A model for computational simulation of intuition

Rūsiņš Freivalds[1]

Abstract. The main difficulty in all previous attempts to construct a mathematical model for human intuition is the lack of explanation wherefrom this outside help can come. If this help-message contains a piece of information about the problem to be solved then it is unbelievable that somebody sends such a help to us. If the help-message contains no information then why this message can help anybody. A possible mechanism of such a help based on finite automata is proposed. In the proposed model the advice comes in a form of a random sequence of symbols. It is quite possible that human organism can sometimes receive random signals from a non-living source outside the organism. This help indeed contains zero information about the problem but nonetheless it helps. Moreover, it turns out that this mechanism is different from probabilistic, nondeterministic and quantum computation.

1 INTRODUCTION

The Oxford English Dictionary defines intuition as the ability to acquire knowledge without inference and/or the use of reason.

Sharon A. Klinger writes in [15]: ”You have probably heard a dozen different terms describing the concept of ’intuition’. You may call it instinct, second sight, a gut feeling, an inner voice, your sixth sense, or even your soul. Whatever you call it, you have the opportunity to experience it every single day and use it to benefit your life and your world.

The dictionary defines intuition as ’direct knowing or learning of something without the conscious use of reasoning’. Clinical parapsychologists have defined this ’knowing’ as the perceiving of information without the use of the *physical* senses (hence, the term ’extrasensory perception’). They have determined it to be a function belonging to the vast ’unused’ portion of the brain. Still others (myself included) see it as the voice of our eternal (and consequently divine) self, our spirit. And, finally, those who wish to stay away from the more philosophical, clinical, spiritual or even technical interpretations simply call any intuitive experience ’a hunch’. ”

No wonder, there is a serious difficulty to construct even a primitive mathematical model for human intuition. This paper originated from re-considering the notion of ”automata that take an advice”. This notion was introduced by R.Karp/R.Lipton [14] for Turing machines, by T.Yamakami [29] for finite automata and in a different way by R.Freivalds [8]. Theorems of this paper consider the case when the advice is a random string of bits. It turns out that several theorems from the author’s paper [11] can easily be used in this paper as well.

The use of nonconstructive methods of proof in mathematics has a long and dramatic history. In 1888 a young German mathematician David Hilbert presented to his colleagues three short papers on invariant theory. Invariant theory was the highly estimated achievement of Paul Gordan who had produced highly complicated constructive proofs but left several important open problems. The young David Hilbert had solved all these problems and had done much-much more. Paul Gordan was furious. He was not ready to accept the new solutions because they provided no explicit constructions. Hilbert merely proved that the solutions cannot fail to exist. Gordan refused to accept this as mathematics. He even used the term theology and categorically objected to publication of these papers. Nonetheless the papers were published first in *Götingen Nachrichten* and later, in final form, in [12]. In the nineteen-forties the situation, however, changed. In spite of all philosophical battles the nonconstructive methods found their way even to discrete mathematics. This was particularly surprising because here all the objects were finite and it seemed that no kind of distinction between actual infinity and potential infinity could influence these proofs while most of the discussions between intuitionists and classicists were around these notions. Paul Erdös produced many nice nonconstructive proofs, the first paper of this kind being [5]. Many such proofs are considered in a survey paper by Joel Spencer [25] and a recent monograph by Noga Alon and Joel H. Spencer [1]. R. Karp and R. Lipton have introduced in [14] a notion Turing machine that takes advice which is in fact a usage of a nonconstructive help from outside in a process of computation. Later C. Damm and M. Holzer [3] have adapted this notion of advice for finite automata. The adaptation was performed in the most straightforward way (what is quite natural) and later extensively used by T.Yamakami and his coauthors [21, 27, 28, 29].

Another version of the notion a finite automaton that takes advice was introduced in [9, 10] under the name nonconstructive finite automaton. These notions are equivalent for large amounts of nonconstructivity (or large amounts of advice) but, for the notion introduced in [3] languages recognizable with polynomial advice are the same languages which are recognizable with a constant advice. The notion of the amount of nonconstructivity in [10] is such that the most interesting results concern the smallest possible amounts of nonconstructivity. A similar situation was in sixties of the 20th century with space complexity of Turing machines. At first space complexity was considered for one-tape offline Turing machines and it turned out that space complexity is never less than linear. However, it is difficult to prove such lower bounds. Then the seminal paper by R.E.Stearns, J.Hartmanis and P.M.Lewis [26] was published and many-tape Turing machines became a standard tool to study sublinear space complexity.

The essence of nonconstructive methods is as follows. An algorithm is presented in a situation where (seemingly) no algorithm is possible. However, this algorithm has an additional input where a special help is fed in. If this help is correct, the algorithm works correctly. On the other hand, this help on the additional input does not just provide the answer. There still remains much work for the al-

[1] Institute of Mathematics and Computer Science, University of Latvia, email: Rusins.Freivalds@mii.lu.lv

gorithm. Is this nonconstructivism merely a version of nondeterminism? Not at all. Nondeterministic finite automata (both with 1-way and 2-way inputs) recognize only regular languages while nonconstructive finite automata (as defined in [3, 21]) can recognize some nonregular and even nonrecursive languages. We will see below that this notion is different also from probabilistic finite automata.

What is a random string of bits? Are we to demand a correct result for arbitrary "random" string, including a string consisting only of zeros? We are to answer all these questions before we propose a formal definition. In our case, the advice is supposed to be an arbitrary infinite Martin-Löf random sequence [18] but the finite automaton can use only a finite initial fragment of this sequence. The length of the used fragment of the random sequence is up to the finite automaton. We demand that the result is to be correct with arbitrary Martin-Löf random sequence used. We demand also that no other restrictions on the randomness are used. Moreover, we notice that this our "naive" definition allows to simulate an additional counter, and it is well-known that automata with one counter can recognize nonregular languages. To avoid this additional possibility, we make our definition more complicated demanding that our sequence of random bits is 2-infinite (i.e., infinite to both ends). One no more can simulate a counter but still our determnistic finite automaton with random bits written on a separate tape can recognize languages not recognizable by probabilistic finite automata. Martin-Löf's original definition of a random sequence was in terms of constructive null covers; he defined a sequence to be random if it is not contained in any such cover. Since then it is discovered that Martin-Löf random sequences can be equivalently defined in many different ways. This is a convincing argument proving that "random sequences" are Martin-Löf sequences. M.Li and P.Vitanyi's book [19] is an excellent introduction to these ideas.

It is important to note that the theorems below are not valid if the advice string is not random. This proves that randomness is indeed a resource of computation. Artists, especially visual artists have noticed this effect long ago.

Deterministic, nondeterministic and alternating 2-way finite automata recognize only regular languages. On the other hand, it was proved in [7] that 2-way probabilistic finite automata with bounded error can recognize nonregular languages.

Definition 1 *A deterministic finite automaton with written random bits (shortly: wrb) is a deterministic non-writing 2-tape finite automaton one tape of which contains the input word , and the other tape contains a 2-infinite primitive Martin-Löf random sequence, the automaton is 2-way on every tape, and it stops producing a the correct result in a finite number of steps for arbitrary input word. Additionally it is demanded that the head of the automaton never goes beyond the markers showing the beginning and the end of the input word.*

C. Dwork and L. Stockmeyer proved in [4] a theorem on limitations of 2-way probabilistic finite automata (shortly: 2pfa) with bounded error. This theorem is useful for us:

Theorem A. [4] *Let $L \subseteq \Sigma^*$. Suppose there is an infinite set I of positive integers and, for each $m \in I$, an integer $N(m)$ and sets $W_m = \{w_1, w_2, \cdots, w_{N(m)}\}$,
$U_m = \{u_1, u_2, \cdots, u_{N(m)}\}$ and $V_m = \{v_1, v_2, \cdots, v_{N(m)}\}$ of words such that*

1. *$\mid w \mid \leq m$ for all $w \in W_m$,*
2. *for every integer k there is an m_k such that $N(m) \geq m^k$ for all $m \in I$ with $m \geq m_k$, and*
3. *for all $1 \leq i, j \leq N(m), u_j w_i v_j \in L$ iff $i = j$.*

Then $L \notin AM(2pfa)$.

We use this result to prove

Theorem 1 *1. The language $L = \{x2x \mid x \in \{0,1\}^*\}$ cannot be recognized with a bounded error by a probabilistic 2-way finite automaton, 2. The language $L = \{x2x \mid x \in \{0,1\}^*\}$ can be recognized by a deterministic finite automaton with wrb.*

Proof. (1) Let m be an arbitrary integer. For arbitrary $i \in \{0, 1, 2, \cdots, 2^m - 1\}$ we define the word $x_i(m)$ as the word number i in the lexicographical ordering of all the binary words of the length m. We define the words u_i, w_i, v_i in our usage of Theorem A as $\{\oslash, x_i(m), 2x_i(m)\}$.

(2) Let the input word be $x(r)2z(s)$ where r and s are the lengths of the corresponding words. At first, the 2-tape automaton finds a fragment $01111\cdots$ which has the length at least r and uses it as a counter to test whether $r = s$. Then the automaton searches for another help-word. If the help-word turns out to be y then the automaton tests whether $x(r) = y$ and whether $z(s) = y$. ◯

2 SEARCH FOR A SUITABLE DEFINITION

The definition used in the second item of Theorem 1 is our first (but not final) attempt to formalize the main idea of the notion of help from outside bringing zero information about the problem to be solved. Unfortunately, this definition allows something that was not intended to use. Such automata can easily simulate a counter, and 2-way automata with a counter, of course, can recognize nonregular languages. On the other hand, the language L in our Theorem 1 cannot be recognized by a finite automaton with one counter. Hence we try to present a more complicated definition of help from outside bringing zero information to avoid the possibility to simulate a counter.

Martin-Löf's original definition of a random sequence was in terms of constructive null covers; he defined a sequence to be random if it is not contained in any such cover. Leonid Levin and Claus-Peter Schnorr proved a characterization in terms of Kolmogorov complexity: a sequence is random if there is a uniform bound on the compressibility of its initial segments. Schnorr gave a third equivalent definition in terms of martingales (a type of betting strategy). Li and Vitanyi's book [19] is the standard introduction to these ideas.

Kolmogorov complexity [24, 17]: Kolmogorov complexity can be thought of as a lower bound on the algorithmic compressibility of a finite sequence (of characters or binary digits). It assigns to each such sequence w a natural number $K(w)$ that, intuitively, measures the minimum length of a computer program (written in some fixed programming language) that takes no input and will output w when run. Given a natural number c and a sequence w, we say that w isc-incompressible if $K(w) \geq\mid w \mid -c$.

An infinite sequence S is Martin-Löf random if and only if there is a constant c such that all of S's finite prefixes are c-incompressible.

Constructive null covers [18]: This is Martin-Löf's original definition. For a finite binary string w we let C_w denote the cylinder generated by w. This is the set of all infinite sequences beginning with w, which is a basic open set in Cantor space. The product measure $\mu(C_w)$ of the cylinder generated by w is defined to be $2- \mid w \mid$.

Every open subset of Cantor space is the union of a countable sequence of disjoint basic open sets, and the measure of an open set is the sum of the measures of any such sequence. An effective open set is an open set that is the union of the sequence of basic open sets determined by a recursively enumerable sequence of binary strings. A constructive null cover or effective measure 0 set is a recursively enumerable sequence U_i of effective open sets such that $U_{i+1} \subseteq U_i$ and $\mu(U_i) \leq 2^{-i}$ for each natural number i. Every effective null cover determines a G_δ set of measure 0, namely the intersection of the sets U_i.

A sequence is defined to be Martin-Löf random if it is not contained in any G_δ set determined by a constructive null cover.

Constructive martingales [23]: A martingale is a function $d : \{0,1\}^* \to [0,\infty)$ such that, for all finite strings w, $d(w) = (d(w0) + d(w1))/2$, where ab is the concatenation of the strings a and b. This is called the "fairness condition"; a martingale is viewed as a betting strategy, and the above condition requires that the better plays against fair odds. A martingale d is said to succeed on a sequence S if $\limsup_{n\to\infty} d(S_n) = \infty$, where S_n is the first n bits of S. A martingale d is constructive (also known as weakly computable, lower semi-computable, subcomputable) if there exists a computable function $\widehat{d} : \{0,1\}^* \times N \to Q$ such that, for all finite binary strings w:

1. $\widehat{d}(w,t) \leq \widehat{d}(w,t+1) < d(w)$, for all positive integers t,
2. $\lim_{t\to\infty} \widehat{d}(w,t) = d(w)$.

A sequence is Martin-Löf random if and only if no constructive martingale succeeds on it.

Definition 2 *A 2-infinite sequence of bits is a sequence $\{a_i\}$ where $i \in (-\infty,\infty)$ and all $a_i \in \{0,1\}$.*

Definition 3 *We say that a 2-infinite sequence of bits is Martin-Löf random if for arbitrary $i \in (-\infty,\infty)$ the sequence $\{b_n\}$ where $b_n = a_{i+n}$ for all $i \in N$ is Martin-Löf random, and the sequence $\{c_n\}$ where $c_n = a_{i-n}$ for all $i \in N$ is Martin-Löf random.*

Definition 4 *A deterministic* finite automaton with written random bits *(shortly: wrb) is a deterministic non-writing 2-tape finite automaton one tape of which contains the input word , and the other tape contains a 2-infinite Martin-Löf random sequence, the automaton is 2-way on every tape, and it stops producing a the correct result in a finite number of steps for arbitrary input word. Additionally it is demanded that the head of the automaton never goes beyond the markers showing the beginning and the end of the input word.*

Nondeterministic, probabilistic, alternating, etc. automata with wrb differ from deterministic ones only in the nature of the automata but not in usage of tapes or Martin-Löf random sequences.

Definition 5 *We say that a language L is recognizable by a deterministic finite automaton A with wrb if A for arbitrary 2-infinite Martin-Löf random sequence accepts every input word $x \in L$ and rejects every input word $x \notin L$.*

Definition 6 *We say that a language L is enumerable by a deterministic finite automaton A with wrb if A for arbitrary 2-infinite Martin-Löf random sequence accepts every input word $x \in L$ and do not accept any input word $x \notin L$.*

Definition 7 *A deterministic finite automaton with wrb on unbounded input is a deterministic non-writing 2-tape finite automaton one tape of which contains the input word , and the other tape contains a 2-infinite Martin-Löf random sequence, the automaton is 2-way on every tape, and it stops producing a the correct result in a finite number of steps for arbitrary input word. It is not demanded that the head of the automaton always remains between the markers showing the beginning and the end of the input word.*

Recognition and enumeration of languages by deterministic finite automata with wrb is not particularly interesting because of the following two theorems.

Theorem 2 *A language L is enumerable by a deterministic finite automaton with wrb on unbounded input if and only if it is recursively enumerable.*

Proof. J.Bārzdiņš [2] proved that arbitrary one-tape deterministic Turing machine can be simulated by a 2-way finite deterministic automaton with 3 counters directly and by a 2-way finite deterministic automaton with 2 counters using a simple coding of the input word. (Later essentially the same result was re-discovered by other authors.) Hence there exists a a 2-way finite deterministic automaton with 3 counters accepting every word in L and only words in L.

Let x be an arbitrary word in L. To describe the processing of x by the 3-counter automaton we denote the content of the counter i ($i \in \{1,2,3\}$) at the moment t by $d(i,t)$. The word

$$00000101^{d(1,0)}0101^{d(2,0)}0101^{d(3,0)}000101^{d(1,1)}-$$
$$0101^{d(2,1)}0101^{d(3,1)}00\cdots$$
$$\cdots 00101^{d(1,s)}0101^{d(2,s)}0101^{d(3,s)}0000$$

where s is the halting moment, is a complete description of the processing of x by the automaton.

Our automaton with wrb tries to find a fragment of the 2-infinite Martin-Löf random sequence on the help-tape such that:

1. it starts and ends by 0000,
2. the initial fragment
$$0101^{d(1,0)}0101^{d(2,0)}0101^{d(3,0)}00$$
is exactly 0000010010010, (i.e., the all 3 counters are empty,
3. for arbitrary t the fragment
$$0101^{d(1,t)}0101^{d(2,t)}0101^{d(3,t)}0101^{d(1,t+1)}-$$
$$0101^{d(2,t+1)}0101^{d(3,t+1)}$$
corresponds to a legal instruction of the automaton with the counters.

Since the 2-infinite sequence is Martin-Löf random, such a fragment definitely exists in the sequence infinitely many times. The correctness of the fragment can be tested using the 3 auxiliary constructions below.

Construction 1. Assume that $w_k \in \{0,1\}^*$ and $w_m \in \{0,1\}^*$ are two subwords of the input word x such that:

1. they are immediately preceded and immediately followed by symbols other than $\{0,1\}$,

2. a deterministic finite 1-tape 2-way automaton has no difficulty to move from w_k to w_m and back, clearly identifying these subwords,

Then there is a deterministic finite automaton with wrb recognizing whether or not $w_k = w_m$.

Proof. As in Theorem 1. ◯

Construction 2. Assume that 1^k and 1^m are two subwords of the help-word y such that:

1. they are immediately preceded and immediately followed by symbols other than $\{0, 1\}$,
2. a deterministic finite 1-tape 2-way automaton has no difficulty to move from w_k to w_m and back, clearly identifying these subwords,
3. both k and m are integers not exceeding the length of the input word.

Then there is a deterministic finite automaton with wrb recognizing whether or not $k = m$.

Proof. Similar the proof of Construction 1. ◯

Construction 3. Assume that $1^{k_1}, 1^{k_2}, \cdots, 1^{k_s}$ and $1^{m_1}, 1^{m_2}, \cdots, 1^{m_t}$ are subwords of the help-word y such that:

1. they are immediately preceded and immediately followed by symbols other than 1,
2. a deterministic finite 1-tape 2-way automaton has no difficulty to move from one subword to another and back, clearly identifying these subwords,
3. both $k_1 + k_2 + \cdots + k_s$ and $m_1 + m_2 + \cdots + m_t$ are integers not exceeding the length of the input word.

Then there is a deterministic finite automaton with wrb recognizing whether or not $k_1 + k_2 + \cdots + k_s = m_1 + m_2 + \cdots + m_t$.

Proof. Similar the proof of Construction 2. ◯

Corollary of Theorem 2. *A language L is recognizable by a deterministic finite automaton with wrb on unbounded input if and only if it is recursive.*

Theorem 2 and its corollary show that the standard definition of the automaton with wrb should avoid the possibility to use the input tape outside the markers. However, even our standard definition allows recognition and enumeration of nontrivial languages. The proof of Theorem 1 can be easily modified to prove

Theorem 3 *1. The language $L = \{x2x \mid x \in \{0, 1\}^*\}$ cannot be recognized with a bounded error by a probabilistic 2-way finite automaton,*

2. The language $L = \{x2x \mid x \in \{0, 1\}^*\}$ can be recognized by a deterministic finite automaton with wrb.

What happens if we allow to have two (or more) help-tapes containing 2-infinite Martin-Löf sequences? We will see below that again this help turns out to be superfluous.

Definition 8 *A deterministic finite automaton with wrb with 2 help tapes is a deterministic non-writing 3-tape finite automaton one tape of which contains the input word , and each of the two other tapes contains a 2-infinite Martin-Löf random sequence, the automaton is 2-way on every tape, and it stops producing a the correct result in a finite number of steps for arbitrary input word. It is not demanded that the head of the automaton always remains between the markers showing the beginning and the end of the input word.*

Theorem 4 *A language L is enumerable by a deterministic finite automaton with wrb with 2 help tapes if and only if it is recursively enumerable.*

Theorem 5 *A language L is recognizable by a deterministic finite automaton with wrb with 2 help tapes if and only if it is recursive.*

3 MAIN RESULTS

Theorem 6 *The unary language PERFECT SQUARES $= \{1^n \mid (\exists m)(n = m^2)\}$ can be recognized by a deterministic finite automaton with wrb.*

Proof. It is well-known that

$$1 + 3 + 5 + \cdots + (2n - 1) = n^2.$$

The deterministic automaton with wrb searches for a help-word (being a fragment of the given 2-infinite Martin-Löf sequence)

$$00101110111110 \cdots 01^{2n-1}00.$$

At first, the input word is used as a counter to test whether each substring of 1's is exactly 2 symbols longer than the preceding one. Then the help-word is used to test whether the length of the input word coincides with the number of 1's in the help-word. ◯

Theorem 7 *The unary language PERFECT CUBES $= \{1^n \mid (\exists m)(n = m^3)\}$ can be recognized by a deterministic finite automaton with wrb.*

Proof. In a similar manner the formula

$$1 + 3(n - 1) + 3(n - 1)^2 = n^3 - (n - 1)^3$$

suggests a help-word

$$000[1]00[101110111]00[1011111101111111111]-$$

$$00 \cdots 00[101^{n-1}01^{(n-1)^2}]000$$

where symbols $[,]$ are invisible (i.e. they are not written explicitly; they are given in our text only for the readers convenience). At first, the input word is used as a counter to test whether the help-word is correct but not whether its length is sufficient. Then the help-word is used to test whether the length of the input word coincides with the number of 1's in the help-word. ◯

Theorem 8 *The unary language PRIMES $= \{1^n \mid n \text{ is prime}\}$ can be recognized by a deterministic finite automaton with wrb.*

Idea of the proof. The automaton searches for a help-word (being a fragment of the given 2-infinite Martin-Löf sequence)

$$101101110 \ldots 01^{n-1}.$$

This fragment is used to test whether n is divided by an m such that $1 < m < n$. ◯

We define a language UNARY 3-SAT as follows. The term $term_1 = x_k$ is coded as $[term_1]$ being 21^k, the term $term_2 = \neg x_k$ is coded as $[term_2]$ being 31^k, the subformula f being $(term_1 \vee term_2 \vee term_3)$ is coded as $[f]$ being $[term_1] \vee [term_2] \vee [term_3]$. The CNF being $f_1 \wedge f_2 \wedge \cdots \wedge f_m$ is coded as $[f_1] \wedge [f_2] \wedge \cdots \wedge [f_m]$. ◯

Theorem 9 *Every $L \in NP$ is reducible by a deterministic log-space bounded Turing machine to a language L' such that L' is enumerable by a deterministic finite automaton with wrb.*

Proof. 3-SAT is NP-complete. Hence L is reducible by a deterministic log-space bounded Turing machine to 3-SAT. The language 3-SAT is reducible by a deterministic log-space bounded Turing machine to $unary3 - SAT$. The language UNARY 3-SAT is enumerable by a deterministic finite automaton B with wrb which can be constructed using Construction 1, Construction 2 and Construction 3. ◯

Theorem 10 *If a language L is enumerable by a nondeterministic finite automaton with wrb then $L \in NP$.*

Proof. R.Fagin's theorem [6] in descriptive complexity theory states that the set of all properties expressible in existential second-order logic is precisely the complexity class NP. N.Immerman 1999 gave a detailed proof of the theorem [13].

Our proof rather closely simulates Immerman's proof. Essentially, we use second-order existential quantifiers to choose existentially a help-word and a computation tableau. For every timestep, we arbitrarily choose the finite state control's state, the contents of every tape cell, and which nondeterministic choice we must make. Verifying that each timestep follows from each previous timestep can then be done with a first-order formula. ◯

The paper [9] contains the following

Theorem 11 *There exists a nonrecursive language L such that it can be nonconstructively recognized with nonconstructivity $(\log n)^2$.*

In contrast, we have a result showing that if the nonconstructive help is a Martin-Löf sequence, then the language can be only recursive. Moreover, we have

Theorem 12 *If a language L is recognizable by a nondeterministic finite automaton with wrb then $L \in NP \cap co - NP$.*

Unfortunately, we have no strengthening of Theorems 10,12 for deterministic finite automata with wrb. Theorem 13 below shows that this open problem can be difficult.

Theorem 13 *Every language enumerable by a deterministic finite automaton with wrb is also recognizable by a nondeterministic finite automaton with wrb if and only if $P = NP$.*

Proof. Immediately from Theorem 12 and Lemma 1 below. ◯

Lemma 1 *If every language enumerable by a deterministic finite automaton with wrb is also recognizable by a nondeterministic finite automaton with wrb then $P = NP$.*

Proof. Let L be an arbitrary language in NP. Then by Theorem 9 L is reducible by a log-space DTM to a language $L' \in NP$ such that L' is enumerable by a deterministic finite automaton with wrb. The assumption of our theorem implies that L' recognizable by a nondeterministic finite automaton with wrb, and, consequently, also the complement of L' is recognizable by a nondeterministic finite automaton with wrb. By Theorem 12 it follows that $L' \in co - NP$, and by Theorem 9 it follows that $L \in co - NP$. ◯

Theorem 14 *If a language L is enumerable by a nondeterministic finite automaton with wrb then L is also enumerable by a deterministic finite automaton with wrb.*

Proof. The deterministic automaton with wrb searches for a help-word (being a fragment of the given 2-infinite Martin-Löf sequence) of a special kind described below.

Let $x \in L$ be an input word , a help-word w (we denote the length of w by h) and let an computation path P by the nondeterministic automaton on (x, w) be fixed such that the head on w never leaves w. At first, we describe a word y containing enough information about the nondeterministic choices and later we use this word y to construct a deterministic finite automaton with wrb to accept the word (x, z) with an appropriate z. Let w be a unary word $w_1 w_2 w_3 \cdots w_m$. Then

$$y = w_1 2 c_{(1,1)} c_{(2,1)} \cdots c_{(h,1)} 2 w_2 2 -$$

$$c_{(1,2)} \cdots c_{(h,2)} 2 w_3 \cdots 2 w_m 2 c_{(1,m)} \cdots c_{(h,m)}$$

where $c_{(i,j)}$ denotes:

- $\oslash$, if at the computation path P there is no occurrence when the head on the help-tape is on the symbol w_j and the head on the input tape at this moment is on the i-th symbol of x;
- code of triple (p, s, i), if at the computation path P there is an occurrence when the head on the help-tape is on the symbol w_j and the head on the input tape at this moment is on the i-th symbol of x, and at this moment the state of the automaton is p , the instruction s is performed on the computation path P, and the number i in a unary notation. (Please notice that p and s are elements of finite sets with a cardinality bounded by a constant depending only on the program of the nondeterministic automaton.)

Let z be an expression of y in binary notation by a symbol-to-symbol translation of the word y. The needed deterministic automaton working on arbitrary 2-infinite Martin-Löf sequence searches for a fragment z of the given 2-infinite sequence. This search involves a huge amount of comparisons (1) whether or not the tested help-word is compatible with the instructions of the nondeterministic finite automaton with wrb and (2) whether the tested help-word is compatible with the computation path of the nondeterministic finite automaton with wrb. For instance, let at some moment it appears that the current instruction of the nondeterministic automaton (contained in $c_{(i,j)}$)prescribes moving the head on the help-tape one position to the right with the head on the input tape staying at the same position. Then the head of the deterministic automaton with wrb leaves its position and for a time being the input tape is used only as a counter. The moves to the leftmost position and then the counter is used to move the help-tape head to the position of $c_{(i,j+1)}$ simultaneously comparing whether $c_{(i,j+1)}$ contains an instruction compatible with the instruction performed at the previous step. If at some moment it turns out that the help word is not correct (i.e. it does not correspond either to the instructions of the nondeterministic automaton, or it does not correspond to a legal path of computation), the deterministic automaton searches for a new help-word. Since the help tape contains a 2-infinite Martin-Löf random sequence, if there is an accepting path of the nondeterministic automaton there is also an accepting path of the deterministic automaton. ◯

Corollary of Theorem 14. If a language L is recognizable by a nondeterministic finite automaton with wrb then L is also recognizable by a deterministic finite automaton with wrb.

Now we consider closure properties of the class of languages enumerable by deterministic finite automata with wrb.

A family of languages is an ordered pair (Σ, Λ), where (1) Σ is an infinite set of symbols; (2) Λ is a set of formal languages; (3) For each L in Λ there exists a finite subset $\Sigma_1 \subset \Sigma$ such that $L \subseteq \Sigma_1^*$; and (4) $L \neq \oslash$ for some L in Λ.

Given alphabets Σ_1 and Σ_2, a function $h : \Sigma_1 \to \Sigma_2^*$ such that $h(uv) = h(u)h(v)$ for all u and v in Σ_1^* is called a homomorphism on Σ_1^*. Let ϵ denote the empty word. If h is a homomorphism on Σ_1^* and $h(x) \neq \epsilon$ for all $x \neq \epsilon$ in Σ_1^*, then h is called an ϵ-free homomorphism.

A trio is a family of languages closed under ϵ-free homomorphism, inverse homomorphism, and intersection with regular language.

A full trio, also called a cone, is a trio closed under arbitrary homomorphism.

A (full) semi-AFL is a (full) trio closed under union.

A (full) AFL is a (full) semi-AFL closed under concatenation and the Kleene plus.

It is known that the regular languages, the context-free languages, and the recursively enumerable languages are all full AFLs. However, the context sensitive languages and the recursive languages are AFLs, but not full AFLs because they are not closed under arbitrary homomorphisms.

Theorem 15 *The class of all languages enumerable (recognizable) by deterministic finite automata with wrb is an AFL but not a full AFL.*

Proof. 1. Inverse homomorphisms. Let L be a language enumerable (recognizable) by deterministic finite automaton A with wrb, h be a homomorphism, and L' be a language such that $h(L') = L$. Then L' can be enumerated (recognized) by an automaton B simulating A on a word $y = h(x)$ which is not written on any tape but can be easily uniquely restored from x by a deterministic finite automaton.

2. ϵ-free homomorphisms. The proof is technically complicated but similar to the proof of Theorem 14.

3. Intersections, unions, concatenations, Kleene plus. Immediate.

4. Arbitrary homomorphisms. The class of all languages recognizable by deterministic finite automata with wrb is not closed under arbitrary homomorphisms.

Idea of the proof. Using the technique of our Theorem 2 it is possible to prove that every recursively enumerable language is a homomorphic image of a language recognizable by a deterministic finite automaton with wrb.

ACKNOWLEDGEMENTS

The research is supported by the project 271/2012 from the Latvian Council of Science.

REFERENCES

[1] Noga Alon and Joel H. Spencer. *The Probabilistic Method.* John Wiley and Sons, 2000.

[2] Jānis Bārzdiņš (Ja.M. Barzdin). On a Class of Turing Machines (Minsky Machines). *Algebra i Logika*, vol. 3, No. 1, 1963 (Russian) Review in *The Journal of Symbolic Logic,* vol. 32, No. 4, pp. 523–524, 1967.

[3] Carsten Damm and Markus Holzer. Automata that take advice. *Lecture Notes in Computer Science,* vol. 969, pp. 565–613, 1995.

[4] Cyntia Dwork and Larry Stockmeyer. Finite state verifiers I: the power of interaction. *Journal of the Association for Computing Machinery,* vol. 39, No. 4, pp. 800–828, 1992.

[5] Paul Erdös. Some remarks on the theory of graphs. *Bulletin of the American Mathematical Society,* vol. 53, No. 4 , pp. 292–294, 1947.

[6] Ronald Fagin. Generalized First-Order Spectra and Polynomial-Time Recognizable Sets. *Complexity of Computation,* ed. R. Karp, SIAM-AMS Proceedings, vol. 7, pp. 27–41, 1974.

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

[8] Rūsiņš Freivalds. Non-constructive methods for finite probabilistic automata. *International Journal of Foundations of Computer Science,* v. 19, No. 3, pp. 565–580, 2008.

[9] Rūsiņš Freivalds. Amount of nonconstructivity in finite automata. *Lecture Notes in Computer Science,* vol. 5642, pp. 227–236, 2009.

[10] Rūsiņš Freivalds. Amount of nonconstructivity in deterministic finite automata. *Theoretical Computer Science,* vol. 411, No. 38-39, pp. 3436–3443, 2010.

[11] Rūsiņš Freivalds. Multiple Usage of Random Bits in Finite Automata. *Lecture Notes in Computer Science,* vol. 7287, pp. 537–547, 2012.

[12] David Hilbert. Über die Theorie der algebraischen Formen. *Mathematische Annalen,* Bd. 36, pp. 473–534, 1890.

[13] Neil Immerman. *Descriptive Complexity,* New York: Springer-Verlag. pp. 113–119, 1999.

[14] Richard M. Karp and Richard Lipton. Turing machines that take advice. *L' Enseignement Mathematique*, vol. 28, pp. 191–209, 1982.

[15] Sharon A. Klinger. *Intuition And Beyond: A Step-by-Step Approach to Discovering Your Inner Voice.* Rider & Co, 2002.

[16] Andrei Nikolaevich Kolmogorov. Three approaches to the quantitative definition of information. *Problems in Information Transmission*, vol. 1, pp. 1–7, 1965.

[17] Leonid A. Levin. On the notion of a random sequence. *Soviet Mathematics Doklady,* vol. 14, pp. 1413–1416, 1973.

[18] Per Martin-Löf. The definition of random sequences. *Information and Control,* vol. 9, No. 6, pp. 602–619, 1966.

[19] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications,* (Second ed.). Springer-Verlag, 1997.

[20] Ilan Newman. Private vs. Common Random Bits in Communication Complexity. *Information Processing Letters,* vol. 39, 1991, pp. 67–71.

[21] Harumichi Nishimura, Tomoyuki Yamakami. Polynomial time quantum computation with advice. Information Processing Letters, vol. 90, No. 4, pp. 195204, 2004.

[22] Michael O. Rabin. Probabilistic automata. *Information and Control,* vol. 6, pp. 230–245, 1963.

[23] Claus-Peter Schnorr. A unified approach to the definition of random sequences. *Mathematical Systems Theory,* vol. 5, No. 3, pp. 246–258, 1971.

[24] Claus-Peter Schnorr. Process Complexity and Effective Random Tests. *Journal of Computer and System Sciences,* vol. 7, No. 4, pp. 376–388, 1973.

[25] Joel Spencer. Nonconstructive methods in discrete mathematics. *Studies in Mathematics,* vol. 17 of MAA Studies in Mathematics, Gian-Carlo Rota, editor, MAA, pp. 142–178, 1978.

[26] Richard Edwin Stearns, Juris Hartmanis, Philip M. Lewis II. Hierarchies of memory limited computations. *Proceedings of FOCS,* pp.179–190, 1965.

[27] Kohtaro Tadaki, Tomoyuki Yamakami, Jack C. H. Lin. Theory of one tape linear time Turing machines. *Lecture Notes in Computer Science,* vol. 2932, pp. 335–348, 2004.

[28] Tomoyuki Yamakami. Swapping lemmas for regular and context-free languages with advice. The Computing Research Repository (CoRR), CoRR abs/0808.4122, 2008.

[29] Tomoyuki Yamakami. The Roles of Advice to One-Tape Linear-time Turing machines and finite automata. *International Journal of Foundations of Computer Science,* v. 21, pp. 941–962, 2010.