COLLOQUIUM MATHEMATICUM

VOL. LXVII 1994 FASC. 1

SOME DECIDABLE THEORIES WITH FINITELY MANY COVERS WHICH ARE DECIDABLE AND ALGORITHMICALLY FOUND

BY

CORNELIA KALFA (THESSALONIKI)

In any recursive algebraic language, I find an interval of the lattice of equational theories, every element of which has finitely many covers. With every finite set of equations of this language, an equational theory of this interval is associated, which is decidable with decidable covers that can be algorithmically found. If the language is finite, both this theory and its covers are finitely based. Also, for every finite language and for every natural number n, I construct a finitely based decidable theory together with its exactly n covers which are decidable and finitely based. The construction is algorithmic.

1. Introduction and preliminaries. Let L be an *algebraic* language, i.e. a first-order language with equality, with at least one operation symbol and no relation symbol. An L-*equation* is a universal sentence of the form $(\forall\overline{v})(\varphi = \psi)$, where φ and ψ are L-terms. An (*equational*) *theory* is a set of L-equations closed under its logical consequences which are equations. Denote by Th_L and Th_Σ, respectively, the set of theories of L and the set of theories of L which imply Σ. Then $\langle \mathrm{Th}_L, \subset\rangle$ and $\langle \mathrm{Th}_\Sigma, \subset\rangle$ are lattices. If Φ and Ψ are L-theories, we say that *Φ is a cover of Ψ* if Φ is an immediate successor of Ψ in the lattice $\langle \mathrm{Th}_L, \subset\rangle$. The first element of $\langle \mathrm{Th}_\Sigma, \subset\rangle$, which always exists, is called the *theory generated by* Σ and it is denoted by $\Theta[\Sigma]$. Σ is then called a *basis* of $\Theta[\Sigma]$. If a theory Φ has a finite basis, it is called *finitely based*.

Consider the language

$$L' = \langle f, s\rangle,$$

with one unary operation symbol f and at most one constant symbol s.

In Ježek [2] it is proved that *every L-theory has a unique basis which is a* r e d u c e d s e t, *i.e. a set of one of the following kinds*:

(1) $\Sigma' = \emptyset$,

1991 *Mathematics Subject Classification*: Primary 03B25.

(2) $\Sigma' = \{f^l s = f^{l+e} s\}$, for $e > 0$,

(3) $\Sigma' = \{(\forall\overline{v})(f^k v_1 = f^{k+d} v_1),\ f^l s = f^{l+e} s\}$, for $d > 0$, $e > 0$, $l \le k$ and e a divisor of d,

(4) $\Sigma' = \{(\forall\overline{v})(f^k v_1 = f^k v_2), f^l s = f^{l+1} s\}$, for $l \le k$.

In [3] I find

(i) *an algorithm which, upon the input of any finite set Σ' of L'-equations, outputs the reduced set which is a basis of $\Theta[\Sigma']$,*

(ii) *an algorithm which, upon the input of any reduced set R', outputs the finite set of all reduced sets that are bases of covers of $\Theta[\Sigma']$.*

In Ehrenfeucht [1] it is proved that the first-order theory generated by the logical axiom of L'—as well as of all other trivial languages—is decidable. From this it follows that *every equational theory of L', a finite basis of which we know, is decidable.* (Given any equation ε'_0, check by the existing procedure whether the sentence $\bigwedge_{\varepsilon'\in\Sigma'} \varepsilon' \to \varepsilon'_0$ is an L'-tautology or not.)

In this paper, I consider any *recursive* algebraic language L, i.e. any algebraic language with recursive sets of operation symbols and constant symbols. I show that the lattice $\langle \mathrm{Th}_L, \subset\rangle$ contains sublattice a $\langle \mathrm{Th}_\Lambda, \subset\rangle$ isomorphic to the lattice $\langle \mathrm{Th}_L, \subset\rangle$ and, making use of the above mentioned facts about L', I get the following results about L:

(1) *All theories of the interval $\langle \mathrm{Th}_\Lambda, \subset\rangle$ have finitely many covers and, provided that we know a finite basis of them, they are decidable. Moreover, if L is finite, then all L-theories of the above interval are finitely based.*

(2) *There is an algorithm which, upon the input of any finite set Σ of L-equations, outputs the set of all covers of $\Theta[\Sigma \cup \Lambda]$.*

(3) *In all finite L's, there is an algorithm which, upon the input of any natural number n, outputs a finitely based decidable L-theory of the interval $\langle \mathrm{Th}_\Lambda, \subset\rangle$ together with its exactly n covers which are also finitely based and decidable.*

The problem of whether, for a given language L and a given cardinal k, there exists an L-theory with exactly k covers is not new, and in [4] a summary of the existing results can be found. Since with every cover of a theory an equation can be associated, for finite L's, the problem is restricted only to countable k's. In the same line with my third result, McNulty found in [4], in all countable languages L and for all natural numbers n, finitely based L-theories which cover exactly n others.

Before proving my claims, I make some more notational conventions: Va, Ter_L and Eq_L stand for the sets of variables, terms and equations of L, respectively. An equation $(\forall\overline{v})(\varphi = \psi)$ is denoted, from now on, simply by $\varphi = \psi$. Gothic letters $\mathfrak{A}, \mathfrak{B}, \ldots$ are used for L-algebras, $|\mathfrak{A}|, |\mathfrak{B}|, \ldots$ for their universes and $\sigma^{\mathfrak{A}}$ for the interpretation of the L-expression σ in $\mathfrak{A}$. Finally, Mod_L and Mod_Σ are used to denote the class of all L-algebras and the class of all L-algebras satisfying $\Sigma \subset \mathrm{Eq}_L$, respectively.

2. Embedding. Let L be any *recursive* algebraic language with at least one operation symbol Q of rank $r(Q)$. If L has only operation symbols, i.e. if it is of the form

$$L = \langle \{Q\} \cup \{Q_i\}_{i \in I} \rangle,$$

with Q_i an operation symbol of rank $r(i)$, take

$$L' = \langle f \rangle.$$

If L has at least one constant symbol c, i.e. if it is of the form

$$L = \langle \{Q\} \cup \{Q_i\}_{i \in I},\ \{c\} \cup \{c_j\}_{j \in J} \rangle,$$

take

$$L' = \langle f, s \rangle.$$

Consider the recursive functions $h' : \mathrm{Ter}_{L'} \to \mathrm{Ter}_L$ and $h : \mathrm{Ter}_L \to \mathrm{Ter}_{L'}$, given by the rules

$$\begin{aligned} &h'(v_i) = v_i, \\ &h'(s) = c, \\ &h'(f\vartheta') = Qh'(\vartheta')h'(\vartheta') \ldots h'(\vartheta') \end{aligned}$$

and

$$\begin{aligned} &h(v_i) = v_i, \\ &h(c) = h(c_j) = s, \\ &h(Q\vartheta_1\vartheta_2 \ldots \vartheta_{r(Q)}) = fh(\vartheta_1), \\ &h(Q_i\vartheta_1\vartheta_2 \ldots \vartheta_{r(i)}) = h(\vartheta_1), \end{aligned}$$

for the terms existing in the particular language. Denote also by h' the induced recursive functions on $\mathrm{Eq}_{L'}$, and on its power set $P(\mathrm{Eq}_{L'})$, and by h the induced recursive functions on Eq_L and $P(\mathrm{Eq}_L)$. *Obviously, h' and h act as interpretors from the one language to the other*.

Consider also the class-functions $H' : \mathrm{Mod}_{L'} \to \mathrm{Mod}_L$ and $H : \mathrm{Mod}_L \to \mathrm{Mod}_{L'}$ given, respectively, by the rules

$$\begin{aligned} &|H'(\mathfrak{A}')| = |\mathfrak{A}'|, \\ &c^{H'(\mathfrak{A}')} = c_j^{H'(\mathfrak{A}')} = s^{\mathfrak{A}'}, \\ &Q^{H'(\mathfrak{A}')}(\overline{a}) = f^{\mathfrak{A}'}(a_1),\ \text{for } \overline{a} = \langle a_1, \ldots, a_{r(Q)} \rangle, \\ &Q_i^{H'(\mathfrak{A}')}(\overline{a}) = a_1,\ \text{for } \overline{a} = \langle a_1, \ldots, a_{r(i)} \rangle, \end{aligned}$$

and

$$|H(\mathfrak{A})| = |\mathfrak{A}|,$$
$$s^{H(\mathfrak{A})} = c^{\mathfrak{A}},$$
$$f^{H(\mathfrak{A})}(a) = Q^{\mathfrak{A}}(a, a, \ldots, a).$$

It can be easily shown that

$$(\forall \mathfrak{A}' \in \mathrm{Mod}_{L'})(H(H'(\mathfrak{A}')) = \mathfrak{A}'),$$

from which it follows that H' *is injective and* H *is surjective*.

Finally, consider the set of L-equations

$$\Lambda = \{Qv_1v_2 \ldots v_{r(Q)} = Qv_1v_1 \ldots v_1\} \\ \cup \{Q_iv_1 \ldots v_{r(i)} = v_1 : i \in I\} \cup \{c_j = c : j \in J\}.$$

Informally speaking, Λ eliminates all symbols but Q and c, and makes Q act as unary operation symbol. I will prove that

THEOREM 1. *The function* $g : \mathrm{Th}_{L'} \to \mathrm{Th}_\Lambda$ *given by the rule*

$$\forall \Phi' \in \mathrm{Th}_{L'}, \quad g(\Phi') = \Theta[h'(\Phi) \cup \Lambda] \in \mathrm{Th}_L,$$

is a lattice isomorphism of $\langle \mathrm{Th}_{L'}, \subset \rangle$ *onto* $\langle \mathrm{Th}_\Lambda, \subset \rangle$.

For the proof of Theorem 1, I need three lemmas:

LEMMA 2. (a) $(\forall t' \in \mathrm{Ter}_{L'})(h(h'(t')) = t')$.
(b) $(\forall t \in \mathrm{Ter}_L)(\Lambda \vDash h'(h(t)) = t)$.

Proof. By two easy inductions on the complexity of the term. ∎

LEMMA 3. (a) $(\forall \mathfrak{A}' \in \mathrm{Mod}_{L'})(\forall \Sigma' \subset \mathrm{Eq}_{L'})(\mathfrak{A}' \vDash \Sigma' \Leftrightarrow H'(\mathfrak{A}') \vDash h'(\Sigma'))$.
(b) $(\forall \mathfrak{A}' \in \mathrm{Mod}_{L'})(\forall \Sigma \subset \mathrm{Eq}_L)(\mathfrak{A}' \vDash h(\Sigma) \Leftrightarrow H'(\mathfrak{A}') \vDash \Sigma)$.
(c) $(\forall \mathfrak{A} \in \mathrm{Mod}_L)(\forall \Sigma' \subset \mathrm{Eq}_{L'})(\mathfrak{A} \vDash h'(\Sigma') \Leftrightarrow H(\mathfrak{A}) \vDash \Sigma')$.
(d) $(\forall \mathfrak{A} \in \mathrm{Mod}_\Lambda)(\forall \Sigma \subset \mathrm{Eq}_L)(\mathfrak{A} \vDash \Sigma \Leftrightarrow H(\mathfrak{A}) \vDash h(\Sigma))$.

Proof. One can prove by induction that

$$(1) \qquad (\forall \mathfrak{A}' \in \mathrm{Mod}_{L'})(\forall t' \in \mathrm{Ter}_{L'})(t'^{\mathfrak{A}'} = h'(t')^{H'(\mathfrak{A}')}).$$

From (1), (a) follows. Since all $H'(\mathfrak{A}')$'s are models of Λ, (1) implies that

$$(\forall \mathfrak{A}' \in \mathrm{Mod}_L)(\forall t \in \mathrm{Ter}_L)(h(t)^{\mathfrak{A}'} = h'(h(t))^{H'(\mathfrak{A}')} = t^{H'(\mathfrak{A}')}),$$

from which (b) immediately follows.

One can also prove by induction that

$$(\forall \mathfrak{A} \in \mathrm{Mod}_L)(\forall t' \in \mathrm{Ter}_{L'})(t'^{H(\mathfrak{A})} = h'(t')^{\mathfrak{A}})$$

and use it to prove (c) and (d). ∎

LEMMA 4. (a) $(\forall \Sigma' \subset \mathrm{Eq}_{L'})(\forall \varepsilon' \in \mathrm{Eq}_{L'})(\Sigma' \vDash \varepsilon' \Leftrightarrow h'(\Sigma') \vDash h'(\varepsilon'))$.
(b) $(\forall \Sigma \subset \mathrm{Eq}_L)(\forall \varepsilon \in \mathrm{Eq}_L)(\Sigma \cup \Lambda \vDash \varepsilon \Leftrightarrow h(\Sigma) \vDash h(\varepsilon))$.

Proof. I leave the easiest part to the reader and prove (b):

Suppose that $\Sigma \cup \Lambda \vDash \varepsilon$. Then from the fact that all $H'(\mathfrak{A}')$'s are models of Λ and from Lemma 3(b) it follows that $\mathfrak{A}' \vDash h(\Sigma) \Rightarrow H'(\mathfrak{A}') \vDash \Sigma \Rightarrow H'(\mathfrak{A}') \vDash \Sigma \cup \Lambda \Rightarrow H'(\mathfrak{A}') \vDash \varepsilon \Rightarrow \mathfrak{A}' \vDash h(\varepsilon)$. Consequently, $h(\Sigma) \vDash h(\varepsilon)$.

Suppose that $h(\Sigma) \vDash h(\varepsilon)$. Then from Lemma 3(d) it follows that $\mathfrak{A} \vDash \Sigma \cup \Lambda \Rightarrow H(\mathfrak{A}) \vDash h(\Sigma) \Rightarrow H(\mathfrak{A}) \vDash h(\varepsilon) \Rightarrow \mathfrak{A} \vDash \varepsilon$. Consequently, $\Sigma \cup \Lambda \vDash \varepsilon$.

Proof of Theorem 1. *g is injective*: Suppose that $g(\Phi'_1) = g(\Phi'_2)$. Then, by Lemmas 2(a) and 4(b), we get $\varepsilon' \in \Phi'_1 \Rightarrow h'(\varepsilon') \in h'(\Phi'_1) \cup \Lambda \Rightarrow h'(\Phi'_1) \cup \Lambda \vDash h'(\varepsilon') \Rightarrow h'(\Phi'_2) \cup \Lambda \vDash h'(\varepsilon') \Rightarrow h(h'(\Phi'_2)) \vDash h(h'(\varepsilon')) \Rightarrow \Phi'_2 \vDash \varepsilon' \Rightarrow \varepsilon' \in \Phi'_2$. The converse is proved similarly.

g is surjective: Consider any $\Phi \in \mathrm{Th}_\Lambda$. From Lemmas 2(a) and 4(b) it follows that $h(\Phi) \vDash \varepsilon' \Rightarrow h(\Phi) \vDash h(h'(\varepsilon')) \Rightarrow \Phi \vDash h'(\varepsilon') \Rightarrow h'(\varepsilon') \in \Phi \Rightarrow h(h'(\varepsilon')) \in h(\Phi) \Rightarrow \varepsilon' \in h(\Phi)$. So $h(\Phi) \in \mathrm{Th}_{L'}$.

Also, $\varepsilon \in g(h(\Phi)) \Leftrightarrow h'(h(\Phi)) \cup \Lambda \vDash \varepsilon \Leftrightarrow h(h'(h(\Phi))) \vDash h(\varepsilon) \Leftrightarrow h(\Phi) \vDash h(\varepsilon) \Leftrightarrow \Phi \cup \Lambda \vDash \varepsilon \Leftrightarrow \varepsilon \in \Phi$.

I have proved that $(\forall \Phi \in \mathrm{Th}_\Lambda)(\exists h(\Phi) \in \mathrm{Th}_{L'})(g(h(\Phi)) = \Phi)$.

The rest of the proof is left to the reader. ∎

3. Corollaries

COROLLARY 5. *For every $\Phi' \in \mathrm{Th}_{L'}$, the set of covers of $g(\Phi')$ is the set $\{g(\Psi') : \Psi'$ is a cover of $\Phi'\}$.*

Proof. Since g is a lattice isomorphism, it follows that Φ'_1 covers Φ'_2 iff $g(\Phi'_1)$ covers $g(\Phi'_2)$. From this and the fact that g is surjective, the assertion follows. ∎

COROLLARY 6. *Every L-theory in Th_Λ has finitely many covers.*

Proof. In [3] it is proved that all L'-theories have finitely many covers. So, the set of covers $\{g(\Psi') : \Psi'$ is a cover of $g^{-1}(\Phi)\}$ of $\Phi \in \mathrm{Th}_\Lambda$ is finite. ∎

COROLLARY 7. *There is an algorithm which, upon the input of any finite set Σ of L-equations, outputs the finite set of all covers of $\Theta[\Sigma \cup \Lambda]$.*

Proof. Firstly, I show that

$$(2) \qquad (\forall \Phi' \in \mathrm{Th}_{L'})(\forall \Sigma' \subset \mathrm{Eq}_{L'})(\Theta[\Sigma'] = \Phi' \Leftrightarrow \Theta[h'(\Sigma') \cup \Lambda] = g(\Phi)).$$

Indeed, $\varepsilon \in g(\Phi) \Rightarrow h(\Theta[\Sigma']) \cup \Lambda \vDash \varepsilon \Rightarrow h(h'(\Theta[\Sigma'])) \vDash h(\varepsilon) \Rightarrow \Sigma' = h(h'(\Sigma')) \vDash h(\varepsilon) \Rightarrow h'(\Sigma') \cup \Lambda \vDash \varepsilon$. This proves (2).

Now, I write the required algorithm: Given any finite $\Sigma \subset \mathrm{Eq}_L$,

(a) Find $h(\Sigma)$.

(b) Find, by the algorithm constructed in [3], the finite set B of all reduced sets which are bases of covers of $\Theta[h(\Sigma)]$.

(c) Find the finite set $\{h'(R') : R' \in B\}$.

By (2) and the fact that g is injective, $\{h'(R') \cup \Lambda : R' \in B\}$ is a set of representatives of the bases of all covers of $\Theta[\Sigma]$. ∎

COROLLARY 8. *For every finite* $\Sigma \subset \mathrm{Eq}_L$, *the theory* $\Theta[\Sigma \cup \Lambda]$ *is decidable. So, every theory in* Th_Λ, *a finite basis of which we know, is decidable.*

Proof. Given any $\varepsilon \in \mathrm{Eq}_L$,

(a) Find $h(\Sigma)$ and $h(\varepsilon)$.

(b) Check whether $h(\Sigma) \vDash h(\varepsilon)$ (since $\Theta[h(\Sigma)]$ is decidable, this can be done). *If yes*, then $\Sigma \cup \Lambda \vDash \varepsilon$. *If no*, then $\Sigma \cup \Lambda \nvDash \varepsilon$.

So $\Theta[\Sigma \cup \Lambda]$ is decidable. ∎

COROLLARY 9. *Every L-theory in* Th_Λ *of a finite L is finitely based.*

Proof. In [2] it is proved that all L'-theories are finitely based. If Φ is an L-theory and Σ' is a finite basis of $g^{-1}(\Phi)$, then, by (2) in the proof of Corollary 7, $h'(\Sigma) \cup \Lambda$ is a finite basis of Φ. ∎

COROLLARY 10. *If L is finite, there is an algorithm which, upon the input of any natural number n, outputs a finitely based, decidable L-theory together with its exactly n covers which are also finitely based and decidable.*

Proof. From Theorems 2.0 and 2.1 of [3] it follows that the theory of $\langle f \rangle$ based on the equation

$$\varepsilon' : v_1 = f^{p_1 p_2 \ldots p_n} v_1,$$

where p_i is the ith prime number, has as covers exactly the n theories based on the equations

$$\varepsilon'_i : v_1 = f^{p_1 \ldots p_{i-1} p_{i+1} \ldots p_n} v_1,$$

for $i \in \{1, \ldots, n\}$. It also follows that the theory of $\langle f, s \rangle$, based on the equation

$$\varepsilon' : s = f^{p_1 p_2 \ldots p_n} s,$$

has as covers exactly the n theories based on the equations

$$\varepsilon'_i : s = f^{p_1 \ldots p_{i-1} p_{i+1} \ldots p_n} s.$$

So, by Corollary 5 and (2), *the L-theories based on the finite sets* $\{h'(\varepsilon')\} \cup \Lambda$ *have as covers exactly the n theories based on the finite sets* $\{h'(\varepsilon'_i)\} \cup \Lambda$. By Corollary 8, all these theories are decidable.

The above construction is, obviously, algorithmic: Given $n \in \mathbb{N}$, one can write down the $(n+1)$-tuple of L'-equations $\langle \varepsilon', \varepsilon'_1, \varepsilon'_2, \ldots, \varepsilon'_n \rangle$, then the $(n+1)$-tuple of L-equations $\langle h'(\varepsilon'), h'(\varepsilon'_1), \ldots, h'(\varepsilon'_n) \rangle$ and finally the $(n+1)$-tuple of finite sets of L-equations $\langle \{h'(\varepsilon')\} \cup \Lambda, \{h'(\varepsilon'_1)\} \cup \Lambda, \ldots$ $\ldots, \{h'(\varepsilon'_n)\} \cup \Lambda \rangle$. ∎

REFERENCES

[1] A. Ehrenfeucht, *Decidability at the theory of one function*, Notices Amer. Math. Soc. 6 (1959), 268.

[2] J. Ježek, *Primitive classes of algebras with unary and nullary operations*, Colloq. Math. 20 (1969), 159–179.

[3] C. Kalfa, *Covering relation in the language of mono-unary algebras with at most one constant symbol*, Algebra Universalis 26 (1989), 143–148.

[4] G. F. McNulty, *Covering in the lattice of equational theories and some properties of term finite theories*, ibid. 15 (1982), 115–125.

DEPARTMENT OF MATHEMATICS
FACULTY OF SCIENCE
ARISTOTLE UNIVERSITY
THESSALONIKI, GREECE

Reçu par la Rédaction le 5.5.1993;
en version modifiée le 2.9.1993