ON FINITELY PRESENTED EXPANSIONS OF COMPUTABLY ENUMERABLE SEMIGROUPS

DENIS R. HIRSCHFELDT AND BAKHADYR KHOUSSAINOV

Abstract. Every computable universal algebra has a finitely presented expansion, but there are examples of finitely generated, computably enumerable universal algebras with no finitely presented expansions. It is natural to ask whether such examples can be found in well-known classes of algebras such as groups and semigroups. In this paper, we build an example of a finitely generated, infinite, computably enumerable semigroup with no finitely presented expansions. We also discuss other interesting computability theoretic properties of this semigroup. This paper is based on the invited talk given by B. Khoussainov at the Mal'cev meeting 2011 dedicated to the 60th birthday of Professor Sergei Goncharov.

1. Introduction

The purpose of this paper is to construct a semigroup with certain properties that are interesting from the point of view of the theory of effective universal algebra. We begin by outlining some of the basic notions of this theory. In subsection 1.3, we discuss the most significant property of our semigroup, namely that it is a solution within the class of semigroups of the equational specification problem of Bergstra and Tucker [1, 2] and Goncharov [4].

1.1. **Preliminaries.** A *universal algebra*, or simply an *algebra*, for short, is a structure $\mathcal{A}$ of the form $(A; f_1, \dots, f_n, c_1, \dots, c_m)$, where A is a nonempty set called the *domain* of $\mathcal{A}$, each f_i is a total function $A^{k_i} \to A$ called a *basic operation* of arity k_i, and each c_j is a *distinguished element* (or a *constant*) of $\mathcal{A}$. Constants can be viewed as functions of arity 0, so we will often suppress particular mention of them unless we wish to highlight their role. The *signature* of $\mathcal{A}$ is the sequence $f_1, \dots, f_n, c_1, \dots, c_m$ of symbols representing the operations and constants. We always assume this signature contains a

The first author was partially supported by the National Science Foundation of the United States, grants DMS-0801033 and DMS-1101458.

The second author was partially supported by the Marsden Fund of the Royal Society of New Zealand.

function symbol of arity greater than 0. Note that we used the sequence $f_1, \dots, f_n, c_1, \dots, c_m$ in two ways: one as representing operations and elements of the algebra and the other as a sequence of symbols. We will sometimes do the same below, in cases where which meaning is being used is clear from the context. An *expansion* of the algebra $\mathcal{A} = (A; f_1, \dots, f_n, c_1, \dots, c_m)$ is any algebra of the form $\mathcal{A}' = (A; f_1, \dots, f_n, h_1, \dots, h_k, c_1, \dots, c_m)$. If the h_i are all constants, then we say that $\mathcal{A}'$ is an *expansion by constants* of $\mathcal{A}$. For background on universal algebras, see [7].

A countable algebra $\mathcal{A} = (A; f_1, \dots, f_n)$ is *computable* if the domain A is a computable set and each of the operations f_i is a computable function. For simplicity, when we are given such a computable algebra, we may assume that A is a subset of ω; indeed, if A is infinite, we may assume that $A = \omega$. However, when convenient we may also take A to be a computable subset of any space that can be naturally identified with ω via a standard coding, such as the set of finite binary strings or the set of sentences in a computable language.

There are many natural examples of computable algebras, for instance arithmetic $(\omega; 0, +, \times)$. There are many other important examples, however, in which the domain and basic operations can be made computable only if we are willing to identify elements of the domain via a computably enumerable equivalence relation. Examples include the Lindenbaum Boolean algebras of computably enumerable first order theories (such as Peano arithmetic) and finitely presented groups and semigroups. To capture this class of algebras, we have the notion of a computably enumerable algebra, which is our main object of study in this paper. To define it, we begin with a few auxiliary definitions.

Let E be an equivalence relation on a set B. Denote the equivalence class of x by $[x]_E$, and the set of all such equivalence classes by B/E. We say that a function $F : B^n \to B$ *respects* E if whenever $[x_i]_E = [y_i]_E$ for all $i < n$, we have $[F(x_0, \dots, x_{n-1})]_E = [F(y_0, \dots, y_{n-1})]_E$. In this case, the function $f : (B/E)^n \to B/E$ *induced* by F is the one defined by $f([x_1]_E, \dots, [x_n]_E) = [F(x_1, \dots, x_n)]_E$.

Let $\mathcal{B} = (B; F_1, \dots, F_n)$ be an algebra. An equivalence relation E on B is called a *congruence relation* on $\mathcal{B}$ if every basic operation of $\mathcal{B}$ respects E. For a congruence relation E on $\mathcal{B}$, let $\mathcal{B}/E$ be the quotient of $\mathcal{B}$ by E, that is, the algebra $(B/E; f_1, \dots, f_n)$, where f_i is the function induced by F_i.

Definition 1.1. A countable algebra is *computably enumerable* (*c.e.*) if it is of the form $\mathcal{B}/E$ for a computable algebra $\mathcal{B}$ and a c.e. congruence relation E on the domain of $\mathcal{B}$.

When we have a fixed c.e. algebra $\mathcal{A} = \mathcal{B}/E$, we often drop the subscript and write $[x]$ in place of $[x]_E$. We also often speak of the elements of the domain of $\mathcal{B}$ as elements of $\mathcal{A}$, identifying x with $[x]$, and speaking of E as the *equality relation* of $\mathcal{A}$. It is easy to see that we can always assume that the domain of $\mathcal{B}$ is infinite, and hence that it is in fact ω (or any other infinite set we identify with ω, such as the set of finite binary strings).

Although we will not study it in this paper, we can also define the notion of a *co-computably enumerable algebra*, which is defined as in Definition 1.1, but with E being co-c.e. A typical example of a co-computably enumerable algebra is the group generated by a finite number of computable permutations of ω. If g and g' are elements of this group then their nonequality is confirmed by the existence of an n such that $g(n) \neq g'(n)$. In Definition 1.1, if E is computable, then $\mathcal{B}/E$ is a computable algebra. Furthermore, an algebra is computable if and only if it is both computably enumerable and co-computably enumerable. For a modern treatment of computable algebra and computable model theory, see for instance [5].

We will need the following notion of homomorphism between algebras.

Definition 1.2. Let $\mathcal{B} = (B; f_1, \dots, f_m)$ and $\mathcal{C} = (C; g_1, \dots, g_m)$ be algebras with the same signature. Let k_i be the arity of f_i (and hence also of g_i). A *homomorphism* from $\mathcal{B}$ to $\mathcal{C}$ is a map $h : B \to C$ such that for each $i = 1, \dots, m$ and each $b_1, \dots, b_{k_i} \in B$, we have $h(f_i(b_1, \dots, b_{k_i})) = g_i(h(b_1), \dots, h(b_{k_i}))$. If h is surjective, then we say that $\mathcal{C}$ is a *homomorphic image* of $\mathcal{B}$. An *isomorphism* is a homomorphism that is both injective and bijective.

Note that if h is a homomorphism from $\mathcal{B}$ to $\mathcal{C}$ then each constant of $\mathcal{B}$ is mapped to the corresponding constant of $\mathcal{C}$.

There is a one-to-one correspondence between the homomorphisms of an algebra $\mathcal{B}$ and its congruence relations: For any congruence relation E on $\mathcal{B}$, we have a homomorphism from $\mathcal{B}$ to $\mathcal{B}/E$ defined by $b \mapsto [b]_E$. Conversely, any homomorphism h from $\mathcal{B}$ determines the congruence relation $\{(x, y) \mid h(x) = h(y)\}$. From the definitions above it is clear that every c.e. algebra is a homomorphic image of a computable algebra.

1.2. **Finitely presented algebras.** Let $\sigma = f_1, \dots, f_n, c_1, \dots, c_m$ be a signature with at least one constant symbol. Let k_i be the arity of f_i. Let V be an infinite set of variables. The set T of *terms* of this signature is defined inductively as follows. Each constant symbol and each variable is a term. If $t_1, \dots, t_{k_i}$ are terms, then so is the expression $f_i(t_1, \dots, t_{k_i})$. We call a term a *ground term* if it contains no variables. The set T_G of ground terms can be turned into an algebra with signature σ as follows. The interpretation of each c_i is c_i itself. For each function symbol f_i and tuple of ground terms $(t_1, \dots, t_{k_i})$, let the value of the interpretation of f_i on this tuple be $f_i(t_1, \dots, t_{k_i})$. We call the resulting algebra the *term algebra* and denote it by $\mathcal{T}_G$.

Let $\mathcal{A} = (A; F_1, \dots, F_n, a_1, \dots, a_m)$ be an algebra with signature σ. Fix a function $s : V \to A$, which we think of as an interpretation of the variables in A. We extend s to a *interpretation* $i(t)$ of each term $t \in T$ in $\mathcal{A}$ by induction as follows. First, for each variable x, let $i(x) = s(x)$, and for each constant symbol c_j, let $i(c_j) = a_j$. In the inductive step, for each basic operation f_j, let $i(f_j(t_1, \dots, t_{k_j})) = F_j(i(t_1), \dots, i(t_{k_j}))$. Note that the value of $i(t)$ depends only on the values of s on variables occurring in t. In particular, if $t \in T_G$ then $i(t)$ does not depend on s, so we will use the notation $i(t)$ for ground terms t without specifying a function s. Note also that in the term algebra $\mathcal{T}_G$, we have $i(t) = t$ for all $t \in T_G$.

We say that $\mathcal{A}$ is *generated by its constants* if every element of $\mathcal{A}$ is the interpretation of some ground term, or in other words, if every element of $\mathcal{A}$ can be obtained from its constants by some chain of basic operations. We say that $\mathcal{A}$ is *freely generated* by its constants if, in addition, whenever t, t' are different ground terms, $i(t) \neq i(t')$.

It is not hard to see that the term algebra $\mathcal{T}_G$ has the following properties (see for instance [7]).

(1) The algebra $\mathcal{T}_G$ is generated by its constants and computable.
(2) Any algebra with signature σ that is generated by its constants is a homomorphic image of $\mathcal{T}_G$.
(3) An algebra with the signature σ is freely generated by its constants iff it is isomorphic to $\mathcal{T}_G$.

Definition 1.3. An algebra $\mathcal{A}$ is *finitely generated* if some expansion by constants $\mathcal{A}'$ of $\mathcal{A}$ is generated by the constants of $\mathcal{A}'$. We call these constants *generators.*

Let $\mathcal{A}$ and $\mathcal{A}'$ be as in the above definition, where $\mathcal{A}'$ has signature σ. Let $i(t)$ be the interpretation of the term t in $\mathcal{A}'$. The *word problem* for $\mathcal{A}$ is the set $\{(t, t') \in T_G \times T_G \mid i(t) = i(t')\}$. Although the word problem for $\mathcal{A}$ depends on the choice of $\mathcal{A}'$ (i.e., on the choice of generators), it is easy to see that the Turing degree of the word problem is independent of this choice. If this degree is $\mathbf{0}$, then we say that the word problem for $\mathcal{A}$ is *decidable.* If $\mathcal{A} = \mathcal{B}/E$ is a c.e. algebra, where $\mathcal{B}$ is a computable algebra and E is a congruence relation on $\mathcal{B}$, then the word problem for $\mathcal{A}$ is Turing equivalent to E. Thus, in this case the word problem for $\mathcal{A}$ is decidable iff E is computable.

We now define the concept of equational specification.

Definition 1.4. An *atomic formula* or *equation* is a formula of the form $t = t'$, where t and t' are terms (which might contain variables). An *equational specification* is a finite set of equations.

We say that an algebra $\mathcal{A}$ *satisfies* an equational specification S if for all $t = t' \in S$ and all functions $s : V \to A$, we have $i(t) = i(t')$ in $\mathcal{A}$, where i is defined as above using s.

It is easy to see that if an algebra satisfies a given equational specification S, then so do all its homomorphic images, and that we can computably check whether a given finite algebra satisfies S.

Let S be an equational specification. Let $\mathrm{Cn}(S)$ be the set of all equations that can be deduced from S (in the usual sense of first-order logic). Note that $\mathrm{Cn}(S)$ is c.e. Let

$$E(S) = \{(t, t') \in T_G \times T_G \mid t = t' \in \mathrm{Cn}(S)\},$$

and let

$$\mathcal{T}(S) = \mathcal{T}_G / E(S).$$

It is not hard to see that the following facts hold (see for instance [11]).

(1) The algebra $\mathcal{T}(S)$ is generated by its constants.
(2) The relation $E(S)$ is c.e., so $\mathcal{T}(S)$ is a c.e. algebra.
(3) The algebra $\mathcal{T}(S)$ satisfies S.
(4) Any algebra $\mathcal{A}$ that is generated by its constants and satisfies S is a homomorphic image of $\mathcal{T}(S)$.
(5) If $\mathcal{B}$ is generated by its constants and satisfies S, and property (4) remains true with $\mathcal{B}$ in place of $\mathcal{T}(S)$, then $\mathcal{B}$ is isomorphic to $\mathcal{T}(S)$.

The following is one of our central definitions.

Definition 1.5. An algebra $\mathcal{A}$ is *finitely presented* if there are an expansion by constants $\mathcal{A}'$ of $\mathcal{A}$ and an equational specification S (in the language corresponding to the signature of $\mathcal{A}'$) such that $\mathcal{A}'$ is isomorphic to $\mathcal{T}(S)$.

Examples of finitely presented algebras include finitely presented groups, which have been extensively studied in group theory. By property (2) above, finitely presented algebras give us natural examples of c.e. algebras.

Note that if $\mathcal{A}$ is finitely presented then it is finitely generated. Let S be as in the above definition. Then it is easy to see that the word problem for $\mathcal{A}$ is Turing equivalent to the word problem for $\mathcal{T}(S)$. In particular, $\mathcal{A}$ has decidable word problem iff $\mathcal{T}(S)$ does, i.e., iff $E(S)$ is computable.

1.3. **The equational specification problem.** Bergstra and Tucker [1] gave the following simple example: The algebra $(\omega; 0, S, 2^x)$, where S is the successor function, is not finitely presented, but its expansion $(\omega; 0, S, 2^x, +, \times)$ is finitely presented; its equational presentation is given by the following set of equations:

$$x + 0 = x,\ x + S(y) = S(x + y),\ x \times 0 = 0,$$
$$x \times S(y) = x \times y + x,\ 2^0 = S(0),\ 2^{S(x)} = 2^x \times S(S(0)).$$

(See [1] for proofs of these facts.) This method of expanding signatures turns out to be quite general. Indeed, Bergstra and Tucker [1] proved that any computable algebra has a finitely presented expansion. These observations led Bergstra and Tucker [1, 2], and independently Goncharov [4], to ask the following question, known as the *equational specification problem*: Does every finitely generated c.e. algebra have a finitely presented expansion?

Kasymov [8] answered this question by constructing an example of a finitely generated c.e. algebra with no finitely presented expansion. A similar example, using Kolmogorov complexity, was constructed by Khoussainov [9]. These examples of c.e. algebras were built for this specific purpose, and their signatures consist of unary operation symbols only. The methods that construct such examples do not carry over to to build, for instance, semigroups with no finitely presented expansions. Hence, it is natural to attempt to find such examples in the classes of semigroups and groups. In the next section, we build an example of a finitely generated, infinite c.e. semigroup. We then study properties of this semigroup, and in particular, prove that it has no finitely presented expansions. (The semigroup we build has an identity element, and hence is in fact a monoid.)

We note that the upcoming paper [10] gives an example of a finitely generated c.e. *group* with no finitely presented expansions. The proof uses Golod-Shafarevich algebras and their properties [6].

2. Construction of the semigroup $\mathcal{A}(Z)$

In this section, we construct an infinite, finitely generated c.e. semigroup whose properties we will analyze in the following section.

Recall that a semigroup is an algebra with exactly one associative binary operation. Let $\{x, y\}^\star$ be the set of all finite strings over the alphabet $\{x, y\}$. We will denote the empty string by λ, and the length of a string u by $|u|$. Let $\cdot$ be the concatenation operation on strings. Then the algebra

$$\mathcal{A} = (\{x, y\}^\star; \cdot)$$

is a semigroup, namely the free semigroup on the generators x and y. Every semigroup with two generators is a homomorphic image of $\mathcal{A}$, which is clearly a computable algebra.

We will construct our semigroup as a quotient of $\mathcal{A}$ by an appropriate congruence relation. We need a few definitions. A string $v \neq \lambda$ is a *substring* of u if we can write u in the form $u = u_1 v u_2$.

Definition 2.1. Let Z be a subset of $\{x, y\}^\star$. We say that a string u *realizes* Z if u contains a substring from Z. Otherwise, we say that u *avoids* Z. We denote the set of all strings that realize Z by $R(Z)$.

It is clear that $Z \subseteq R(Z)$ for all Z. With each $Z \subseteq \{x, y\}^\star$ we associate the equivalence relation

$$\eta_Z = \{(p, q) \mid p = q \ \vee \ p, q \in R(Z)\}.$$

Each equivalence class of η_Z is either a singleton or $R(Z)$. Moreover, it is not hard to see that η_Z is a congruence relation on the free semigroup $\mathcal{A}$; that is, if $(u_1, u_2), (v_1, v_2) \in \eta_Z$ then $(u_1 v_1, u_2 v_2) \in \eta_Z$. We denote the quotient semigroup $\mathcal{A}/\eta_Z$ by $\mathcal{A}(Z)$. This quotient semigroup is finitely generated by the elements $[x]$ and $[y]$. Furthermore, the following facts clearly hold.

Lemma 2.2. (1) *If Z is c.e. then $\mathcal{A}(Z)$ is a c.e. semigroup.*

(2) *If Z is computable then $\mathcal{A}(Z)$ is a computable semigroup.*

(3) *The semigroup $\mathcal{A}(Z)$ is finite iff $R(Z)$ is cofinite.*

One of our goals is to ensure that the semigroup $\mathcal{A}(Z)$ we build is infinite. To do so, we need only make Z sufficiently sparse. Cenzer, Dashti, and King [3] showed that there is a computable sequence $l_0, l_1, \dots$ such that if Z contains at most one string of each length l_i, and no strings of any other length, then there is an infinite binary string that avoids all strings in Z, and hence $R(Z)$ is coinfinite. Miller [13] later showed that we can take $l_i = i + 5$, and we use this fact for simplicity of notation. Thus we have the following lemma.

Lemma 2.3. *Let $Z \subseteq \{x, y\}^\star$. If for each k there are at most k many strings of length $\leqslant k + 4$ in Z, then $R(Z)$ is coinfinite.*

Proof. We can list the elements $v_0, v_1, \dots$ of Z so that $|v_i| \geqslant i + 5$ for all i. Since avoiding a string implies avoiding all its extensions, there is a set Y containing exactly one string of each length of the form $i + 5$ such that any string that avoids Y also avoids Z. By Miller's result mentioned above, there are infinitely many strings avoiding Y, whence $R(Z)$ is coinfinite. □

We now need a computability theoretic concept, the notion of a simple set. A coinfinite c.e. subset of $\{x, y\}^*$ (or of ω) is *simple* if its complement does not contain any infinite c.e. subsets. Simple sets are not computable (see [14]), and it follows from the definition that any coinfinite c.e. set containing a simple set is itself simple. The following lemma provides the set Z we use to define our semigroup.

Lemma 2.4. *There is a simple set Z such that $R(Z)$ is also simple, which implies that $\mathcal{A}(Z)$ is infinite, and that η_Z is not computable.*

Proof. Let $W_0, W_1, \dots$ be a standard enumeration of all c.e. subsets of $\{x, y\}^\star$. Let Z be the set of all u such that, for some i, the string u is the first string of length $\geqslant i + 5$ to be enumerated into W_i. Clearly, Z is c.e., and hence so is $R(Z)$.

It follows easily from the definition of Z that for each k, there are at most k many strings of length $k + 4$ in Z, so by Lemma 2.3, $R(Z)$ is coinfinite, and hence $\mathcal{A}(Z)$ is infinite.

If W_i is infinite then it contains a u such that $|u| \geqslant i+5$, so Z contains an element of W_i. Thus the complement of Z does not contain any infinite c.e. sets, so Z is simple. Since $R(Z)$ is coinfinite and contains Z, it is also simple.

To show that η_Z is not computable, assume otherwise, and let $z \in R(Z)$. Then $\{u \mid (u, z) \notin \eta_Z\}$ is an infinite computable subset of the complement of $R(Z)$, contradicting the simplicity of $R(Z)$. □

From now on, we fix a set Z as in the above lemma. In the next section, we present several properties of $\mathcal{A}(Z)$. In particular, we show that it has no finitely presented expansions.

3. Properties of the semigroup $\mathcal{A}(Z)$

3.1. On expansions of $\mathcal{A}(Z)$.

In this subsection, we prove the main result of this paper by showing that the semigroup $\mathcal{A}(Z)$ has no finitely presented expansions. To prove this fact, we need a few definitions and lemmas.

Definition 3.1. An algebra $\mathcal{A}$ is *residually finite* if for any two distinct elements a and b of $\mathcal{A}$, there is a homomorphism of $\mathcal{A}$ onto a finite algebra such that the images of a and b are distinct.

The following result was first proved by Mal'cev [11].

Lemma 3.2 (Mal'cev [11]). *If an algebra $\mathcal{A}$ is finitely presented and residually finite, then the word problem for $\mathcal{A}$ is decidable.*

Proof. Let S be as in Definition 1.5. By the comments following that definition, it is enough to show that $E(S)$ is computable. Since $E(S)$ is c.e., it suffices to show that $E(S)$ is also co-c.e. To enumerate the complement of $E(S)$, we enumerate all finite algebras with the same signature as $\mathcal{A}$ that are generated by their constants and satisfy S. For each such algebra and each pair of distinct elements b and b' of that algebra, there are ground terms t and t' such that b and b' are the interpretations of t and t', respectively. Let a and a' be the interpretations of t and t' in $\mathcal{A}$. Then we know that $(a, a') \notin E(S)$. The fact that $\mathcal{A}$ is residually finite ensures that every pair of elements of the domain of $\mathcal{A}$ that is not in E is eventually discovered in this fashion. □

A *translation* of an algebra $\mathcal{A}$ with domain A is a map $A \to A$ of the form $x \mapsto f(a_1, \ldots, a_{i-1}, x, a_i, \ldots, a_{n-1})$, where f is an n-ary basic operation of $\mathcal{A}$ and $a_1, \ldots, a_{n-1} \in A$.

Lemma 3.3. *Let E be an equivalence relation on the domain of an algebra $\mathcal{A}$. Then E is a congruence relation on $\mathcal{A}$ iff every translation of $\mathcal{A}$ respects E.*

Proof. It is clear that if E is a congruence relation, then every translation of $\mathcal{A}$ respects E. Now suppose that every translation of $\mathcal{A}$ respects E. Let f be an n-ary basic operation of $\mathcal{A}$ and let $(a_1, b_1), \ldots, (a_n, b_n) \in E$. Then $(f(a_1, a_2, \ldots, a_n), f(b_1, a_2, \ldots, a_n)) \in E$, since the map $x \mapsto f(x, a_2, \ldots a_n)$ is a translation of $\mathcal{A}$. By the same argument, we have $(f(b_1, a_2, a_3, \ldots, a_n), f(b_1, b_2, a_3, \ldots, a_n)) \in E$. Continuing this process and applying transitivity, we have $(f(a_1, a_2, \ldots, a_n), f(b_1, b_2, \ldots, b_n)) \in E$. □

The following lemma was proved by Kasymov in [8]. Recall that the equality relation of the semigroup $\mathcal{A}(Z)$ is denoted by η_Z, and defined by $\eta_Z = \{(p, q) \mid p = q \ \vee \ p, q \in R(Z)\}$.

Lemma 3.4 (Kasymov [8])**.** *Any c.e. algebra with equality relation η_Z is residually finite.*

Proof. Let $\mathcal{B} = (\{x, y\}^\star/\eta_Z; h_1, \ldots, h_k)$ be a c.e. algebra. Then $\mathcal{B} = \mathcal{C}/\eta_Z$ for some computable algebra $\mathcal{C} = (\{x, y\}^\star; H_1, \ldots, H_k)$. Note that the set T of all translations of $\mathcal{C}$ is uniformly computable.

Let $z \in R(Z)$, and let a and b be any pair of distinct elements of $\{x, y\}^* \setminus R(Z)$. We construct a congruence relation E on $\mathcal{B}$ so that E has finitely many equivalence classes, $([z], [a]) \notin E$, $([z], [b]) \notin E$, and $([a], [b]) \notin E$. (In this proof, $[u]$ means $[u]_{\eta_Z}$.) Then $\mathcal{B}/E$ is a finite homomorphic image of $\mathcal{B}$ in which the images of $[z]$, $[a]$, and $[b]$ are all distinct.

Enumerate a subset D of $\{x, y\}^* \setminus R(Z)$ as follows. Put a and b into D. Whenever we find a $g \in T$ and a $c \in \{x, y\}^*$ such that $g(c) \neq g(z)$ and either $g(c) \in D$ or $g(z) \in D$, put c into D. It is not hard to see that D is a c.e. subset of $\{x, y\}^* \setminus R(Z)$, and hence is finite.

Let $E = \{([u], [v]) \mid u = v \ \vee \ u, v \notin D\}$. Clearly E is an equivalence relation. Since D is finite, E has finitely many equivalence classes. Since $a, b \in D$, we have $([z], [a]) \notin E$, $([z], [b]) \notin E$, and $([a], [b]) \notin E$. To show that E is a congruence relation on $\mathcal{B}$, it is enough to show that $\widehat{E} = \{u, v \mid ([u], [v]) \in E\}$ is a congruence relation on $\mathcal{C}$. By Lemma 3.3, it is enough to show that every element of T respects $\widehat{E}$. Fix $g \in T$ and let $(u, v) \in \widehat{E}$. If $[u] = [v]$ then $[g(u)] = [g(v)]$, so $(g(u), g(v)) \in \widehat{E}$. Otherwise, $u, v \notin D$. If $g(z) \in D$ then $g(u) = g(v) = g(z)$; otherwise, $g(u), g(v) \notin D$. In either case, $([g(u)], [g(v)]) \in E$, so again $(g(u), g(v)) \in \widehat{E}$. □

We can now prove the main theorem of this paper.

Theorem 3.5. *There exists a computably enumerable, finitely generated, infinite semigroup with no finitely presented expansions.*

Proof. The semigroup $\mathcal{A}(Z)$ is c.e., finitely generated, and infinite. Let $\mathcal{B} = (\mathcal{A}(Z), h_1, \ldots, h_m)$ be an expansion of $\mathcal{A}(Z)$ that is a c.e. algebra. By Lemma 3.4, $\mathcal{B}$ is residually finite. If it were finitely presented then, by Lemma 3.2, the word problem for $\mathcal{B}$ would be decidable. That is, η_Z would be computable, contradicting Lemma 2.4. □

3.2. **Computability theoretic properties of $\mathcal{A}(Z)$.** Let $\mathcal{A}$ be a c.e. algebra of the form $\mathcal{B}/E$ (see Definition 1.1). Recall that elements of $\mathcal{A}$ are E-equivalence classes denoted by $[x]$, where x is an element of $\mathcal{B}$. Most infinite c.e. algebras are effectively infinite, in the sense that one can effectively list an infinite sequence of pairwise distinct elements of the algebra. For example, any infinite computable algebra is clearly effectively infinite. We formally define this property as follows.

Definition 3.6. The c.e. algebra $\mathcal{A} = \mathcal{B}/E$ is *effectively infinite* if there is a computable sequence $n_0, n_1, \ldots$ of elements of $\mathcal{B}$ such that $[n_i] \neq [n_j]$ for all $i \neq j$. Otherwise, we say that $\mathcal{A}$ is *algorithmically finite.*

Clearly, every finite algebra is algorithmically finite. The semigroup $\mathcal{A}(Z)$ constructed in Lemma 2.4 is an example of an infinite but algorithmically finite algebra.

Theorem 3.7. *The semigroup $\mathcal{A}(Z)$ is an infinite but algorithmically finite algebra.*

Proof. We have already seen that $\mathcal{A}(Z)$ is c.e. and infinite. If $n_0, n_1, \ldots$ are such that $(n_i, n_j) \notin \eta_Z$ for all $i \neq j$, then at most one n_i is in $R(Z)$, so $n_0, n_1, \ldots$ cannot be a computable sequence, as the complement of $R(Z)$ contains no infinite c.e. subsets. □

For $G \subseteq \{x, y\}^\star$, let G_n be the set of all strings of length n that are in G. We say that G is *generic* if

$$\lim_{n\to\infty} \frac{|G_n|}{2^n} = 1.$$

Elements of G are sometimes called *generic inputs.* We say that a problem P is *generically decidable* if there is an algorithm satisfying the following two conditions:

(1) Every output of the algorithm correctly solves P.
(2) There exists a generic set G such that the algorithm halts on all inputs in G.

Thus, a generically decidable problem always possesses an algorithm that solves it on a set of generic inputs while never giving a wrong answer. It turns out that many computationally hard problems, and even some undecidable problems, are generically decidable in polynomial time. Examples include the halting problem for Turing machines with a one-sided tape, the Post correspondence problem, the tree satisfiability problem, and the word problem for many finitely generated groups and semigroups [12]. In contrast, the word problem for the semigroup $\mathcal{A}(Z)$ is not generically decidable.

Theorem 3.8. *The word problem for the semigroup $\mathcal{A}(Z)$, that is, the relation η_Z, is not generically decidable.*

Proof. Suppose otherwise. Let G be a generic set, and f an algorithm such that $f(u, v)$ halts for all $u, v \in G$, and if $f(u, v)$ halts then $(u, v) \in \eta_Z$ iff $f(u, v) = 1$. Fix $z \in Z$. It is easy to see from the construction of Z that there are infinitely many n for which there exists a $w_n \in G_n$ with $w_n \notin R(Z)$. For each such n, we have $(z, w_n) \notin \eta_Z$, so there are infinitely many w such that $f(z, w)$ halts and is not equal to 1. Since f is computable, we obtain an infinite computable sequence of pairwise distinct elements of $\mathcal{A}(Z)$, contradicting Theorem 3.7. □

References

[1] J. A. Bergstra and J. V.Tucker, Initial and final algebra semantics for data type specifications: two characterization theorems, SIAM J. Comput. 12 (1983) 366–387.

[2] J. A. Bergstra and J. V. Tucker, Algebraic specifications of computable and semicomputable data types, Theoret. Comput. Sci. 50 (1987) 137–181.

[3] D. Cenzer, S. A. Dashti, and J. L. F. King, Computable symbolic dynamics, Math. Log. Q. 54 (2008) 460–469.

[4] Yu. L. Ershov and S. S. Goncharov, eds., Logic Notebook (Open Questions in Mathematical Logic), Akad. Nauk SSSR Sibirsk. Otdel., Inst. Mat., Novosibirsk, 1989 (in Russian).

[5] Yu. L. Ershov, S. S. Goncharov, A. Nerode, J. B. Remmel, and V. W. Marek, eds., Handbook of Recursive Mathematics, Vol. 1: Recursive Model Theory, Studies in Logic and the Foundations of Mathematics 138, North-Holland, Amsterdam, 1998.

[6] E. S. Golod and I. R. Shafarevich, On the class field tower, Izv. Akad. Nauk SSSR 28 (1964) 261–272 (in Russian).

[7] G. Grätzer, Universal Algebra, revised reprint of the second edition, Springer, New York, 2008.

[8] N. Kh. Kasymov, Algebras with finitely approximable positively representable enrichments, Algebra Logic 26 (1987) 441–450.

[9] B. Khoussainov. Randomness, computability, and algebraic specifications, Ann. Pure Appl. Logic 91 (1998) 1–15.

[10] B. Khoussainov and A. Miasnikov. On finitely presented expansions of algebras and groups, in preparation.

[11] A. I. Mal'cev, Constructive Algebras I, Uspehi Mat. Nauk 16 (1961) 3–60 (in Russian).

[12] I. Kapovich, A. Miasnikov, P. Schupp, and V. Shpilrain, Generic-case complexity, decision problems in group theory and random walks, J. Algebra 264 (2003) 665–694.

[13] J. S. Miller, Two notes on subshifts, to appear in Proc. Amer. Math. Soc.

[14] R. I. Soare, Recursively Enumerable Sets and Degrees, Springer-Verlag, Berlin, 1987.

Department of Mathematics, University of Chicago, 5734 S. University Ave., Chicago, IL 60637, U.S.A., drh@math.uchicago.edu

Department of Computer Science, University of Auckland, Private Bag 92019, Auckland, New Zealand, bmk@cs.auckland.ac.nz