1	0	0	1	0	0	1	0	0
P_6	0	1	0	0	0	1	0	0	1	0	0	1	0
P_7	0	1	0	0	0	0	1	0	0	1	0	0	1
P_8	0	0	1	0	1	0	0	0	1	0	0	0	1
P_9	0	0	1	0	0	1	0	0	0	1	1	0	0
P_{10}	0	0	1	0	0	0	1	1	0	0	0	1	0
P_{11}	0	0	0	1	1	0	0	0	0	1	0	1	0
P_{12}	0	0	0	1	0	1	0	1	0	0	0	0	1
P_{13}	0	0	0	1	0	0	1	0	1	0	1	0	0

Figure 6: The constructed coloring of a finite projective plane of order 3

exactly one monochromatic line, i.e., all points of line L_1 are colored RED.

Since the QIM M is supposed to learn $\mathcal{U}_m$ with q queries there must be at least one branch in its computation where some subset of $\{0, 1, \ldots, q\}$ (corresponding to the points $P_1, \ldots, P_{q+1}$) of size at most q is queried and where the QIM M then makes the corresponding correct program as output. If the QIM M does not have such a branch then we are already done.

Let i be a point from $\{0, 1, \ldots, q\}$ which is not queried. Now, we change the color of point P_{i+1} from RED to BLUE. Then we obtain a new coloring which has q many lines colored BLUE. This follows again directly from the definition of a finite projective plane. But the values returned to the QIM M at the branch considered before remain unchanged and so it must output the same program as before which is, however, for the current coloring no longer correct. Hence, it violates Condition (2) of Definition 1, a contradiction. ∎

Next, we turn our attention to probabilistic QIM.

Theorem 13. *Let q be any prime power, let $m = q^2 + q + 1$, and let $\mathcal{U}_m \subseteq \mathcal{R}$ be the corresponding class of recursive functions. Then there is a probabilistic QIM M learning $\mathcal{U}_m$ with probability $\frac{q^2}{q^2+q+1}$ with $m-1$ queries.*

Proof. Since $\binom{m}{m-1} = m$, there are m possibilities to choose a subset of $m-1$ points from the m points of a finite projective plane P_m. So the desired probabilistic QIM M branches its computation into m branches corresponding to the m different subset of size $m-1$ of the points of the plane P_m. In each branch all $m-1$ points are queried. If the QIM M finds a line $(i_1, \ldots, i_{q+1})$ in the plane P_m such that all points of it have the same color, e.g., $f(i_1) = \cdots = f(i_{q+1}) = 3^k$, it outputs k. By definition of the class $\mathcal{U}_m$ the result is a correct program computing the function f. Otherwise, the QIM M aborts the computation path.

So, it remains to analyze the probability of success. In the worst-case there is only single monochromatic line $(i_1, \ldots, i_{q+1})$ in the plane P_m. There are q^2 subsets of size $m-1$ containing the points of this monochromatic line and there are $m - q^2 = q + 1$ subsets of size $m-1$ which miss at least one point of this line. Hence, the probability of success is $q^2/(q^2 + q + 1)$. ∎

It should be noted that the probability established in Theorem 13 tends to 1 if m tends to infinity.

Next it is easy to see that Theorem 7 generalizes as follows. The proof is *mutatis mutandis* the same.

Theorem 14. *Let q be any prime power, let $m = q^2 + q + 1$, and let $\mathcal{U}_m \subseteq \mathcal{R}$ be the corresponding class of recursive functions. Then there is a probabilistic QIM M learning $\mathcal{U}_m$ with probability $\frac{1}{m}$ with $q + 1$ queries.*

Unfortunately, so far we could not determine the minimum number of queries needed to achieve a probability exceeding $1/2$.

So, it remains to find out whether or not p-ultrametric QIMs do also achieve an advantage over nondeterministic QIM in the general case. Though so far we could not completely solve this problem, we succeeded to achieve partial results. Let us start by looking at the case that $q = 3$. Then we obtain the class $\mathcal{U}_{13}$ which is defined by using the projective plane P_{13}. We have 13 lines and each line consists of precisely four points. However, as the following theorem shows, there is a p-ultrametric QIM that learns $\mathcal{U}_{13}$ two queries.

Theorem 15. *For every prime number p, there is a p-ultrametric QIM M learning the class $\mathcal{U}_{13}$ with p-probability* 1 *with 2 queries.*

Proof. The desired QIM M branches its computation path into 13 branches at the root, where each branch corresponds to exactly one line of the projective plane P_{13}. We assign to each edge the amplitude $1/13$. Now, the idea to reduce the number of queries necessary is as follows. First, instead of asking 4 queries as the nondeterministic QIM did, we could ask the leftmost three points and the rightmost three points of the corresponding line, respectively. Therefore, we branch the computation at the second level into two subbranches each of which is assigned the amplitude 1/2. In the left branch the leftmost three points are considered and in the right branch the rightmost three points of the corresponding line are considered. Thus, at the second level we have a total of 26 nodes.

Since we already know how to replace 3 queries by 2 queries, at the third level each branch is branched into three subbranches. That is, the sub-DAG shown in Figure 4 is plugged in here. The amplitudes assigned to the edges in the sub-DAGs are the same as before. So for each line we have two branches which consist of the plugged in sub-DAGs. Each of these sub-DAGs has 8 leaves. However, only two leaves has a total amplitude which is not zero, i.e., the blue leave and the red leave.

Now, we join the two corresponding blue leaves and the two corresponding red leaves. Here each outgoing edge has the amplitude 1. Finally, the new leave outputs the value stored in register $r_{i_\ell j_\ell k_\ell}$, where the subindex ℓ indicates the always the left register is used. Figure 7 shows the corresponding sub-DAG for the first line. As shown in Figure 5, the first line contains the points P_1, P_2, P_3, P_4. For the sake of presentation, these points are denoted by $1, 2, 3, 4$ for short in Figure 7.

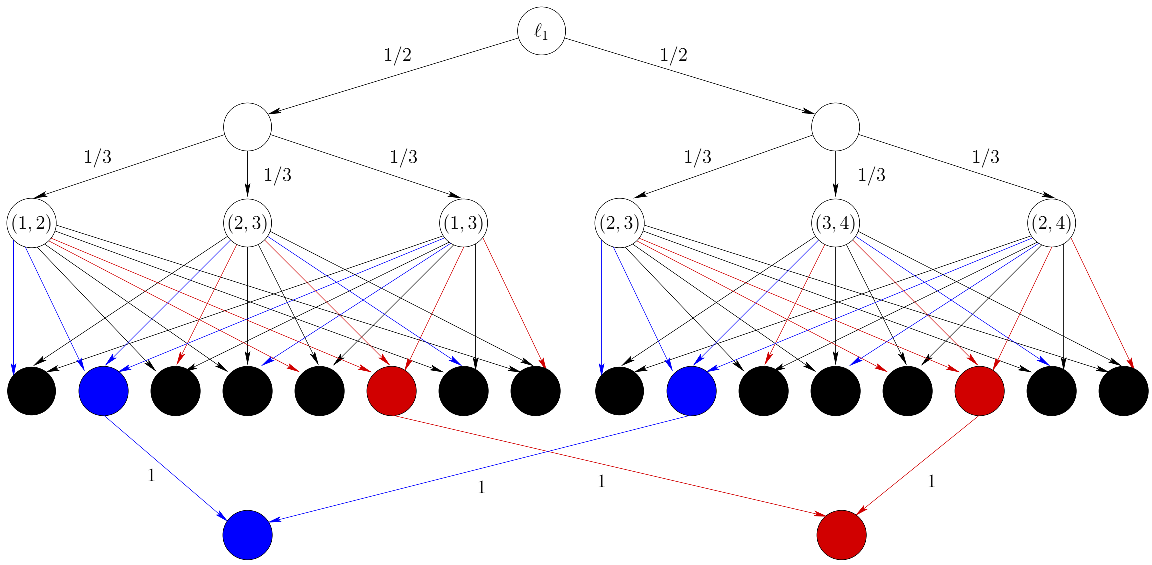

Figure 7: The part of the DAG representing the computation of the QIM M for the line $(0, 1, 3, 4)$ starting at the node of the second level

The labels at the new edges show the new amplitudes.

It remains to show that the QIM M has the desired properties. By construction, at every computation path exactly two queries are asked.

In accordance with Definition 6 we directly see that each node at the second level has the total amplitude 1/13 and the total amplitude of each node at the third level is1/26. Therefore, each node at the fourth level has the total amplitude 1/78. By Definition 6 we thus obtain for the blue nodes and the red nodes at the fifth level the following total amplitude

$$\frac{1}{78}\cdot\frac{1}{2}+\frac{1}{78}\cdot\frac{1}{2}+\frac{1}{78}\cdot\frac{1}{2}=\frac{1}{4\cdot 13}\ .$$

All the black nodes at the fifth level have the total amplitude 0, since for every black node there are always two black edges pointing to it and either one blue edge or one red edge entering it. Thus, we have

$$\frac{1}{78}\cdot\frac{1}{2}-\frac{1}{78}\cdot\frac{1}{4}-\frac{1}{78}\cdot\frac{1}{4}=0\ .$$

Finally, the blue node and the red node at level six have the total amplitude $\frac{1}{2\cdot 13}$. Finally, we perform the measurement. Clearly, for each leaf which has a total amplitude 0 the measurement results in $\|0\|_p = 0$. For the remaining leaves we obtain $\|\frac{1}{2\cdot 13}\|_p$ which is 1 for every prime p such that $p \notin \{2, 13\}$. If $p = 2$ then we have $\|\frac{1}{2\cdot 13}\|_2 = 2$ and for $p = 13$ we obtain $\|\frac{1}{2\cdot 13}\|_7 = 13$.

By construction there must be at least one line such that all nodes have the same color, and there is exactly one monochromatic line in the projective plane P_{13} used to define the class $\mathcal{U}_{13}$ (cf. Property (4) of the Definition of $\mathcal{U}_m$). So at least one node has p-probability at least 1. Furthermore, since two points are queried twice, it cannot happen that the leftmost three points are blue (or red) and the rightmost three points are red (or blue). Therefore, the result output is correct by the definition of the class $\mathcal{U}_{13}$. ∎

We note that Theorem 15 directly generalizes to the class $\mathcal{U}_{21}$, i.e., for the case that $q = 2^2$. In this case the projective plane P_{21} has 21 lines and 21 points and each line contains exactly 5 points. Consequently, we can use exactly the same proof as we did for the class $\mathcal{U}_{13}$, since the leftmost three points and the rightmost three points still share one point which is then queried twice. That is, we directly obtain the following corollary.

Corollary 1. *For every prime number p, there is a p-ultrametric QIM M learning the class $\mathcal{U}_{21}$ with p-probability* 1 *with 2 queries.*

Consequently, it seems that this idea can be pushed even further. If $q = 5$ then in the projective plane P_{31} each line has exactly 6 points. One could then try to ask the leftmost 4 pints and the rightmost 4 points and to use Theorem 15 to reduce 4 queries to 2 queries. However, it is not clear how far such a recursive application may be applied. The obvious difficulty is based on the construction of the classes $\mathcal{U}_m$ on projective planes P_m. Since we only know projective planes to exist if q is a prime power, the recursive application may stop somewhere and to determine the full potential of the ideas developed so far is a future research subject.

4. Conclusions

In this paper we have studied active learning of classes of recursive functions from value queries. We compared the query complexity of deterministic, nondeterministic, probabilistic, and ultrametric QIM and showed the somehow unexpected result that p-ultrametric QIM can learn classes of recursive function with significantly fewer queries than nondeterministic, probabilistic QIM can do.

The situation resembles quantum computation. In quantum computation we also find the usage of amplitudes and of measurements to transform amplitudes into real numbers. Quantum computation is famous for algorithms unimaginable in classical implementation. For instance, there exists a quantum query algorithm computing the Boolean function $x_1 \oplus x_2$ asking only one query but computing the binary xor-function function with probability 1. The essence of this algorithm is the quantum parallelism of the computation process (cf. Bernstein and Vazirani [5]). One computation path queries the value of x_1, the other computation path queries the value of x_2, and the addition of the amplitudes results in the probability 1 for the correct result and in the probability 0 for the wrong result.

Our main result (Theorem 9) is based on the same idea. So, it remains to explore further problems for which ultrametric algorithms can achieve a substantial advantage over nondeterministic algorithm or probabilistic algorithms.

References

[1] Andris Ambainis, Kalvis Apsītis, Rūsiņš Freivalds, and Carl H. Smith. Hierarchies of probabilistic and team *FIN*-learning. *Theoret. Comput. Sci.*, 261(1):91–117, 2001.

[2] Andris Ambainis and Juris Smotrovs. Enumerable classes of total recursive functions: Complexity of inductive inference. In *Algorithmic Learning Theory, 4th International Workshop on Analogical and Inductive Inference, AII '94, 5th International Workshop on Algorithmic Learning Theory, ALT '94, Reinhardsbrunn Castle, Germany, October 1994, Proceedings*, volume 872 of *Lecture Notes in Artificial Intelligence*, pages 10–25. Springer-Verlag, 1994.

[3] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[4] Kaspars Balodis, Anda Beriņa, Kristīne Cīpola, Maksims Dimitrijevs, Jānis Iraids, Kārlis Jēriņs, Vladimirs Kacs, Jānis Kalējs, Rihards Krilšauks, Kārlis Lukstiņš, Reinholds Raumanis, Irina Scegulnaja, Natālija Somova, Anna Vanaga, and Rūsiņš Freivalds. On the state complexity of ultrametric finite automata. unpublished manuscript, 2013.

[5] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.

[6] Thomas Beth, Dieter Jungnickel, and Hans Lenz. *Design Theory*. Bibliographisches Institut, Wissenschaftsverlag, Mannheim, Wien, Zürich, 1985.

[7] Nader H. Bshouty. Exact learning from membership queries: Some techniques, results and new directions. In Sanjay Jain, Rémi Munos, Frank Stephan, and Thomas Zeugmann, editors, *Algorithmic Learning Theory, 24th International Conference, ALT 2013, Singapore, October 6-9, 2013, Proceedings*, volume 8139 of *Lecture Notes in Artificial Intelligence*, pages 33–52, Berlin, Heidelberg, New York, 2013. Springer.

[8] François Denis, Aurélien Lemay, and Alain Terlutte. Learning regular languages using non deterministic finite automata. In *Grammatical Inference: Algorithms and Applications, 5th International Colloquium, ICGI 2000, Lisbon, Portugal, September 11-13, 2000, Proceedings*, volume 1891 of *Lecture Notes in Artificial Intelligence*, pages 39–50. Springer-Verlag, 2000.

[9] B. Dragovich and A. Yu. Dragovich. A p-adic model of DNA sequence and genetic code. *p-Adic Numbers, Ultrametric Analysis, and Applications*, 1(1):34–41, 2009.

[10] Rūsiņš Freivalds. Inductive inference of recursive functions: Qualitative theory. In *Baltic Computer Science*, volume 502 of *Lecture Notes in Computer Science*, pages 77–110. Springer, Berlin, 1991.

[11] Rūsiņš Freivalds. Ultrametric automata and Turing machines. In Andrei Voronkov, editor, *Turing-100*, volume 10 of *EPiC Series*, pages 98–112. EasyChair, 2012.

[12] Rūsiņš Freivalds, Jānis Bārzdiņš, and Kārlis Podnieks. Inductive inference of recursive functions: Complexity bounds. In *Baltic Computer Science*, volume 502 of *Lecture Notes in Computer Science*, pages 111–155. Springer, Berlin, 1991.

[13] Rūsiņš Freivalds, Efim Kinber, and Carl H. Smith. On the impact of forgetting on learning machines. *J. ACM*, 42(6):1146–1168, 1995.

[14] Pedro García, Manuel Vázquez de Parga, Gloria I. Álvarez, and José Ruiz. Learning regular languages using nondeterministic finite automata. In *Implementation and Applications of Automata, 13th International Conference, CIAA 2008, San Francisco, California, USA, July 21-24, 2008. Proceedings*, volume 5148 of *Lecture Notes in Computer Science*, pages 92–101. Springer, 2008.

[15] William Gasarch, Efim Kinber, Mark Pleszkoch, Carl Smith, and Thomas Zeugmann. Learning via queries, teams, and anomalies. *Fundamenta Informaticae*, 23(1):67–89, 1995. Shorter version in Third Annual Workshop on Computational Learning Theory, 1990, pages 327-337, published by Morgan Kaufman.

[16] William Gasarch and Carl H. Smith. A survey of inductive inference with an emphasis on learning via queries. In Andrea Sorbi, editor, *Complexity, Logic, and Recursion Theory*, volume 187 of *Lecture Notes in Pure and Applied Mathematics*, pages 225–260. Marcel Dekker, Inc., 1997.

[17] William I. Gasarch and Carl H. Smith. Learning via queries. *J. ACM*, 39(3):649–675, 1992.

[18] E. Mark Gold. Language identification in the limit. *Inform. Control*, 10(5):447–474, 1967.

[19] Fernando Quadros Gouvea. *p-adic Numbers: An Introduction (Universitext).* Springer, Berlin, second edition, 1983.

[20] Inguna Greitāne. Probabilistic inductive inference of indices in enumerable classes of total recursive functions. In *Analogical and Inductive Inference, International Workshop AII '89, Reinhardsbrunn Castle, GDR, October 1989, Proceedings*, volume 397 of *Lecture Notes in Artificial Intelligence*, pages 277–287. Springer, 1989.

[21] Juraj Hromkovič. *Design and Analysis of Randomized Algorithms. Introduction to Design Paradigms.* Springer, Berlin, Heidelberg, 2005.

[22] Charles H. Jepsen. On colorings of finite projective planes. *Discrete Mathematics*, 63(1):95–96, 1987.

[23] Kārlis Jēriņš, Kaspars Balodis, Rihards Krišlauks, Kristīne Cīpola, and Rūsiņš Freivalds. Ultrametric query algorithms. unpublished manuscript, 2012.

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

[25] Efim Kinber and Thomas Zeugmann. One-sided error probabilistic inductive inference and reliable frequency identification. *Inform. Comput.*, 92(2):253–284, 1991.

[26] Efim B. Kinber and Thomas Zeugmann. Refined query inference. In *Analogical and Inductive Inference, International Workshop AII '89, Reinhardsbrunn Castle, GDR, October 1989, Proceedings*, volume 397 of *Lecture Notes in Artificial Intelligence*, pages 148–160. Springer-Verlag, 1989.

[27] Neal Koblitz. *p-adic Numbers, p-adic Analysis, and Zeta-Functions.* Springer, Berlin, second edition, 1984.

[28] S. V. Kozyrev. Ultrametric analysis and interbasin kinetics. In *2nd International Conference on p-Adic Mathematical Physics, Belgrade, Serbia and Montenegro, 15 - 21 September 2005*, volume 826 of *American Institute of Physics Conference Proceedings*, pages 121–128, 2006.

[29] Steffen Lange, Thomas Zeugmann, and Sandra Zilles. Learning indexed families of recursive languages from positive data: A survey. *Theoret. Comput. Sci.*, 397(1-3):194–232, 2008. Special issue Forty Years of Inductive Inference: Dedicated to the 60th Birthday of Rolf Wiehagen.

[30] Bruce E. Meserve. *Fundamental Concepts of Geometry.* Dover Publications, New York, 1983.

[31] Léa Meyer. Probabilistic language learning under monotonicity constraint. *Theoret. Comput. Sci.*, 185(1):81–128, 1997.

[32] Léa Meyer. Comparing the power of probabilistic learning and oracle identification under monotonicity constraints. In *Algorithmic Learning Theory, 9th International Conference, ALT '98, Otzenhausen, Germany, October 1998, Proceedings*, volume 1501 of *Lecture Notes in Artificial Intelligence*, pages 306–320. Springer, 1998.

[33] Léa Meyer. Aspects of complexity of probabilistic learning under monotonicity constraints. *Theoret. Comput. Sci.*, 268(2):275–322, 2001.

[34] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[35] Alexander Ostrowski. Über einige Lösungen der Funktionalgleichung $\varphi(x)\cdot\varphi(y) = \varphi(xy)$. *Acta Mathematica*, 41(1):271–284, 1916.

[36] L. Pitt. Probabilistic inductive inference. *J. ACM*, 36(2):383–433, 1989.

[37] Leonard Pitt and Carl H. Smith. Probability and plurality for aggregations of learning machines. *Inform. Comput.*, 77(1):77–92, 1988.

[38] Hartley Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, 1967. Reprinted, MIT Press 1987.

[39] Raymond M. Smullyan. *Theory of formal systems*, volume 47 of *Annals of Mathematics Studies.* Princeton University Press, Princeton, New Jersey, USA, 1961.

[40] V. S. Vladimirov, I. V. Volovich, and E. I. Zelenov. *P-Adic Analysis and Mathematical Physics.* World Scientific, Singapore, 1994.

[41] Yu-Chieh Wu, Yue-Shi Lee, Jie-Chi Yang, and Show-Jane Yen. An integrated deterministic and nondeterministic inference algorithm for sequential labeling. In *Information Retrieval Technology 6th Asia Information Retrieval Societies Conference, AIRS 2010, Taipei, Taiwan, December 1-3, 2010. Proceedings*, volume 6458 of *Lecture Notes in Computer Science*, pages 221–230. Springer, 2010.

[42] Thomas Zeugmann and Steffen Lange. A guided tour across the boundaries of learning recursive languages. In *Algorithmic Learning for Knowledge-Based Systems*, volume 961 of *Lecture Notes in Artificial Intelligence*, pages 190–258. Springer, 1995.

[43] Thomas Zeugmann and Sandra Zilles. Learning recursive functions: A survey. *Theoret. Comput. Sci.*, 397(1-3):4–56, 2008. Special issue Forty Years of Inductive Inference: Dedicated to the 60th Birthday of Rolf Wiehagen.