Bulletin of the **S**ection of **L**ogic
Volume 30/3 (2001), pp. 155–164

Gábor Sági*

NON-COMPUTABILITY OF THE EQUATIONAL THEORY OF POLYADIC ALGEBRAS

Abstract

In [3] Daigneault and Monk proved that the class of (ω dimensional) representable polyadic algebras (RPA_ω for short) is axiomatizable by finitely many equation-schemas. However, this result does not imply that the equational theory of RPA_ω would be recursively enumerable; one simple reason is that the language of RPA_ω contains a continuum of operation symbols. Here we prove the following. Roughly, for any reasonable generalization of computability to uncountable languages, the equational theory of RPA_ω remains non-recursively enumerable, or non-computable, in the generalized sense. This result has some implications on the non-computational character of Keisler's completeness theorem for his "infinitary logic" in Keisler [6] as well.

1. Introduction

An abstract formulation of what we show here is the following. There are equational theories T for which there is a finite set of axiom schemas and there is a finite subset G of the primitives such that the following holds: If E_G is the set of those equations in which every function symbol belongs to G, then $T \cap E_G$ is not recursively enumerable, even though E_G is recursive. The theory T which will be used to illustrate this point is the theory of

*Supported by Hungarian National Foundation for Scientific Research grants T30314 and T035192.

all ω dimensional polyadic algebras (PA_ω for short). PA_ω is defined by a finite schema of equations.

PA_ω is an axiomatic approximation of the "concrete", set theoretically defined class RPA_ω of representable polyadic algebras. For completeness, we recall the definition of RPA_ω at the end of this section.

The connection between PA_ω and RPA_ω is analogous to that of Boolean algebras (BA's) and set Boolean algebras ($SetBA$'s). Clearly, $SetBA \subseteq BA$ and Stone's representation theorem says that every BA is isomorphic to a $SetBA$. In particular Stone's representation theorem implies a completeness theorem for the corresponding logic (classical propositional logic) and recursive enumerability for the equational theory $\mathbf{Eq}(SetBA)$. These kinds of consequences (completeness, enumerability) of representation theorems, in most cases, indeed follow from the representation theorem in question (see e.g. [1]). (The train of reasoning here usually is that representation theorems imply completeness theorems which in turn [usually] imply recursive enumerability.)

Here we will see that in the case of polyadic algebras this kind of "common sense reasoning" breaks down, because we will quote a representation theorem which does not imply recursive enumerability (for nontrivial reasons). Hence the quoted representation theorem might (or might not) imply a completeness theorem but the so obtained completeness theorem does not imply computability or recursive enumerability (even in any generalized sense). Further, we note that this observation has some implications on the "non-computational" character of Keisler's completeness theorem for his "infinitary logic" in Keisler [6]. Let us be more concrete.

Daigneault and Monk [3] proved the following representation theorem for polyadic algebras: every PA_ω is isomorphic to an RPA_ω. Since PA_ω is given by a finite schema of equational axioms, one might be templed to think that this representation result will imply some kind of recursive enumerability (in a generalized sense to avoid cardinality problem) for the equational theory of RPA_ω.

Here we will show that this is not the case. The result of Daigneault and Monk does not imply that the equational theory of RPA_ω would be recursively enumerable. Roughly, we will prove that for any reasonable generalization of computability to uncountable languages, $\mathbf{Eq}(RPA_\omega)$ remains non-recursively enumerable, or non-enumerable in the generalized sense as well. This will be seen to be an "intuitive" consequence of the following more formal statement. There is a strictly finite reduct L_G of the language of PA_ω, such that the equational consequences of PA_ω axioms written in L_G form a non-recursively enumerable set.

We note, that RPA_ω has a finite reduct having an undecidable equational theory: the substitutional free subreduct of RPA_ω is the class RDF_ω of representable diagonal free cylindric algebras which has an undecidable equational theory (cf. Theorems 5.1.66, 5.4.2 and 5.4.41 of [5]). The point of the present paper is recursive enumerability. Indeed, there is an essential difference between RDF_ω and RPA_ω: the equational theory of RDF_ω is recursively enumerable, but, as we will see, the equational theory of RPA_ω is not recursively enumerable (even in some generalized sense).

Below we sum up our notation, which is mostly standard. In the next section we prove the main theorem of the present paper.

Throughout, ω denotes the set of natural numbers. If A, B are sets then ${}^A B$ denotes the set of functions from A to B and $\mathcal{P}(A)$ denotes the power set of A, that is, $\mathcal{P}(A)$ consists of all subsets of A. In addition, id_A denotes the identity function on A.

If K is a class of algebras, then $\mathbf{S}K$ and $\mathbf{P}K$ denote the classes of subalgebras and direct products of members of K, respectively. $\mathbf{Eq}(K)$ denotes the equational theory of K.

Throughout we use function composition in such a way that the rightmost factor acts first. That is, for functions f, g we define $f \circ g(x) = f(g(x))$.

Definition 1.1. Let U be a set and α an ordinal. The *full polyadic equality set algebra* of dimension α with base U is the algebra

$$\langle \mathcal{P}({}^\alpha U); \cap, -, C_{(\Gamma)}, S_\tau, D_{i,j} \rangle_{\Gamma \subseteq \alpha,\ \tau \in {}^\alpha\alpha,\ i,j \in \alpha}$$

where $\cap$ and $-$ are intersection and complementation (w.r.t. the top element ${}^\alpha U$), and for any $X \subseteq {}^\alpha U$, $\Gamma \subseteq \alpha$, $\tau \in {}^\alpha\alpha$ and $i, j \in \alpha$

$$
\begin{array}{lcl}
C_{(\Gamma)}(X) & = & \{q \in {}^{\alpha}U : (\exists z \in X)(\forall j \notin \Gamma)(q_j = z_j)\}, \\
C_i(X) & = & C_{(\{i\})}(X), \\
S_\tau(X) & = & \{q \in {}^{\alpha}U : q \circ \tau \in X\}, \\
D_{i,j} & = & \{q \in {}^{\alpha}U : q_i = q_j\}.
\end{array}
$$

$SetPEA_\alpha := \mathbf{S}\{\mathcal{A} : \mathcal{A}$ is a full polyadic equality set algebra of dimension α with base U, for some set $U\}$. $SetPEA_\alpha$ is called the class of set polyadic equality algebras of dimension α.

$RPEA_\alpha := \mathbf{SP}SetPEA_\alpha$. $RPEA_\alpha$ is called the class of representable polyadic equality algebras of dimension α.

The class RPA_α of representable polyadic algebras of dimension α is defined to be the class of $D_{i,j}$ free subreducts of members of $RPEA_\alpha$.

Definition 1.2. Ax_{PA} denotes the set of usual defining equations of (ω dimensional) polyadic algebras.

For more detail and motivation we refer to e.g. Henkin-Monk-Tarski [5] or Németi [7]. The above mentioned definig equations can be found in [5], page 225.

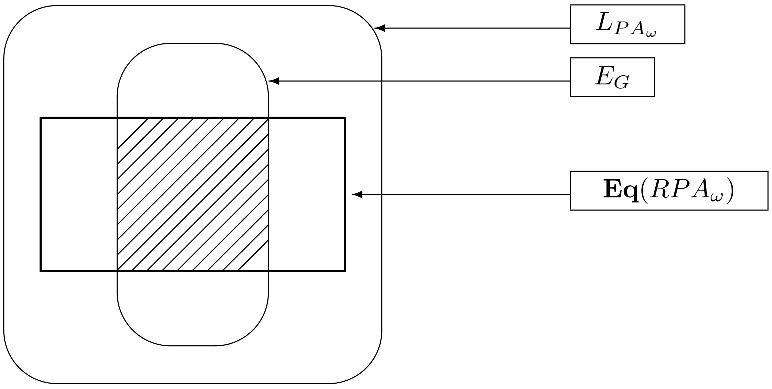
We will denote the polyadic operations by $\cap, -, C_{(\Gamma)}, S_\tau$ and $D_{i,j}$. The corresponding operation symbols (on the language level) are $\wedge, -, c_{(\Gamma)}, s_\tau$ and $d_{i,j}$, respectively.

2. The Construction

The construction presented below will be an example for a finite reduct L_G such that the following set of equations is not recursively enumerable:

$$\{e \in E_G : Ax_{PA} \vdash e\}$$

where E_G is the set of equations written in L_G. The intuitive consequence of this result is that the set of equational consequences of Ax_{PA} is not computable. This is not a precise mathematical statement, because the cardinality of this set is the continuum. But suppose one generalizes the concept of computability in a meaningful way to uncountable sets. Cf. the figure.



The only thing we assume about the new, generalized concept of computability is that the intersection of two computable set is computable, and countable sets are not computable if and only if they are non-computable in the old sense. Now $\{e : Ax_{PA} \vdash e\}$ is not computable in the new, generalized sense, because its intersection with a countable decidable (in the old sense) set E_G forms a non-recursively enumerable set, as we will prove.[1] Summing up, the set $\{e : Ax_{PA} \vdash e\}$ is really not computable.

CONSTRUCTION 2.1.

- *Let $f : \omega \times \omega \times \omega \to \omega$ be the following function.*

$$\forall i, j, k \in \omega, \quad f(i,j,k) = \begin{cases} l+1 & \text{if the } i\text{-th Register Machine program on the input } j \text{ terminates after at most } k \text{ steps, and the result is } l. \\ 0 & \text{otherwise.} \end{cases}$$

- *Let $\sigma_3 : \omega \times \omega \times \omega \to \omega$ be a function, which codes the triples of natural numbers, that is, the domain of σ_3 is $\omega \times \omega \times \omega$ and σ_3 is bijective, and σ_3, σ_3^{-1} are recursive. It is well known that such a σ_3 exists.*
- *Let $g = f \circ \sigma_3^{-1}$.*

[1] More precisely, a careful distinction should be made between (a) recursive, (b) recursively enumerable, (c) a generalized notion of recursive enumerability that also applies to higher infinity. We assume about (c) that the intersection of a set of kind (c) with a set of kind (a) to be of kind (b).

- *Let h be a recursive function which enumerates the σ_3 codes of those triples, in which the first coordinate is equal to zero, and the other two coordinates are arbitrary.*
- *Let $r : \omega \times \omega \times \omega \to \omega \times \omega \times \omega$ be defined by $r(i, j, k) = \langle i + 1, j, k \rangle$.*
- *Let $sc = \sigma_3 \circ r \circ {\sigma_3}^{-1}$. So sc first computes the triple encoded by its input, then increments the first component, then computes the σ_3 code of the new triple.*
- *Let $null$ be the constant zero function (on ω).*
- *Finally, let $G = \{g, h, sc, null\}$.*

Let L_G be the finite reduct of the language of Polyadic Algebras, which consists of the following basic operations: $s_g, s_h, s_{sc}, s_{null}$. Note that the elements of G are recursive functions.

Above we defined a reduct L_G of polyadic algebras. Since it contains substitutions only, L_G is a sublanguage of the language of the substitutional part of algebraic-first order logic as well. Observe in addition, that the indices of the substitutions occurring in L_G are recursive functions. Now we show that the set of equational consequences of Ax_{PA} written in L_G forms a non-recursively enumerable set. Throughout, i is a natural number and for any function f, the symbol $f^{(i)}$ stands for $\underbrace{f \circ \cdots \circ f}_{i}$.

Claim 2.2. *$g \circ sc^{(i)} \circ h = null$ iff the i-th Register Machine program computes the empty function (the domain of the empty function is the empty set).*

Proof. $g \circ sc^{(i)} \circ h = null$ iff
$(\forall n \in \omega)(g \circ sc^{(i)} \circ h\ (n) = 0)$ iff
$(\forall j, k \in \omega)(g \circ sc^{(i)} \circ \sigma_3\ (0, j, k) = 0)$ iff
$(\forall j, k \in \omega)(g \circ \sigma_3\ (i, j, k) = 0)$ iff
$(\forall j, k \in \omega)(f(i, j, k) = 0)$ iff

The i-th Register Machine program computes the empty function. ∎

Claim 2.3. *$Ax_{PA} \vdash s_g s_{sc}^{(i)} s_h(x) = s_{null}(x)$ iff the i-th Register Machine program computes the empty function.*

Proof. First suppose, the i-th Register Machine program computes the empty function. Then by the previous claim, $g \circ sc^{(i)} \circ h = null$ and then

$s_g s_{sc}^{(i)} s_h(x) = s_{g \circ sc^{(i)} \circ h}(x) = s_{null}(x)$. Here we used, that Ax_{PA} contains the axiom schema: $s_\tau \circ s_\sigma(x) = s_{\tau\circ\sigma}(x)$, where $\tau, \sigma \in {}^\omega\omega$ (see [5], page 225).

Second, suppose the i-th Register Machine program terminates, say, on the input j. Then by the previous claim, $g \circ sc^{(i)} \circ h$ and $null$ are not equal. Let

$$\mathcal{A} = \langle \mathcal{P}({}^\omega\omega); \cap, -, C_\Gamma, S_\tau \rangle_{\Gamma \subseteq \omega, \tau \in {}^\omega\omega},$$

(so $\mathcal{A} \in RPA_\omega$) and let $b = \{g \circ sc^{(i)} \circ h\} \in A$. Now one can easily check that $id_\omega \in S_g^{\mathcal{A}} S_{sc}^{(i)\mathcal{A}} S_h^{\mathcal{A}}(b)$, but id_ω is not in $S_{null}^{\mathcal{A}}(b)$, and hence $s_g s_{sc}^{(i)} s_h(x) = s_{null}(x)$ is not valid in $\mathcal{A}$, so (by soundness of equational logic) this equation is not derivable from Ax_{PA}.

Now we will recursively reduce the set of indices of the empty function to the set of equations derivable from Ax_{PA} and written in L_G.

THEOREM 2.4. *The consequences of Ax_{PA} (written in the language L_G) are not recursively enumerable, or equivalently,* $\mathbf{Eq}(RPA_\omega) \cap E_G$ *is not recursively enumerable.*

PROOF. Assume the opposite. Then, there exists an algorithm which enumerates the consequences of Ax_{PA} (written in the language L_G). But then there exists another algorithm which enumerates the following set I.

$$I = \{i \in \omega : Ax_{PA} \vdash s_g s_{sc}^{(i)} s_h(x) = s_{null}(x)\}.$$

By the previous claim $i \in I$ iff the i-th. Register Machine program computes the empty function. But it is well known that this set (and hence I) is not recursively enumerable. This contradiction completes the proof.

3. Related Results

Here we summarize some results about the complexity of the equational theory of some reducts of polyadic (equality) algebras.

DEFINITION 3.1. If Σ is a set of equations then $\mathbf{Ded}(\Sigma)$ denotes the set of all equational consequences of Σ.

Case of algebras without equality

By the theorem of Daigneault and Monk $\mathbf{Eq}(RPA_\omega) = \mathbf{Ded}(Ax_{PA})$, so the "complexity" of the equational theory of RPA_ω is the same as the "complexity" of the set of all equational consequences of Ax_{PA}.

Definition 3.2. Throughout $L_{suc,pred}$ denotes the following language. $L_{suc,pred}$ consists of the Boolean operations, and $c_0, s_{suc}, s_{pred}, s_{[0,1]}, s_{[i/j]}$, $i, j \in \omega$. Here $suc, pred$, $[k,l], [i/j]$ $(k, l \in \omega)$ are the following functions on ω:

$(\forall n \in \omega) \quad suc(n) = n+1,$

$$pred(n) = \begin{cases} \text{n-1} & \text{if } n \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

$$[k,l](n) = \begin{cases} \text{n} & \text{if } n \neq k, n \neq l, \\ \text{k} & \text{if } n = l, \\ \text{l} & \text{if } n = k. \end{cases}$$

$$[i/j](n) = \begin{cases} \text{n} & \text{if } n \neq i, \\ \text{j} & \text{if } n = i. \end{cases}$$

Note that in $L_{suc,pred}$ the following operations are term definable: $c_i, s_{[i,j]}$, $i, j \in \omega$. The following theorem is due to I. Sain and V. Gyuris, (see [11]) and is a possible solution of (some variant of) the finitization problem.

Theorem 3.3. *Let $L_{suc,pred,\omega} = L_{suc,pred} \cup \{c_\omega\}$. Then the reduct of PA_ω to $L_{suc,pred,\omega}$ generates a variety, which is axiomatizable by a recursive set $\Sigma_{suc,pred,\omega}$ of equations. This set is explicitly given by finitely many schemas.*

This theorem is a representation theorem, but the axiom system is recursive, so $\mathbf{Eq}(PA_\omega) \cap L_{suc,pred,\omega} = Ded(\Sigma_{suc,pred,\omega})$ is recursively enumerable.

Case of algebras with equality

It is well known that $RPEA_\omega$ is not closed under taking ultraproducts, and hence this class is not axiomatizable. Particularly, $\mathbf{Eq}(RPEA_\omega)$ does not agree with the set of all equational consequences of PEA_ω axiom schemas (see [5]). The equational theory of $RPEA_\omega$ is very complex. In [9] we proved, that this theory is not axiomatizable by schemas similar to the usual defining schemas of PEA_ω. The following theorem is proved in [11].

THEOREM 3.4. *Let $L_{suc,pred,D} = L_{suc,pred} \cup \{d_{0,1}\}$. Note that in this language all the $d_{i,j}$'s are term definable for $i, j \in \omega$. Then the reduct of $RPEA_\omega$ to $L_{suc,pred,D}$ generates a variety, which is axiomatizable by a recursive set $\Sigma_{suc,pred,D}$ of equations. This set is explicitly given by finitely many schemas.*

The following theorem is due to R. McKenzie (see Chapter 11 in Craig [2]).

THEOREM 3.5. *Let $L_{suc,pred,\omega,D} = L_{suc,pred} \cup \{c_\omega, d_{i,j},\ i, j \in \omega\}$. Then the set $\mathbf{Eq}(RPEA_\omega) \cap L_{suc,pred,\omega,D}$ is not recursively enumerable.*

The complexity of the equational theory of $RPEA_\omega$ is extremely high, as the following theorem claims.

THEOREM 3.6. *$\mathbf{Eq}(RPEA_\omega)$ is Π^1_1 hard [2], in the sense that there is a strictly finite reduct L_* of the language of $RPEA_\omega$ and a recursive function Tr such that Tr maps the Π^1_1 formulas of arithmetic to equations written in L_*, and for any Π^1_1 sentence σ the arithmetical validity of σ is equivalent with $RPEA_\omega \models Tr(\sigma)$. (For the second order logical and recursion theoretic notions used here we refer to [10].)*

For the proof see [9]. The intuitive consequence of this result is non-computability of $\mathbf{Eq}(RPEA_\omega)$ similarly to the argument in the beginning of Section 2.

ACKNOWLEDGEMENTS. Thanks are due to William Craig and István Németi for their remarks and help.

[2] Π^1_1 hardness is a very strong form of non-computability, e.g. it implies non-enumerability by any kind of algorithm, but Π^1_1 hardness is a much stronger negative property than this cf. Odifreddi [10] for the definition of Π^1_1 sets.

References

[1] H. Andréka, Á. Kurucz, I. Németi, I. Sain, *Applying Algebraic Logic; a General Methodology*, Preprint, Math. Inst., Budapest, (1994), 72 pp. Shortened version of this appeared as *Applying algebraic logic to logic*, [in:] **"Algebraic Methodology and Software Technology"** (eds. M. Nivat et al) Springer-Verlag, (1994) pp. 5–26.

[2] W. Craig, *Logic in algebraic form*, North-Holland, Amsterdam (1974).

[3] J.D. Monk, A. Daigneault, *Representation theory for polyadic algebras*, **Fund. Math.**, Vol 52 (1963), pp. 151–176.

[4] P. Halmos, *Algebraic Logic*, Chelsea Publ. Company, New York (1962).

[5] L. Henkin, J.D. Monk, A. Tarski, *Cylindric Algebras Part 2*, North-Holland, Amsterdam (1985).

[6] H.J. Keisler, *A complete first-order logic with infinitary predicates*, **Fund. Math.**, vol. 52 (1963), pp. 177–203.

[7] I. Németi, *Algebraization of Quantifier Logics, an introductory overview 12-th version.* Math. Inst. Budapest, Preprint, No. 13/1996. Available on the following electronic address: http://renyi.hu/pub/algebraic-logic/survey.dvi. An extended abstract of this appeared in **Studia Logica** vol. 50, No 3/4. pp. 485–569.

[8] I. Németi, G. Sági, *On the Equational Theory of Representable Polyadic Equality Algebras (Extended Abstract)*, **Journal of the IGPL**, vol. 1 (1998), pp. 2–16.

[9] I. Németi, G. Sági, *On the Equational Theory of Representable Polyadic Algebras*, **Journal of Symbolic Logic**, Vol. 65, No. 3 (2000), pp. 1143–1167.

[10] P. Odifreddi, *Classical Recursion Theory*, North-Holland, Amsterdam (1989).

[11] I. Sain, V. Gyuris, *Finite Schematizable Algebraic Logic*, **Bulletin of IGPL**, vol. 5, No. 5 (1997), pp. 699–751.

Alfréd Rényi Institute of Mathematics
Hungarian Academy of Sciences
Budapest Pf. 127
H-1364 Hungary
e-mail: sagi@renyi.hu