CDMTCS Research Report Series

Randomness, Computability, and Algebraic Specifications

Bakhadyr Khoussainov
University of Auckland, New Zealand
Cornell University, USA

CDMTCS-017
August 1996

Centre for Discrete Mathematics and
Theoretical Computer Science

Randomness, Computability, and Algebraic Specifications

Khoussainov Bakhadyr
The University of Auckland, Auckland, New Zealand
Cornell University, Ithaca, New York 14850
e-mail: bmk@cs.auckland.ac.nz

1 Motivation

Random finite objects have been defined and investigated by means of tools borrowed from computability theory, so we have a fairly good picture of the interplay between randomness and computability [5] [7] [15]. In contrast, almost nothing has been known about the algebraic nature of random objects. Therefore the basic question investigated in this paper is the following:

Is it possible to develop an algebraic and computable theoretic approach to investigate randomness in strings/terms?

Of course, we need to explain what we mean by the algebraic–computable approach. We can attempt to explain our question in the following informal way:

Is it possible to use the methods, notions, and results of universal algebra and computability theory to understand the nature of random strings/terms?

In this paper we show that one can employ algebraic and computable theoretic methods and notions, such as for example congruence relations, free algebra, initial algebra, generators, computably enumerable sets, immune sets, equations, etc. to understand the nature of randomness in strings/terms. While Chaitin and Kolmogorov complexities are the notions by means of which randomness and computability interact with each other, it has not been clear how randomness can be related to (universal) algebra. The goal of this paper is to show how these three notions – randomness, computability, and (universal) algebra, can naturally interplay with each other.

We mention that Calude and Chatin have asked questions which are related to understanding algebraic nature of randomness [5] [6]. For example, Calude has been interested in introducing and investigating the notion of symmetry and transformations of random objects. Chaitin has been interested in finding instances of randomness in algebra and geometry.

The organization of the paper is as follows. In the first part of the paper we briefly discuss a few basic notions and results from universal algebra, theory of abstract data types, computability theory, complexity theory, and explain the question of Bergstra and Tucker from [2] [3] about specifiability of algebras.

In the second part, using a fixed point theorem from computability theory, we prove that the set of random strings (terms) is immune. Though this fact has been known, our proof gives a new and a simple way of showing noncomputability of random strings (terms). Based on this result, we provide simple, but interesting algebraic facts about (random) terms. For example, we show that any universal algebra effectively defined on the set of random terms is locally finite.

In the third and central part of the paper we show that the notion of randomness naturally defines an infinite algebra. We call this algebra the **Algebra of Random Terms (ART)**. It turns out that the algebra is finitely generated. Moreover, the word problem for this algebra is computably enumerable. Therefore one can investigate this algebra using methods and notions from universal algebra and computability theory. We show, for example, that this algebra can not be equationally specified in the sense of Bergstra-Tucker [1] [2] [3]. To the best of our knowledge, it is the first natural example which gives a negative answer to the problem of Bergstra-Tucker from [2] [3] on equational specifiability of abstract date types.

Finally, in the last part of the paper we formulate an open question concerning the ART.

We adopt a commonly used terminology from computability theory [14], universal algebra [8], theory of abstract data types [1] [2], and algorithmic information theory [5] [6] [7] [15].

2 Basic Notions

Universal Algebra. A **functional signature**, or equivalently **a functional language**, is a finite sequence

$$\Sigma = (\phi_1^{l_1}, \dots, \phi_m^{l_m}, c_1, \dots, c_k),$$

where $k \geq 1$, and $m, k, l_1, \dots, l_m \in \omega$, and ω is the set of natural numbers. Each c_i is called a **constant symbol** and each $\phi_i^{l_i}$ a **functional symbol of arity** l_i. We fix this signature till the end of this paper. A **universal algebra**, or briefly **algebra**, of this language is a system

$$(A, \phi_1^{l_1}, \dots, \phi_m^{l_m}, c_1, \dots, c_k),$$

where A is a nonempty set called the **domain of** $\mathcal{A}$, each $\phi_i^{l_i}$ is an operation on domain A of arity l_i, and each c_i is an element from domain A. Sometimes the operations $\phi_i^{l_i}$ are called **atomic**, or equivalently **basic operations**, of the algebra $\mathcal{A}$. A **subalgebra** of $\mathcal{A}$ is a subset $C \subset A$ together with the basic operations restricted to C such that C is closed under the operations. If $\mathcal{A}$ is an algebra and B is a subset of the domain of $\mathcal{A}$, then we can consider the smallest subalgebra $< B >$ of $\mathcal{A}$ containing B. The domain of $< B >$ is the intersection of all the domains of subalgebras containing B. The set B is called a **generator** of $< B >$. Note that $< B >$ contains all the constants. We say that $\mathcal{A}$ is **finitely generated** if it has a finite generator. An important example of a finitely generated algebra is the algebra defined as follows. The domain of the algebra is $GT(\Sigma)$ the set of all variable free terms, called **ground terms**, of the signature Σ. Each n-ary functional symbol $\phi \in \sigma$ naturally defines the n-ary operation, which is also denoted by ϕ, on $GT(\sigma)$ by

$$\text{the value of } \ \phi \ \text{ on } (t_1, \dots, t_n) \text{ is } \phi(t_1, \dots, t_n).$$

Thus, we have the algebra

$$(GT(\Sigma), \phi_1, \phi_2, \dots, \phi_n, c_1, \dots, c_k).$$

called the **absolutely free algebra**. It is clear that this algebra is finitely generated whose generators are $c_1, \dots, c_k$.

A **congruence relation** on an algebra $\mathcal{A}$ is an equivalence relation η on the domain such that any n-ary basic operation ϕ of the signature **respects** η, that is, for all $(x_1, y_1), \dots, (x_n, y_n) \in \eta$ we have

$$(f(x_1, \dots, x_n), f(y_1, \dots, y_n)) \in \eta.$$

This definition allows one to form a new algebra, called the factor algebra of $\mathcal{A}$ by η. The elements of the factor algebra are the equivalence classes under η; the atomic operations of the factor algebra are naturally induced by the corresponding operations of the underlying algebra $\mathcal{A}$.

Algebraic Specifications. In theoretical computer science a common way of viewing a data type is that of identifying the data type with a universal algebra [1] [2] [3]. The basic idea in this approach is to describe a data type by giving names to basic functions determined on the objects of the data type, and thus form an algebra. In this approach an **abstract data type (ADT)** is defined as being the isomorphism class of the data type, in other words, the isomorphism type of the **algebra**. Informally, an **algebraic specification** is a way to describe the abstract data type, or equivalently the algebra, using formal logical languages. The main idea is to specify the algebra by using its signature and some of special properties of the algebra. A very natural way to do this is to use different fragments of logical formalisms such as for example, equations, conditional equations, existential formulas, etc. We give the following definition for algebraic specifications.

Definition 2.1 *An* **equation** *is an expression of the form $t_1 = t_2$, where t_1, t_2 are terms of the functional signature. An* **algebraic (equational) specification** *E is a finite set of universally quantified equations.*

An algebra $\mathcal{A}$ is **specified** by E if $\mathcal{A}$ is isomorphic to the **initial system** defined by E. The **initial system** can be obtained as follows. Consider the **absolutely free algebra** of the signature Σ with generators $c_1, \ldots, c_k$. Consider the equational theory of E which is

$$Eq(E) = \{t_1 = t_2 | t_1, t_2 \in GT(\Sigma) \text{ and } t_1 = t_2 \text{ can be proved from } E\}.$$

Thus on the set of all ground terms we have an equivalence relation η_E defined by the equational theory $Eq(E)$:

$$\eta_E = \{(t_1, t_2) | t_1 = t_2 \in Eq(E)\}.$$

One can check that this equivalence relation is a congruence on the absolutely free algebra. Therefore we can define the factor algebra

$$\mathcal{A} = (T(\sigma)/\eta_E, \phi_1^{l_1}, \ldots, \phi_m^{l_m}, c_1, \ldots, c_k)$$

called the **initial system**, or equivalently the **initial algebra**, defined by E. Thus, two elements t_1 and t_2 of this algebra are equal if and only if their equality, that is expression $t_1 = t_2$, can be proved from E. This algebra satisfies the following fundamental properties. It is **finitely generated** by the elements c_1, ..., c_k. Every algebra satisfying the specification E and generated by $c_1, \ldots, c_k$ is a **homomorphic image** of $\mathcal{A}$. Thus, one can say that the initial algebra is, in some sense, a universal implementation of the specification E.

Note that the equality relation on every initial algebra $\mathcal{A}$ defined by a specification E is **computably enumerable**. Algebras with computably enumerable (computable) equality relations and computable operations are called **computably enumerable (computable) algebras**.

It turns out that not every computable algebra can be specified in its own language. For example, in [3] Bergstra and Tucker proved that the computable algebra $(\omega, 0, x+1, x^2)$ does not have an algebraic specification in its own langauge, that is, in the language (ϕ_1, ϕ_2, c), where each ϕ_i is a unary functional symbol. However, they provided an algebraic specification for the **expanded algebra** $(\omega, 0, x+1, x^2, x+y, x \times y)$. As a more general result, Bergstra and Tucker proved that any computable algebra has a **functional expansion** which possesses a finite equational algebraic specification [1]. Therefore Bergstra and Tucker [2] [3], and independently Goncharov [11] suggested the idea to specify a given algebra by allowing finite expansions of the initial signature. In other words, they ask the following question:

Can every computably enumerable finitely generated algebra be specified using finite functional expansions of the language Σ?

In this paper we give a natural example of a finitely generated computably enumerable algebra, called the **Algebra of Random Terms**, which can not be specified in all possible finite functional expansions. This answers the above question of Bergstra and Tucker negatively. Kassimov in [10] has already given a negative answer to the above question using a specific coding of a particular computably enumerable set. However, to the best of our knowledge, the Algebra of Random Terms is the first natural and simple example of unspecified algebra.

Notions from Computability Theory. We fix a Gödel enumeration $\Phi_0, \Phi_1, \ldots$ of all Turing Machines which define mappings from the set of natural numbers ω into the set $GT(\Sigma)$ of all ground terms. We assume that this list of Turing Machines contains also programs of the following two types,

$$t \quad \text{and} \quad t'(t_1, \ldots, t_{j-1}, \Phi_j, t_{j+1}, \ldots, t_n),$$

where t, t_1, ..., t_n are ground terms, and t' is a term containing $t_1, \ldots, t_n$ as subterms. In other words, we assume that for each ground term $t \in GT(\Sigma)$ there is an i such that the program of Turing machine

Φ_i is t and $\Phi_i(x) = t$ for all $x \in \omega$. Similarly, we assume that for any term $t'(t_1, \ldots, t_{j-1}, t_j, t_{j+1}, \ldots, t_n)$ and any k there is an i such that the program of Turing machine Φ_i is $t'(t_1, \ldots, t_{j-1}, \Phi_k, t_{j+1}, \ldots, t_n)$ and for any $x \in \omega$,

$$\Phi_i(x) = t'(t_1, \ldots, t_{i-1}, \Phi_j(0), t_{j+1}, \ldots, t_n).$$

In the paper we use the notion of immune set and a fixed point theorem from computability theory. A set S of ground terms is called **immune** if S is infinite and does not have infinite computably enumerable subsets. We will use the following version of the fixed point theorem: For any total computable function $\psi : \omega \to \omega$ there is an x (fixed point) such that $\Phi_x = \Phi_{\psi(x)}$.

3 Complexity, Randomness, and Immune Sets

Suppose that our finite signature $\Sigma = (\phi_1^{l_1}, \ldots, \phi_m^{l_m}, c_1, \ldots, c_k)$ contains either two functional unary symbols or one binary functional symbol and nonempty set of constants. We define the set of ground terms by induction.

Definition 3.1 *The set $GT(\Sigma)$ of ground terms of the signature Σ is defined as follows:*

1. *Every constant of Σ is a ground term.*
2. *If $t_1, \ldots t_n$ are ground terms and ϕ is a functional symbol of arity n, then $\phi(t_1, \ldots, t_n)$ is also a ground term.*
3. *These are all ground terms.*

To define the notion of random term, we first need the following notion of height for terms.

Definition 3.2 *The* **height** *$h(t)$ of a ground term t is defined to be the number of functional and constant symbols appearing in t.*

For example, supposing that c is a constant symbol, f and g are functional symbols of arity 2 and 1, respectively, it is easy to see that the heights of c, $f(c, c)$, $g(c)$, and $f(g(c), f(c, c))$ are 1, 3, 2, and 6, respectively[1]. Note that if Σ contains symbols for only unary functions and only one constant symbol c, then we can identify each term $t = f_1(\ldots f_n(c) \ldots)$ with the string $w(t) = f_1 \ldots f_n c$. Therefore the height of t is the length of the corresponding string $w(t)$.

Consider the i-th Turing Machine Φ_i which defines a partial mapping from ω into the set $GT(\Sigma)$ of all ground terms. Each Φ_i can be thought as a string over a finite alphabet. Therefore one could say that the **size** of Φ_i is its length. However, for technical reasons we would like to be careful and give the following definition for the **size** of Φ_i. If Φ_i is neither a program of the first type nor the second type, then **the size** of Φ_i is the length of Φ_i. If Φ_i is a program of type t , then the size of Φ_i is the height $h(t)$ of t. If Φ_i is

$$t'(t_1, \ldots, t_{i-1}, \Phi_j, t_{j+1}, \ldots, t_n)$$

then the size of Φ_i is $size(\Phi_j) + h(t'(t_1, \ldots, t_{i-1}, c, t_{j+1}, \ldots, t_n)) - 1$, where c is a constant.

Definition 3.3 *If $\Phi_i(0) = t$, then we say that Φ_i is a* **description** *of t.*

Thus, each term t has infinitely many descriptions. Note that by our convention about the list $\Phi_0, \Phi_1, \ldots$ of Turing machines, every term t has a description of size $h(t)$ since the program t computes term t.

Definition 3.4 *The* **Kolmogorov–Chaitin Complexity of** *t is the size of a minimal description of t, that is $min\{size(\Phi_i) | \Phi_i(0) = t\}$. We denote the Kolmogorov–Chaitin complexity of t by $K(t)$.*

[1]The definition of the height for terms is not a traditional one. It will be clear why we need this type of definition when we construct the Algebra of Random Terms

Let f be function from the set $GT(\Sigma)$ into ω such that for each $t \in GT(\Sigma)$, $f(t) = h(t) - m$, where $m \in \omega$ is a fixed number.

Definition 3.5 *We say that a term t is f–***random** *if the Kolmogorov complexity $K(t)$ of t is greater or equal than $f(t)$, that is $K(t) \geq f(t)$. When $f(t) = h(t)$ for each t, then f-random term is called simply a* **random term***.*

We denote the set of all f–random terms by $RAND_f(\Sigma)$. If $f(t) = h(t)$, then we simply omit the index f and write $RAND$. Now we first show that the set $RAND$ of random terms is an immune set. Then using immunity of $RAND$, we derive several algebraic facts about the set of random terms.

Theorem 3.1 *The set $RAND$ of all random terms is immune.*

Proof. To prove the immunity of the set of random terms, we use the fixed point theorem from computability theory. Suppose that the set $RAND$ is not immune. Hence there exists an infinite effective sequence

$$t_0, t_1, t_2 \ldots$$

of random terms. Thus one can construct an effective sequence

$$\Phi_{i_0}, \Phi_{i_1}, \Phi_{i_2}, \ldots$$

of computable partial functions such that for each $m \in \omega$, the Turing Machine Φ_{i_m} is of size $h(t_m)$ and gives a definition to the term t_m. Because t_m is a random term, note that any Φ_i of size less than $h(t_m)$ is not a description of t_m. Let s_m be the size of Φ_{i_m}. Without lost of generality we also can assume that the effective sequence

$$s_0, s_1, s_2 \ldots$$

is in strictly increasing order. We define the following function ψ. For any $x \in \omega$ find the natural numbers s_t and s_{t+1} such that the size of Φ_x is among integers of the half open interval $[s_t, s_{t+1})$. Define $\psi(x)$ to be s_{t+1}. Clearly ψ is a computable function defined on every $x \in \omega$. Thus, by definition of ψ, we see that $\Phi_x \neq \Phi_{\psi(x)}$ for every x. In other words, the total computable function ψ does not have a fixed point. This contradicts the fixed point theorem. $\Box$

A very similar but more careful construction can be applied to prove the following slightly more general result:

Theorem 3.2 *The set $RAND_f$ of all f–random terms is immune.*$\Box$

Now our goal is to obtain from this theorem several consequences of algebraic nature. We first are interested in the question as whether it is possible to find a method of generating random terms. We give a definition.

Definition 3.6 *A* **generator** *is a system $G = (t_1, \ldots, t_n, F_1, \ldots F_m)$, where $n, m \in \omega$, $t_1, \ldots, t_n$ are terms called* **generating elements***, and $F_1, \ldots, F_m$ are computable functions defined on the set of all ground terms called* **generating rules***.*

Any generator G determines a method of generating terms. We describe the method in the following stagewise procedure:

Stage 0. At this stage generate the set S_G^0 which is the set of generating elements $\{t_1, \ldots, t_n\}$.
Stage t+1. Suppose that we have defined the set S_G^t. Then

$$S_G^{t+1} = S_G^t \bigcup \{F_i(s_1, \ldots, s_m) | i = 1, \ldots n,\ s_1, \ldots, s_m \in S_G^t\}.$$

Informally S_G^{t+1} is obtained by applying the generating rules to the set S_G^t defined at the previous stage.
Now we can define the set S_G to be $\bigcup_i S_g^i$. We say that the generator G **generates** S_G.

Definition 3.7 *The growth function of generator G, denoted by gr_G, is the function $gr_G : \omega \to \omega$ such that $gr_G(i) = card(S_G^i)$ for all $i \in \omega$.*

Proposition 3.1 *The growth function of any generator G can be majorized by a primitive recursive function.*

Proof. Indeed, suppose that $F_1, \ldots F_m$ are all generating rules of $G = (t_1, \ldots, t_n, F_1, \ldots F_m)$ such that arity of F_i is q_i. Define the following function g: $g(0) = n$, $g(i+1) = g(i) + g(i)^{q_1} + \ldots + g(i)^{q_m}$. Clearly g is a primitive recursive function majorazing the growth function gr_G. □

Definition 3.8 *A set $S \subset GT(\Sigma)$* **has a generator** *if for some generator G we have $S = S_G$.*

Example. Every infinite computable, or equivalently decidable, subset $S \subset GT(\Sigma)$ has a generator. Indeed, let $t_0, t_1, \ldots$ be an effective sequence of all terms from S such that $h(t_i) \leq h(t_{i+1})$, for all $i \in \omega$. Define the following function g: if $t \notin S$, then $g(t) = t_0$; if $t = t_i$, then $g(t_i) = t_{i+1}$.

The next proposition generalizes the above example by showing that computably enumerable subsets of $GT(\Sigma)$ are the only ones which have generators.

Proposition 3.2 *An infinite subset S of the set $GT(\Sigma)$ has a generator if and only if S is computably enumerable.*

Proof. First, note that S_G is computably enumerable for every generator G. Hence, if S is not computably enumerable, then S does not have a generator.

Suppose that S is infinite and computably enumerable. There exists an infinite computable subset S' of S. Let $s_0, s_1, \ldots$ be an effective sequence of all terms from S' such that $h(s_i) \leq h(s_{i+1})$ for all $i \in \omega$.

Define the following functions F_1 and F_2: $F_1(x) = x$ if $x \notin S'$; $F_1(x) = s_{i+1}$ of $x = s_i$. $F_2(x) = x$ if $x \notin S'$; $F_2(x) = t_i$ of $x = s_i$, where $t_0, t_1, \ldots$ is an effective sequence of all elements from S. Therefore, (s_0, F_1, F_2) is a generator which generates S. □

Corollary 3.1 *$RAND$ ($RAND_f$) does not have a generator.* □

In fact, a stronger result can be stated about $RAND$ ($RAND_f$). We need a definition.

Definition 3.9 *A set $S \subset GT(\Sigma)$ is* **locally finite** *if any subset of S which has a generator is finite.*

Thus from the proof of the previous proposition we get the following corollary.

Corollary 3.2 *$S \subset GT(\sigma)$ is locally finite if and only if S is immune. Hence $RAND$ ($RAND_f$) is locally finite.* □

Thus, the immunity of the set of random terms does not allow one to find a method of generating random terms. The reader familiar with the basics of universal algebra can easily notice that in order to develop an algebraic theory for random terms we have tried to use the notions and ideas from the theory of finitely generated algebras (generator, finitely generated algebra, locally finite algebra, etc. [8]). However, the last corollary can be interpreted that one can not develop a rich algebraic theory on the set $RAND_f$ unless one is interested in locally finite algebras over random terms or finds some new ideas for investigating randomness by means of (possibly) infinite algebras. In the next section we propose another view on the random universe and define an infinite algebra which we call the Algebra of Random Terms.

4 Algebra of Random Terms

Consider the set of all ground terms $GT(\Sigma)$. Since our basic interest is in random terms, we can think of each random term as an individual unique object while we can think of the set of nonrandom terms U as an object representing nonrandomness. In other words, we can look at the set of all ground terms as a domain every object of which is either a random term or the object U obtained by identifying all nonrandom terms. Now we explain this formally. Consider the following equivalence relation $eq(RAND)$ on the set of all ground terms:

$$(t,s) \in eq(RAND) \text{ iff } (t \in RAND \to t = s) \bigvee (t \notin RAND \to s \notin RAND).$$

It follows we can consider the set whose elements are the equivalence classes of $eq(RAND)$. We call this set the pseudo-random domain and denote this domain by $P\text{-}RAND$. Thus, we have defined $P\text{-}RAND$ to be the set

$$\{t | t \in RAND\} \bigcup \{U\}, \quad \text{where } U = \{t | \ t \text{ is not random }\}.$$

Lemma 4.1 *If t is not random, then any term containing t is also not random.*

Proof. Since t is not random, there is a description P of t such that the size of P is strictly less than $h(t)$. Let $t' = t''(t_1, \ldots, t, \ldots, t_n)$ be a term containg t as a subterm. Consider the following program P':

$$t''(t_1, \ldots, P, \ldots, t_n).$$

The meaning of this program is as follows: [Begin by constructing the term t'. As soon as the occurrence of t in t' is reached apply the description P for the term t]. Thus, since $size(P) < h(t)$, we have that the size of the program P' is $size(P) + ht''(t_1, \ldots, c, \ldots, t_n) - 1$ and less than the height of term t'. Hence t' is not random. □

Definition 4.1 *We say that a function* $\phi : GT(\Sigma)^m \to GT(\Sigma)$ **respects the pseudo-random domain** $P\text{-}RAND$ *if for any pair of m–tuples $(t_1, \ldots, t_m)$, $(s_1, \ldots, s_m)$ the condition $(t_1, s_1), \ldots, (t_m, s_m) \in eq(RAND_f)$ implies that $(\phi(t_1, \ldots, t_m), \phi(s_1, \ldots, s_m) \in eq(RAND_f)$.*

The next lemma shows that any function symbol $\phi \in \Sigma$ respects the pseudo-random domain $P\text{-}RAND$. In terms of universal algebra this means that $eq(RAND)$ is a congruence relation on the absolutely free algebra

$$(GT(\Sigma), \phi_1^{l_1}, \ldots, \phi_m^{l_m}, c_1, \ldots, c_k).$$

Lemma 4.2 *Every functional symbol $\phi \in \Sigma$ respects the pseudo-random domain $P\text{-}RAND$.*

Proof. Let ϕ be in Σ of arity m. Let $(t_1, \ldots, t_m)$, $(s_1, \ldots, s_m) \in GT(\Sigma)^m$ be such that $(t_1, s_1), \ldots, (t_m, s_m) \in eq(RAND)$. If each t_i is random, then by the definition of $eq(RAND)$, we have $t_i = s_i$, and hence $(\phi(t_1, \ldots, t_m), \phi(s_1, \ldots, s_m)) \in eq(RAND)$. Suppose t_i is not random. Then s_i is also not random. Hence by Lemma 4.1 the terms $\phi(t_1, \ldots, t_m)$ and $\phi(s_1, \ldots, s_m)$ do not belong to $RAND$. It follows that ϕ respects the pseudo-random domain $P\text{-}RAND$. □

Definition 4.2 *The* **Algebra of Random Terms (***ART***)** *is the pseudorandom domain together with all functional and constant symbols from Σ, that is*

$$ART = (P\text{-}RAND, \phi_1, \ldots, \phi_m, c_1, \ldots, c_k).$$

Note that this ART is a correctly defined algebra due to Lemma 4.2. In algebraic terms ART is the homomorphic image of the free algebra $GT(\Sigma)$ under homomorphism $t \to \{s | (t,s) \in eq(RAND)\}$. Now we can easily prove the following theorem.

Theorem 4.1 *The algebra of random terms ART is finitely generated infinite algebra with computably enumerable equality relation.*

Proof. Indeed, the generators of ART are the equivalence classes containing the constant symbols $c_1, \ldots c_k$. The algebra is infinite since the set of random terms is infinite and each $eq(RAND)$–class containing a random term is singleton. The equality relation in ART is computably enumerable since the set of nonrandom terms is computably enumerable. □

Corollary 4.1 *The pseudorandom domain has a generator.*

Proof. Indeed, the generator of the domain is

$$G = (eq(c_1), \ldots, eq(c_k), \phi_1, \ldots, \phi_m),$$

where $eq(x)$ is the equivalence class containing x under $eq(RAND)$. □

Thus, we can "generate" the set of random terms. A "procedure of generating" the set of random terms can be described as follows. Take the generator $G = (eq(c_1), \ldots, eq(c_k); \phi_1^{l_1}, \ldots, \phi_k^{l_k})$. Begin by generating the sets $S_G^0, S_G^1, S_G^2, \ldots$ not applying the generating rules to nonrandom terms. This procedure is effective provided that there is an oracle which decides the set of nonrandom terms. Thus informally one can say that modulo nonrandom universe the set of random terms has a generator.

Now we are ready to prove the theorem about impossibility of specifying the Algebra of Random Terms. In our proof we use three lemmas. Our first lemma is a known lemma, probably first proved by Malcev [12], which states a condition sufficient for the decidability of the word problem for initial algebras. The second lemma is a technical lemma which reduces our study of algebras of finite signature to algebras with infinitely many unary functions. The proof of the third lemma extends a proof from [10] of a similar lemma applied to Algebra of Random Terms.

Theorem 4.2 *The Algebra of Random Terms ART can not be algebraically specified.*

Proof. We need the notion of residually finite algebra for the first lemma. We say that an algebra $\mathcal{A}$ is **residually finite** if for any two distinct elements a and b of $\mathcal{A}$ there is a finite homomorphic image of $\mathcal{A}$ in which the images of a and b are distinct.

Lemma 4.3 *If a $\mathcal{A}$ is the initial algebra for a finite set of equations E and is residually finite, then the word problem for $\mathcal{A}$ is decidable.*

Proof of the Lemma. Let $a_1, \ldots, a_n$ be generators of $\mathcal{A}$. Consider an effective sequence $\mathcal{A}_0, \mathcal{A}_1, \ldots$ of all finite algebras generated by $a_1, \ldots, a_n$ which satisfy E. Since $\mathcal{A}$ is an initial algebra each $\mathcal{A}_i$ is a homomorphic image of $\mathcal{A}$. Let $x, y \in A$. Consider the following two procedures.

Procedure 1. Compute all elements in $\mathcal{A}$ equal to x.

Procedure 2. For each i check whether the image of x in $\mathcal{A}_i$ does not equal to the image of y in $\mathcal{A}_i$.

If $x = y$ in $\mathcal{A}$, then Procedure 1 halts. If $x \neq y$ in $\mathcal{A}$, then there exists an i such that the image of x will be distinct from the image of y in $\mathcal{A}_i$. Thus, we can check whether x equals to y in $\mathcal{A}$ or not. Hence the word problem in $\mathcal{A}$ is decidable. □

The next rather technical lemma is of a general character. Let f be an atomic operation on $\mathcal{A}$ of arity n. A **transition of** $\mathcal{A}$ is any of the mappings $f(a_1, \ldots, a_{n-1}, x)$, …, $f(x, a_1, \ldots, a_{n-1})$, where $a_1, \ldots, a_{n-1}$ are elements of the algebra $\mathcal{A}$.

Lemma 4.4 *Let $\mathcal{A}$ be an algebra and let η be an equivalence relation on $\mathcal{A}$. The relation η is a congruence relation of $\mathcal{A}$ if and only if any transition of $\mathcal{A}$ respects η.*

Proof. It is clear that if η is a congruence relation, then any transition of $\mathcal{A}$ respects η. Now suppose that every transition respects η. Consider any n–tuple of pairs (a_1, b_1), ..., (a_n, b_n) from η. Then

$$(f(a_1, a_2, \ldots, a_n), f(b_1, a_2, \ldots, a_n)) \in \eta$$

and

$$(f(b_1, a_2, a_3, \ldots, a_n), f(b_1, b_2, a_3, \ldots, a_n)) \in \eta.$$

Hence $(f(a_1, a_2, \ldots, a_n), f(b_1, b_2, a_3, \ldots, a_n)) \in \eta$, etc. It follows by induction and transitivity that $(f(a_1, a_2, \ldots, a_n), f(b_1, b_2, \ldots, b_n)) \in \eta$. □

Our third lemma shows that any expansion of the Algebra of Random Terms is a residually finite algebra.

Lemma 4.5 *Let $f_1, \ldots, f_n$ be functions which respect the pseudorandom domain. Then the expanded algebra $(ART, f_1, \ldots, f_n)$ is residually finite.*

Proof of the Lemma. Consider the expanded algebra

$$\mathcal{A}' = (ART, f_1, \ldots, f_n).$$

We can effectively list all the transitions of this expanded algebra. Let $F_0, F_1, \ldots$ be an effective list of the transitions. Define a new universal algebra called **the transition algebra of $\mathcal{A}'$**:

$$Tr(\mathcal{A}') = (P\text{-}RAND : F_0, F_1, F_2, \ldots).$$

By the above lemma it suffices to prove that the transition algebra $Tr(\mathcal{A}')$ is residually finite.

Let t_1, t_2 be two distinct random terms. We will show that there exists a set S with the following three properties:

1. The set S is finite and contains only random terms.
2. The terms t_1 and t_2 belong to S.
3. Every transition F_i respects the equivalence relation:

$$eq(S) = \{(x, y) | x, y \in GT(\Sigma) \setminus S\} \bigcup \{(x, y) | x = y\}.$$

If a such S exists, then the mapping $h : t \to \{s | (t, s) \in eq(S)\}$ will be a homomorphism from ART to a finite algebra in which $h(t_1) \neq h(t_2)$.

In order to prove that there exists a set S with the above three properties we need to make several notes. Take a nonrandom term $u \in U$ and any transition F_i. Let S' be any finite subset of $RAND$. If $F_i(u) \notin S'$, then since F_i respects the pseudodomain of random terms the set $\{t | F_i(t) \in S'\}$ is a subset of $RAND$. This set is computable, and hence finite since $RAND$ is immune. If $F_i(u) \in S'$, then $F_i(s) = F_i(u)$ for all $s \in U$ (F_i respects the pseudodomain of random terms), and so the set $\{t | F_i(t) \neq F_i(u)\}$ is a computable subset of $RAND$, and hence finite. We also note the following fact: A transition F_i respects the equivalence relation

$$eq(S) = \{(x, y) | x, y \in GT(\Sigma) \setminus S\} \bigcup \{(x, y) | x = y\}.$$

if and only if the following conditions are satisfied:

1. $F_i(u) \in S$ if and only if $F_i(t') = F_i(u)$, for any term $t' \notin S$.
2. $F_i(u) \notin S$ if and only if $F_i(t') \notin S$ for any term $t' \notin S$.

Now we show how to construct S in such a way that the mapping

$$h : t \to \{s|(t,s) \in eq(S)\}$$

is a homomorphism from ART to a finite algebra in which $h(t_1) \neq h(t_2)$. Our construction of S is a stagewise construction, that is at stage j we have a finite set S_j of random terms. We will put S to the union of all S_js.

Stage 0. Put $S_0 = \{t_1, t_2\}$. Clearly $S_0 \subset RAND$.

Stage j+1. Suppose that S_j has been constructed and $S_j \subset RAND$. Consider the transitions $F_0, \ldots, F_{j+1}$. For each $i \leq j+1$, consider $F_i(u)$.

Case 1. Suppose that $F_i(u) \notin S_j$. In this case set

$$S_{j+1,i} = S_j \bigcup \{t|F_i(t) \in S_j\}.$$

Case 2. Suppose that $F_i(u) \in S_j$. In this case set

$$S_{j+1,i} = S_j \bigcup \{t|F_i(t) \neq F_i(u)\}.$$

Define S_{j+1} to be $S_{j+1,0} \bigcup \ldots \bigcup S_{j+1,j+1}$.

Now we can define the set S to be the union of all sets S_j, that is $S = \bigcup_j S_j$.

Now by the previous remarks we see that S is a finite set whose elements are random terms. Therefore there exists a stage j_0 such that $S = S_{j_0}$. It is clear that the terms t_1 and t_2 belong to S. We have to show that every transition F_i respects the equivalence relation:

$$eq(S) = \{(x,y)|x, y \in GT(\Sigma) \setminus S\} \bigcup \{(x,y)|x = y\}.$$

It suffices to prove that if s does not belong to S, then $(F_i(u), F_i(s)) \in eq(S)$. Consider any stage $j \geq j_0$. Suppose that $F_i(u) \notin S_j$. Then $F_i(s) \notin S_j$, otherwise $s \in S_j$ and hence $S_{j_0} \neq S_j$. Similarly, if $F_j(u) \in S_j$, then $F_j(s) = F_j(u)$, otherwise $s \in S_j$ and hence $S_{j_0} \neq S_j$. Thus, the homomorphism h defined by $h : t \to \{s|(t,s) \in eq(S)\}$ maps ART onto a finite algebra. In this finite algebra $h(t_1) \neq h(t_2)$. The lemma is proved.

Now it is clear the above three lemmas that the algebra of random terms ART can not be specified. The theorem is proved.

Finally, we would like to add that similar as to ART one can define the Algebra of f–Random Terms. This will slightly generalize Theorem 4.1:

Theorem 4.3 *The Algebra of f–Random Terms can not be algebraically specified.* □

Open Question and Acknowledgement. We would like to end this paper with the following question. Instead of considering initial algebras defined by equations, one can consider initial algebras defined by conditional equations, that is by finite set of formulas of the form

$$\forall \bar{x}(t_1(\bar{x}) = s_1(\bar{x}) \& \ldots \& t_n(\bar{x}) = s_n(\bar{x}) \to t(\bar{x}) = s(\bar{x}).$$

These are logic programs of a functional language. Thus one can define that an algebra $\mathcal{A}$ is **specified** if the algebra possesses an expansion which is the initial algebra defined by a finite set of conditional equations. Thus, our question is: *Can the Algebra of Random Terms be specified by conditional equations?*. If an answer were positive, then this example would be the first example of an algebra which is specified by conditional eqations but not specified by equations. On the other hand, if an answer were negative, then ART would be the first example which could not be specified by conditional equations. This would give a negative answer to the question of Bergstra and Tucker about specifying any finitely generated computably enumerable algebra by conditional equations [3].

Finally the author would like to thank Cristian Calude for his interest in this research, careful reading and helpful suggestions on improvements of the paper. The author also thanks Azat Arslanov for discussions of the initial parts of this research.

References

[1] J.A. Bergstra, J.V.Tucker, The Completencess of The Algebraic Specification Methods for Computable Data Types, Inform. and Control, 54, 1987.

[2] J.A. Bergstra, J.V.Tucker, Initial and Final Algebra Semantics for Data Type Specifications: Two characterization Theorems, SIAM J. Comput. 12, 1983.

[3] J.A. Bergstra, J.V.Tucker, Algebraic Specifications of Computable and Semicomputable Data Types, Theoretical Comp. science, 50, 1987.

[4] J.A. Bergstra, J. Heering, P. Klint, Algebraic Specifications, ACM Press, New York, 1989.

[5] C.Calude, Information and Randomness, An Algorithmic Perspective, Springer-Verlag, EATCS series, 1994.

[6] C. Calude, Algorithmic Information Theory: Open Problems, Journal of Universal Computer Science, 2, 1996.

[7] G. Chaitin, Information, Randomness, and Incompleteness, Series in Computer Science –Vol 8, World Scientific Publishing Co., 1987.

[8] G. Gratzer, Universal Algebra, Van Nostrand, Princeton, NJ, 1968.

[9] N.K. Kassimov, B.M. Khoussainov B.M, Logic Specifications and Effective Representations of Abstract Data Types, Reprint, Tashkent University Press, 1991.

[10] N.K. Kassimov, On Finitely Approximable and R.E. Representable Algebras, Algebra and Logic, 26, No 6,1986.

[11] Logic Notebook (Open questions in Logic), Novosibirsk University Press, editors Yu.Ershov and S. Goncharov, 1989.

[12] A.I. Malcev, Constructive Algebras, Uspekhi Matem. Nauk, 16, No 3, 1961.

[13] M.O. Rabin, Computable Algebra, General Theory and the Theory of Computable Fields, Trans. Amer. Math. Soc., 98, 1960.

[14] H. Rogers, Theory of Recursive Function and Effective Computability, New York, 1967.

[15] V. Uspensky, Complexity and Entropy: An introduction to Kolmogorov Complexity, in O.Watanabe ed., Kolmogorov Complexity and Computational Complexity, Springer–Verlag, Berlin, 1992.

[16] M. Wirsing, Algebraic Specifications, Handbook of Theoretical Computer Science, volume B, 1990, The MIT Press.