Math. Struct. in Comp. Science (2006), *vol.* 16, *pp.* 763–788.
doi:10.1017/S0960129506005585 Printed in the United Kingdom

On the complexity of inductive definitions

DOUGLAS CENZER[†] and JEFFREY B. REMMEL[‡]

[†]*Department of Mathematics, University of Florida, P.O. Box 118105, Gainesville, Florida 32611, U.S.A.*
[‡]*Department of Mathematics, University of California at San Diego, La Jolla, CA 92093, U.S.A.*

Received 25 September 2005; revised 15 December 2005

We study the complexity of computable and Σ^0_1 inductive definitions of sets of natural numbers. For example, we show how to assign natural indices to monotone Σ^0_1-definitions and then use these to calculate the complexity of the set of all indices of monotone Σ^0_1-definitions that are computable. We also examine the complexity of a new type of inductive definition, which we call *weakly finitary* monotone inductive definitions. Applications are given in proof theory and in logic programming.

1. Introduction

Inductive definitions play a central role in mathematical logic and computer science. For example, the set of formulas in a first-order language is given by an inductive definition. Given a set A of axioms for a mathematical theory T and a set of logical axioms and rules, the theory T is obtained by an inductive definition. The set of computable functions can be realised by an inductive definition. Similarly, for any Horn logic program P, the unique stable model of P is obtained by an inductive definition.

It is well known that for any computable or Σ^0_1 monotone inductive definition Γ, one can construct the closure of Γ, $\mathrm{Cl}(\Gamma)$, in at most ω steps and $\mathrm{Cl}(\Gamma)$ is always a Σ^0_1 set. In some situations it is important that $\mathrm{Cl}(\Gamma)$ is computable. For example, it is important that the set of formulas in a typical first-order theory is computable. In other situations we know that $\mathrm{Cl}(\Gamma)$ is Σ^0_1 but not computable. For example, even a finitely axiomatisable theory T may be Σ^0_1 but not decidable (computable). In this paper, we explore the complexity of various properties of the closure of a Σ^0_1 monotone inductive definition Γ. As examples, we consider properties such as when the closure of Γ is finite, cofinite or computable, or when the closure ordinal of Γ is finite or equal to ω. We do this by assigning indices to Σ^0_1 monotone inductive operators. In particular, this means that we can effectively enumerate the family of all Σ^0_1 monotone inductive operators as $\Gamma_0, \Gamma_1, \ldots$. Then, for example, we show that the set C of indices e such that the closure or *least fixed point* $\mathrm{lfp}(\Gamma_e)$ is computable is Σ^0_3 complete.

We also define a new class of inductive operators called *weakly finitary* monotone inductive operators. The basic idea is that for a weakly finitary operator there may exist a finite set of elements x such that x is forced into $\Gamma(A)$ only if A contains one of a collection of possibly infinite sets. We show that if Γ is a weakly finitary monotone

[‡] This work partially supported by NSF under grant DMS 0400507.

inductive operator, it is still the case that lfp(Γ) will be Σ^0_1 but that it can take more than ω steps to construct lfp(Γ). An example of such an operator is when we allow finitely many instances of the ω-rule to generate a partial theory of arithmetic. We also assign indices to the family of weakly finite Σ^0_1 monotone inductive operators. We show that the set of indices of weakly finitary Σ^0_1 monotone inductive operators Γ such that lfp(Γ) is computable is also Σ^0_3 complete. However, for certain computable sets R, the set of indices of weakly finitary Σ^0_1 monotone inductive operators Γ such that lfp(Γ) ∩ R is computable lies in the difference hierarchy over the Σ^0_3 sets.

We use standard notation from computability theory (Soare 1987). Let $\mathbb{N}$ denote the set of natural numbers and $\mathscr{P}(\mathbb{N})$ denote the set of all subsets of $\mathbb{N}$. In particular, we let ϕ_e (ϕ^A_e) denote the e-th partial computable function (e-th A-partial computable function) from $\mathbb{N}$ to $\mathbb{N}$ and let $W_e = \mathrm{Dom}(\phi_e)$ ($W^A_e = \mathrm{Dom}(\phi^A_e)$) be the e-th computably enumerable (c.e.) (e-th A-computably enumerable) subset of $\mathbb{N}$. Note that computably enumerable and *recursively enumerable (r.e.)* have the same meaning, and, similarly, computable functions are also known as recursive functions. We let $W_{e,s}$ ($W^A_{e,s}$) denote the set of numbers $m \leqslant s$ such that $\phi_e(m)$ ($\phi^A_e(m)$) converges in s or fewer steps. Given a finite set $S = \{a_1 < \ldots a_n\}$, the canonical index of S is $\sum_{i=1}^n 2^{a_i}$. The canonical index of the empty set is 0. We let D_n denote the finite set whose canonical index is n.

We fix a primitive recursive pairing function, $[x, y] = \frac{1}{2}((x + y)^2 + 3x + y)$ from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$. For any sequence $a_1, \ldots, a_n$ with $n \geqslant 3$, we define $[a_1, \ldots, a_n]$ by the usual inductive procedure of defining $[a_1, \ldots, a_n] = [a_1, [a_2, \ldots, a_n]$. The *explicit index* of the sequence $(a_1, \ldots, a_n)$ is defined by $\langle a_1 \rangle = [1, a_1]$ if $n = 1$ and $\langle a_1, \ldots, a_n \rangle = [n, [a_1, \ldots, a_n]]$ if $n \geqslant 2$.

2. Inductive definitions

In this paper, we are going to consider inductive operators $\Gamma : \mathscr{P}(\mathbb{N}) \to \mathscr{P}(\mathbb{N})$ that inductively define subsets of $\mathbb{N}$. We begin with a review of basic definitions and results, which can be found, for example, in Hinman (1978).

Definition 2.1. Let $\Gamma : \mathscr{P}(\mathbb{N}) \to \mathscr{P}(\mathbb{N})$.

1 Γ is said to be *monotone* if $A \subset B$ implies $\Gamma(A) \subseteq \Gamma(B)$ for all A, B.
2 Γ is said to be *inclusive* if $A \subseteq \Gamma(A)$ for all A.
3 Γ is said to be *inductive* if it is either monotone or inclusive.

An inductive operator Γ recursively defines a sequence $\{\Gamma^\alpha : \alpha \text{ an ordinal}\}$ by setting $\Gamma^0 = \emptyset$, $\Gamma^{\alpha+1} = \Gamma(\Gamma^\alpha)$ for all α and $\Gamma^\lambda = \bigcup_{\alpha<\lambda} \Gamma^\alpha$. It is easy to see that $\Gamma^\alpha \subseteq \Gamma^\beta$ whenever $\alpha < \beta$. By cardinality considerations, there exists a countable ordinal α such that $\Gamma^\alpha = \Gamma^\beta$ for all $\beta > \alpha$. The least such α is called the *closure ordinal* of Γ and will be denoted by $|\Gamma|$. The set $\Gamma^{|\Gamma|}$ is called the closure of Γ or the set inductively defined by Γ and will be denoted by Cl(Γ).

For a monotone operator, the closure is also the least fixed point lfp(Γ) as indicated by the following lemma, see Hinman (1978).

Lemma 2.1. If Γ is a monotone operator, Cl(Γ) is the unique least set C such that $\Gamma(C) = C$. In fact, for any set A, we have $\Gamma(A) \subseteq A$ if and only if $\mathrm{cl}(C) \subseteq A$.

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

For any operator $\Gamma : \mathscr{P}(\mathbb{N}) \to \mathscr{P}(\mathbb{N})$, let $R_\Gamma \subseteq \mathbb{N} \times \mathscr{P}(\mathbb{N})$ be given by $R_\Gamma(m, A) \iff m \in \Gamma(A)$. In general, we say that a predicate $R(x_1, \ldots, x_k, A) \subseteq \mathbb{N}^k \times \mathscr{P}(\mathbb{N})$ is computable if there is an oracle Turing machine M_e such that for any $A \in \mathscr{P}(\mathbb{N})$, M_e with oracle A and input $(x_1, \ldots, x_n)$ outputs 1 if $R(x_1, \ldots, x_n, A)$ holds, and outputs 0 otherwise. The notation of a predicate being $\Sigma^0_n, \Pi^0_n, \Sigma^1_1, \Pi^1_1$, *etc.* can then be defined as usual over the class of computable predicates. We then say that an operator Γ is computable (respectively, Σ^0_1, arithmetical, *etc.*) if the relation R_Γ is computable (respectively, Σ^0_1, arithmetical, *etc.*). The following results are well known.

Theorem 2.1. Let Γ be an inductive operator.

(a) If Γ is computable, the sequence $\{\Gamma^n : n \in \omega\}$ is uniformly computable, $|\Gamma| \leqslant \omega$, and $\mathrm{Cl}(\Gamma)$ is Σ^0_1.
(b) If Γ is Σ^0_1, then $|\Gamma| \leqslant \omega$ and if Γ is monotone Σ^0_1, then $\mathrm{Cl}(\Gamma)$ is Σ^0_1.
(c) Any Σ^0_1 set is 1-1 reducible to the closure of some computable monotone operator.
(d) If Γ is monotone arithmetical, $|\Gamma| \leqslant \omega_1^{\mathrm{CK}}$ (the least non-computable ordinal) and $\mathrm{Cl}(\Gamma)$ is Π^1_1.
(e) Any Π^1_1 set is 1-1 reducible to the closure of a monotone Π^0_1 operator.

Example 2.1. The classic example of a computable monotone operator is given by the definition of the set of sentences of a propositional logic over an infinite set $a_0, a_1, \ldots$ of propositional variables. Identifying sentences p, q with their Gödel number $\mathrm{gn}(p), \mathrm{gn}(q)$, we have for any i, p, q, and A:

(0) $a_i \in \Gamma(A)$.
(1) $\neg p \in \Gamma(A)$ if $p \in A$.
(2) $p \wedge q \in \Gamma(A)$ if $p \in A$ and $q \in A$.
(3) $p \in \Gamma(A)$ if $p \in A$.

Other clauses could be added to include disjunction, implication or other binary connectives. This operator is computable because for any sentence p, we can compute the (at most two) other sentences that need to be in A for p to get into $\Gamma(A)$. Similar computable inductive definitions can be given for the set of terms in a first-order language and the set of formulas in predicate logic. In each case, the closure ordinal of such a Γ is ω and the set of sentences (respectively, terms, formulas) is computable since for any sentence (term, formula) p of length n, $p \in \mathrm{lfp}(\Gamma)$ if and only if $p \in \Gamma^n$.

Example 2.2. Suppose we are given a computable or Σ^0_1 set A_0 of axioms for propositional logic together with the logical axioms $\neg p \vee p$ for each p and a finite set of rules as indicated below. Then the set of consequences of A_0 is generated by the operator Γ where, for all sentences p, q, r and all A:

(0) $p \in \Gamma(A)$ if p is an axiom.
(1) $p \vee q \in \Gamma(A)$ if $p \in A$ or $q \in A$.
(2) $p \in \Gamma(A)$ if $p \vee p \in A$.
(3) $(p \vee q) \vee r \in \Gamma(A)$ if $p \vee (q \vee r) \in A$.
(4) $q \vee r \in \Gamma(A)$ if $p \vee q \in A$ and $\neg p \vee r \in A$.

In this case, Γ is a Σ_1^0 operator but is not computable since, for example, the Cut Rule (4) asks for the existence of a p such that $p \vee q$ and $\neg p \vee r$ are in A.

Now, in this particular case, the consequences of a computable set A_0 will be a computable set but a similar example can be given for first-order logic where the consequences of a finite set of axioms for arithmetic is Σ_1^0 but not computable.

Example 2.3. The one-step provability operator for a computable Horn logic program is a Σ_1^0 monotone operator. That is, suppose A is a computable set of propositional letters or atoms. We assume that $A = \mathbb{N}$. A logic programming clause is a construct of the form

$$C = p \leftarrow q_1, \dots, q_m, \neg r_1, \dots, \neg r_n \tag{1}$$

where $p, q_1, \dots, q_m, r_1, \dots, r_n$ are atoms. Given a clause C, we let

$$[C] = [p, \langle q_1, \dots, q_m \rangle, \langle r_1, \dots, r_n \rangle]$$

where, by convention, we let $\langle q_1, \dots, q_m \rangle = 0$ if $m = 0$ and $\langle r_1, \dots, r_n \rangle = 0$ if $n = 0$. The atoms $q_1, \dots, q_m, \neg r_1, \dots, \neg r_n$ form the *body* of C and the atom p is its *head*. Given a set of atoms $M \subseteq A$, we say M is a model of C if either

(i) there is a q_i such that $q_i \notin M$ or there is an r_j such that $r_j \in M$ (M does not satisfy the body of C); or
(ii) $p \in M$ (M satisfies the head of C).

The clauses C for which $n = 0$ are called *Horn* clauses.

A program P is a set of clauses. We say that P is computable (Σ_1^0, arithmetical, *etc.*) if $\{[C] : C \in P\}$ is computable (Σ_1^0, arithmetical, *etc.*). A program entirely composed of Horn clauses is called a Horn program. If P is a Horn program, there is a one-step provability operator associated with P, $T_P : \mathscr{P}(\mathbb{N}) \to \mathscr{P}(\mathbb{N})$, which is defined by

$T_P(A)$ equals the set of all p such that there exists a clause $C = p \leftarrow q_1, \dots, q_n$ in P such that $q_1, \dots, q_n \in A$.

A Horn program always has a least model, which is the closure of T_p. It is the intended semantics of such a program.

For programs with bodies containing the negation operator *not*, we will use the stable model semantics. Following Gelfond and Lifschitz (1988), we define a *stable model* of the program as follows. Assume M is a collection of atoms. The *Gelfond–Lifschitz reduct* of P by M is a Horn program arising from P by first eliminating those clauses in P that contain $\neg r$ with $r \in M$. In the remaining clauses, we drop all negative literals from the body. The resulting program $GL_M(P)$ is a Horn program. We call M a stable model of P if M is the least model of $GL_M(P)$. In the case of a Horn program, there is a unique stable model, namely, the least model of P. Alternatively, one can define a one-step provability operator $T_{P,M}$ relative to a logic program P consisting of clauses of the form of (1) and a collection of atoms M by defining $T_{P,M}(A)$ to be the set all p such that there exists a

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

clause $C = p \leftarrow q_1, \dots, q_n, \neg r_1, \dots, \neg r_m$ in P such that

(i) $\{q_1, \dots, q_m\} \subseteq A$; and
(ii) $\{r_1, \dots, r_m\} \cap M = \emptyset$.

Then M is a stable model if and only if the closure of $T_{P,M}$ equals M. In general, if M is a computable set, $T_{P,M}$ is a monotone Σ_1^0 operator.

It should be pointed out that both Examples 1 and 2 can be reformulated in the framework of logic programming as computable Horn programs. That is, the set of rules is a computable set, even though the corresponding inductive operator need not be computable.

Example 2.4. Another setting where computable inductive operators arise is in computable algebra and computable model theory. Surveys on various topics in computable algebra and model theory can be found in Ershov *et al.* (1998a; 1998b).

A generic example of computable inductive operators that arise in computable algebra are effective closure systems, which were introduced by Remmel (Remmel 1980). An effective closure system $\mathscr{M} = (M, \mathrm{cl})$ consists of a computable set M of the natural numbers $\mathbb{N}$ together with an operation $\mathrm{cl} : \mathscr{P}(M) \to \mathscr{P}(M)$, where $\mathscr{P}(M)$ denotes the power set of M, which satisfies the following:

(i) $A \subseteq \mathrm{cl}(A)$.
(ii) $A \subseteq B$ implies $\mathrm{cl}(A) \subseteq \mathrm{cl}(B)$.
(iii) $\mathrm{cl}(\mathrm{cl}(A)) = \mathrm{cl}(A)$.
(iv) $x \in \mathrm{cl}(A)$ implies that for some finite $A' \subseteq A$, $x \in \mathrm{cl}(A')$.

Furthermore, we require that cl is effective on (indices of) finite sets. That is, we assume that there is an effective algorithm that, given $x, y_1, \dots, y_n \in M$, will decide whether or not $x \in \mathrm{cl}(y_1, \dots, y_n)$, where $\mathrm{cl}(y_1, \dots, y_n)$ denotes $\mathrm{cl}(\{y, \dots, y_n\})$. Note that this condition plus conditions (i)–(iv) ensure that such closure operators are at least Σ_1^0 monotone operators.

We also assume that$(\mathscr{M}, \mathrm{cl})$ always satisfy the non-triviality axiom (v):

(v) $\mathrm{cl}(\emptyset) \neq^* M$.

Here we write $A =^* B$ if there exist finite sets, E and F, such that $\mathrm{cl}(A \cup E) = \mathrm{cl}(B \cup F)$. Similarly, we write that $A \subseteq^* B$ if there is a finite set F such that $A \subseteq \mathrm{cl}(B \cup F)$.

We say V is a *substructure* of $\mathscr{M}$, or V is *closed*, if $V \subseteq M$ and $\mathrm{cl}(V) = V$. It is easy to see that both the set of c.e. substructures and the set of all substructures of $\mathscr{M}$ form a lattice, where the meet operation is just the set theoretic intersection and the join of two substructures V and W, denoted $V + W$, is given by $V + W = \mathrm{cl}(V \cup W)$. We use $L(\mathscr{M})$ to denote the lattice of c.e. substructures of $\mathscr{M} = (M, \mathrm{cl})$ and $S(\mathscr{M})$ to denote the lattice of all substructures of $\mathscr{M}$.

If $\mathscr{M}$ also satisfies

(vi) (Exchange) $x \in \mathrm{cl}(A \cup \{y\}) - \mathrm{cl}(A)$ implies $y \in \mathrm{cl}(A \cup \{x\})$,

we say $\mathscr{M}$ is an *effective Steinitz system.* Effective Steinitz systems have been extensively studied: see Nerode and Remmel (1982; 1983), Downey (1983a; 1983b) and Baldwin (1982; 1984).

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

Effective algebras form another natural class of examples. These are obtained as follows. Let (M, R) be an effective universal algebra in the sense that M is a computable set and R is a computable set of uniformly computable operations on M. Then we naturally associate an effective closure system (M, cl_R) with (M, R) by setting $\mathrm{cl}_R(A)$ to be the closure of A under the operations of R and their projections. We say that an effective closure system $\mathscr{M}$ formed in this way is an *effective algebra*. As we shall see, most natural examples, such as groups, rings, fields and vector spaces, are effective algebras.

However, not all effective closure systems are effective algebras. For example, for any effective closure system $\mathscr{M} = (M, \mathrm{cl})$, we can define an *intersection subsystem* (A, cl^*_A) for $A \subseteq M$ where for any $B \subseteq A$,

$$\mathrm{cl}^*_A(B) = \mathrm{cl}(B) \cap A.$$

It is easy to check that (A, cl^*_A) is an effective closure system, but not necessarily an effective algebra.

We conclude this example with a partial list of some specific examples of effective closure systems that have been studied extensively in the literature. In particular, there has been considerable work on the lattice of c.e. substructures of various structures. Details can be found in the survey article Nerode and Remmel (1985). Some general results on the lattice of substructures of effective closure systems can be found in Downey and Remmel (1998). Here we shall only give a brief description of the closure systems; refer to Nerode and Remmel (1985) or Downey and Remmel (1998) for more details.

Sets. Let $\mathscr{M} = (\omega, \mathrm{cl})$ where $\mathrm{cl}(A) = A$. In this case $L(\mathscr{M})$ is the lattice of c.e. sets. Clearly, cl is a computable monotone operator in this case.

Vector spaces. Let V_∞ denote a fully effective infinite dimensional vector space over a computable field. That is, V_∞ consists of a computable subset U of ω with computable operations for addition and scalar multiplication on V_∞. Moreover, we assume that V_∞ has an effective dependence algorithm, that is, there is a uniform algorithm that given any $x, y_1, \ldots, y_n$ in U, decides whether or not $x \in (\{y_1, \ldots, y_n\})^*$, where $(A)^*$ denotes the subspace generated by A. In this case, $\mathrm{cl}(A) = (A)^*$ and $L(V_\infty)$ is the lattice of c.e. subspaces.

In this case, cl is a Σ^0_1 monotone operator, but it is not computable. This follows from a result of Dekker (Dekker 1971), which says that every c.e. subspace V of V_∞ has a computable basis B. Thus, since there are c.e. subspaces that are not computable, it follows that the relation R_{cl} is only Σ^0_1. Similar results hold for the remaining examples of closure operators given below.

Fields. Here F_∞ denotes a fully effective algebraically closed field with infinite computable transcendence base, and $\mathrm{cl}(A)$ denotes the algebraic closure of A.

Affine spaces. In this case $\mathscr{M} = (V_\infty, K\ell)$ where V_∞ a computable vector space over a computable ordered field. We define $y \in K\ell(y_1, \ldots, y_n)$ if and only if $y = \Sigma\lambda_i y_i$ with $\Sigma\lambda_i = 1$. Again this is a Steinitz algebra. We denote its lattice of c.e. affine subspaces by $L(V_\infty, K\ell)$ to distinguish it from $L(V_\infty)$ (*cf.* Downey (1983b)).

Locally computable rings and modules. Many other computable rings and modules are effective closure systems. For example, consider $G = \oplus_{i\in\omega}\mathbb{Z}$, the free Abelian group on ω generators.

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

Subalgebras of Boolean Algebras (Remmel 1978; 1980). A computable Boolean algebra $\mathscr{B} = (B, \vee_{\mathscr{B}}, \wedge_{\mathscr{B}}, \neg_{\mathscr{B}})$ consists of a computable subset B of ω and computable operations for the meet, $\wedge_{\mathscr{B}}$, join, $\vee_{\mathscr{B}}$, and complement, $\neg_{\mathscr{B}}$, operations, which turn B into a Boolean algebra. In this case, $\mathrm{cl}(A)$ is the subalgebra generated by A.

Convex sets, $K(V_\infty)$. Finally, consider the structure $K(V_\infty) = (V_\infty, \langle\rangle)$ from Kalantari (1981) and Downey (1984). Here we consider V_∞ where the underlying field is the rationals, Q, and $\langle\rangle$ is the operation of taking the convex hull, *viz*.,

$$\langle\{x_1, \ldots, x_n\}\rangle = \{y | y = \Sigma\lambda_i x_i \text{ with } \Sigma\lambda_i = 1 \text{ and } 0 \leqslant \lambda_i \leqslant 1\}.$$

Then $(V_\infty, \langle\rangle)$ is obviously an effective closure system.

We note that in all the structures above, we can generate many classes of Σ^0_1 inductive operators by simply letting A be any computable or c.e. subset of the structure and defining a new closure operator Γ_A by $\Gamma_A(S) = \mathrm{cl}(A \cup S)$.

3. Index sets for Σ^0_1 and computable monotone operators

An important property of Σ^0_1 monotone operators Γ is that the relation $m \in \Gamma(A)$ depends only on positive information about A. That is, we have the following lemma.

Lemma 3.1 (Hinman 1978, page 92). For any Σ^0_1 monotone operator Γ, there is a computable relation R such that, for all $m \in \mathbb{N}$ and $A \in \mathscr{P}(\mathbb{N})$,

$$m \in \Gamma(A) \iff (\exists n)(D_n \subseteq A \ \&\ R(m,n)). \tag{2}$$

It follows from Lemma 3.1 that the Σ^0_1 monotone inductive operators may be effectively enumerated as $\Gamma_0, \Gamma_1, \ldots$ in the following manner. For all $e, m \in \mathbb{N}$ and all $A \in \mathscr{P}(\mathbb{N})$, let

$$m \in \Gamma_e(A) \iff (\exists n)[D_n \subseteq A \text{ and } \langle m,n\rangle \in W_e].$$

Lemma 3.2.

(a) There is a primitive recursive function f such that for all m, e, a

$$\Gamma_e(W_a) = W_{f(e,a)}.$$

(b) The relation $m \in \Gamma^t_e$ is Σ^0_1 in m, e and t.

(c) The relation $m \in \mathrm{lfp}(\Gamma_e)$ is Σ^0_1 in m and e.

(d) There is a computable function h such that $\mathrm{lfp}(\Gamma_e) = W_{h(e)}$.

Proof.

(a) We have

$$m \in \Gamma_e(W_a) \iff (\exists n)[D_n \subseteq W_a \text{ and } \langle m,n\rangle \in W_e].$$

So we may define a partial computable function ϕ_c such that to compute $\phi_c(e,a,m)$, we search for the least pair $\langle n,s\rangle$ such that $D_n \subseteq W_{a,s}$ and $[m,n] \in W_{e,s}$. If we find such a pair, we set $\phi_c(e,a,m) = 1$; otherwise, $\phi_c(e,a,m)$ is undefined. Then

$$m \in \Gamma_e(W_a) \iff [e,a,m] \in \mathrm{Dom}(\phi_c).$$

Now the *s-m-n* theorem will provide a primitive recursive f such $\phi_{f(e,a)}(m) = \phi_c(e,a,m)$.

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

(b) Let $W_0 = \varnothing$ and let f be given by (a). For any fixed e, let g_e be the partial computable function defined by $g_e(a) = f(e,a)$. Then clearly, $\Gamma_e^t = W_{g_e^t(0)}$.
(c) This follows from the fact that $m \in \text{lfp}(\Gamma_e) \iff (\exists t)(m \in \Gamma_e^t)$.
(d) This follows from part (c) by the *s-m-n* theorem. □

Theorem 3.1. Fix an infinite c.e. set W. Then $\{e : W \cap \text{lfp}(\Gamma_e) \text{ is computable}\}$ is Σ_3^0 complete.

Proof. We make use of the well-known fact that $\text{Rec} = \{e : W_e \text{ is computable}\}$ is Σ_3^0 complete (Soare 1987). Let ψ be a computable function such that $W \cap W_e = W_{\psi(e)}$ for all e. Now let $C = \{e : W \cap \text{lfp}(\Gamma_e) \text{ is computable}\}$ and h be the computable function defined in the proof of Lemma 3.2.(d). Then $e \in C \iff \psi(h(e)) \in \text{Rec}$, so C is a Σ_3^0 set.

For the completeness, first consider the case where $W = \mathbb{N}$. We can use the *s-m-n* theorem to obtain a 1:1 computable function g such that

$$\langle m, s\rangle \in \Gamma_{g(e)}(A) \iff m \in W_{e,s} \text{ or } \langle m, s+1\rangle \in A.$$

It is easy to see that $\text{lfp}(\Gamma_{g(e)}) = W_e \times \mathbb{N}$, so W_e is computable if and only if $\text{lfp}(\Gamma_{g(e)})$ is computable. Hence, g witnesses the fact that Rec is 1:1 reducible to C since $e \in \text{Rec} \iff g(e) \in C$. Thus C is Σ_3^0 complete.

For an arbitrary infinite c.e. set W, let R be an infinite computable subset of W and let f be an increasing, computable function such that $R = \{f(0), f(1), \ldots\}$. Then, for any e, let $W_{p(e)} = \{f(i) : i \in W_e\}$, and observe that $W_{p(e)} \subset W$ for all e and that $W_{p(e)}$ is computable if and only if W_e is computable. It follows that W_e is computable if and only if $W \cap \text{lfp}(\Gamma_{g(p(e))})$ is computable. Thus $g \circ p$ shows that, in general, Rec is 1:1 reducible to C, so C is Σ_3^0-complete for all W. □

Computable operators are continuous and we can use the indexing of Cenzer and Remmel (1999, page 135) to define the e-th computable monotone operator Δ_e for e in the Π_2^0 set of indices such that ϕ_e is a total function. That is, let $\sigma_0, \sigma_1, \ldots$ enumerate the set $\{0,1\}^*$ of finite strings of 0's and 1's. For $\sigma, \tau \in \{0,1\}^*$, we write $\sigma \sqsubseteq \tau$ if σ is an initial segment of τ and write $\sigma \subseteq \tau$ if $\{i : \sigma(i) = 1\} \subseteq \{i : \tau(i) = 1\}$. Then the partial computable function $\phi_e : \mathbb{N} \to \mathbb{N}$ defines a computable monotone operator $\Delta_e : \mathscr{P}(\mathbb{N}) \to \mathscr{P}(\mathbb{N})$ if it satisfies the following four conditions.

(1) $(\forall m)(\exists n)[\phi_e(m) = n]$, that is, ϕ_e is total.
(2) $(\forall m)(\forall n)[\sigma_m \sqsubseteq \sigma_n \longrightarrow \sigma_{\phi_e(m)} \sqsubseteq \sigma_{\phi_e(n)}]$.
(3) $(\forall m)(\exists n)(\forall \sigma_i \in \{0,1\}^n)[|\sigma_{\phi_e(i)}| \geqslant m]$.
(4) $(\forall m)(\forall n)[\sigma_m \subseteq \sigma_n \longrightarrow \sigma_{\phi_e(m)} \subseteq \sigma_{\phi_e(n)}]$.

The first three clauses above simply define the set of indices of computably continous functions from $\{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$. Then clause (4) ensures that the resulting operator is monotone. Let ICM denote the set of indices e satisfying (1)–(4). For $A \subseteq \mathbb{N}$ and $n \in \mathbb{N}$, identify A with its characteristic function and let $A_n = i$ where $\sigma_i = A\lceil n = (A(0), A(1), \ldots, A(n-1))$. Then we may define the e-th computable monotone operator by declaring that

$$m \in \Delta_e(A) \iff (\exists n)(\forall \sigma_i \in \{0,1\}^n)[|\sigma_{\phi_e(i)}| \geqslant m \ \& \ \sigma_{\phi_e(A_n)}(m) = 1]. \tag{3}$$

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

Note that if ϕ_e satisfies conditions (1)–(4), then $\Delta_e(A)$ also has a Π^0_1 definition, namely,

$$m \in \Delta_e(A) \iff (\forall n)[(\forall \sigma_i \in \{0,1\}^n)[|\sigma_{\phi_e(i)}| \geqslant m] \longrightarrow \sigma_{\phi_e(A_n)}(m) = 1]. \tag{4}$$

Theorem 3.2. The set ICM of indices of computable monotone operators is Π^0_2 complete.

Proof. It is clear that ICM is a Π^0_2 set. For the completeness, we define a reduction of the Π^0_2 complete set $Tot = \{e : \phi_e \text{ is total}\}$ to ICM as follows. Let f be the computable function such that for any i, we have $\phi_{f(e)}(i) = j$ where

$$\sigma_j = (\phi_e(0), \phi_e(1), \dots, \phi_e(|\sigma_i| - 1)).$$

Now, if $e \notin Tot$, then, clearly, $\phi_{f(e)}$ is not total, so $f(e) \notin$ ICM. However, if $e \in Tot$, it is easy to see that for all A, we have $\Delta_{f(e)}(A) = \{m : \phi_e(m) = 1\}$, so $\Delta_{f(e)}$ is a computable monotone operator. Thus, $e \in Tot \iff f(e) \in$ ICM. □

Lemma 3.3. There is a primitive computable function g such that for all $e \in$ ICM, $\Delta_e = \Gamma_{g(e)}$.

Proof. Define $\langle m, n\rangle \in W_{g(e)}$ if and only if $(\exists k)(\forall \sigma_i \in \{0,1\}^k)[|\sigma_{\phi_e(i)}| > m]$ and there exists $\sigma_i \in \{0,1\}^k$ such that $\sigma_{\phi_e(i)}(m) = 1$ and $\{j : \sigma_i(j) = 1\} \subseteq D_n$. We now verify that $\Delta_e = \Gamma_{g(e)}$ if $e \in$ ICM.

Suppose first that $m \in \Delta_e(A)$. Then we find the least k such that

$$(\forall \sigma_i \in \{0,1\}^k)[|\sigma_{\phi_e(i)}| > m]\,.$$

Thus, for $\sigma_i = A\lceil k$, we have $\sigma_{\phi_e(i)}(m) = 1$. Now let

$$D_n = A \cap \{0, 1 \dots, k-1\} = \{j < k : \sigma_i(j) = 1\}.$$

It follows that $\langle m, n\rangle \in W_{g(e)}$, so $m \in \Gamma_{g(e)}(A)$.

Next suppose that $m \in \Gamma_{g(e)}(A)$ and let n, k and $\sigma_i \in \{0,1\}^k$ be given as above so that $\sigma_{\phi_e(i)}(m) = 1$ and $\{j : \sigma(j) = 1\} \subseteq D_n \subseteq A$. It follows from clause (4) above that $\sigma_{\phi(e)(A_k)}(m) = 1$ and, therefore, $m \in \Delta_e(A)$.

Hence we have shown that, for all A, we have $\Delta_e(A) = \Gamma_{g(e)}(A)$ and hence $\Delta_e = \Gamma_{g(e)}$. □

Lemma 3.4.

(a) There is a partial computable function δ such that for all m, e, a with $a \in Tot$ and $e \in$ ICM, we have $\delta(e, a) \in Tot$ and $\Delta_e(\{m : \phi_a(m) = 1\}) = \{m : \phi_{\delta(e,a)}(m) = 1\}$.

(b) There is a partial computable function ψ such that for all e, t with $e \in$ ICM, we have $\phi_{\psi(e,t)}$ is the characteristic function of Δ^t_e.

(c) There is a Σ^0_1 relation S such that

$$m \in \mathrm{lfp}(\Delta_e) \iff \langle m, e\rangle \in S.$$

Proof.

(a) To compute $\phi_{\delta(e,a)}(m)$, first find k so that $|\sigma_{\phi_e(i)}| > m$ for all $\sigma_i \in \{0,1\}^k$. Then let $\sigma_i = (\phi_a(0), \phi_a(1), \dots, \phi_a(k-1))$ and set $\phi_{\delta(e,a)}(m) = \sigma_{\phi_e(i)}(m)$.

Parts (b) and (c) follow easily. □

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

This shows that the closure of any computable monotone inductive operator is a c.e. set. In Cenzer (1978), the first author considered the converse problem of whether any c.e. set is the closure of some computable monotone inductive operator. It is shown there that not every c.e. set is the closure of such an operator, but that every c.e. set is one-one reducible to such a closure. Here is an index set version of that result.

Theorem 3.3. There are primitive recursive functions f and g such that for all e and m, we have $f(e) \in \mathrm{ICM}$ and $m \in W_e \iff g(m) \in \mathrm{lfp}(\Delta_{f(e)})$.

Proof. Define the computable monotone inductive operator $\Delta_{f(e)}$ by

$$\langle m, s\rangle \in \Delta_{f(e)}(A) \iff [m \in W_{e,s} \ \vee \ \langle m, s+1\rangle \in A].$$

It is easy to see that $\mathrm{lfp}(\Delta_{f(e)}) = \{\langle m, s\rangle : m \in W_e\}$, so for any m and e,

$$m \in W_e \iff \langle m, 0\rangle \in \mathrm{lfp}(\Delta_{f(e)}).$$

Thus we can take $g(m) = \langle m, 0\rangle$. □

The index set complexity for Σ^0_1 operators given in Theorem 3.1 easily carries over for computable monotone operators since the operator $\Gamma_{g(e)}$ defined in the proof is uniformly computable. Thus we have the following theorem.

Theorem 3.4. $\{e : \mathrm{lfp}(\Delta_e) \text{ is computable}\}$ is Σ^0_3 complete.

For the rest of this section, we consider the complexity of two types of index sets associated with monotone operators. The first type comes from the cardinality of the least fixed point. For example, we will determine the complexity of the problem of deciding whether $\mathrm{lfp}(\Gamma_e)$ is a finite or an infinite set. The second type comes from the closure ordinal of the operator. For example, we will determine the complexity of the problem of deciding whether the closure ordinal of Δ_e is finite or equals ω. For the remaining results in this section, we will omit the routine verifications of the complexity upper bounds.

Theorem 3.5. $\{e : |\Gamma_e| > 0\} = \{e : \mathrm{lfp}(\Gamma_e) \neq \varnothing\}$ is Σ^0_1 complete and $\{e : |\Gamma_e| = 0\} = \{e : \mathrm{lfp}(\Gamma_e) = \varnothing\}$ is Π^0_1 complete.

Proof. For the completeness, let E be an arbitrary c.e. set and define a computable function f_E so that

$$m \in \Gamma_{f_E(e)}(A) \iff (m = 0 \ \& \ e \in E).$$

Clearly, if $e \notin E$, then $|\Gamma_{f_E(e)}| = 0$ and $\mathrm{lfp}(\Gamma_{f_E(e)}) = \varnothing$, and if $e \in E$, then $|\Gamma_{f_E(e)}| = 1$ and $\mathrm{lfp}(\Gamma_{f_E(e)}) = \{0\}$. Thus, f_E shows that the arbitrary Σ^0_1 set E is 1:1 reducible to $\{e : |\Gamma_e| > 0\}$, and at the same time $\mathbb{N} - E$ is 1:1 reducible to $\{e : |\Gamma_e| = 0\}$. □

A set is said to be *d.c.e.* if it is a difference of two c.e. sets.

Theorem 3.6. For any natural number $k > 0$,

(a) $\{e : \mathrm{card}(\mathrm{lfp}(\Gamma_e)) > k\}$ is Σ^0_1 complete and
$\{e : \mathrm{card}(\mathrm{lfp}(\Gamma_e)) \leqslant k\}$ is Π^0_1 complete.
(b) $\{e : \mathrm{card}(\mathrm{lfp}(\Gamma_e) = k\}$ is d.c.e. complete.

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

Proof.

(a) For the completeness, modify the definition of f_E in the proof of Theorem 3.5 so that

$$m \in \Gamma_{f_E(e)}(A) \iff [m \leqslant k \;\&\; e \in E].$$

Then $\mathrm{lfp}(\Gamma_{f_E(e)}) = \emptyset$ if $e \notin E$ and $\mathrm{lfp}(\Gamma_{f_E(e)}) = \{0, 1, \ldots, k\}$ if $e \in E$. Again f_e shows that E is 1:1 reducible to $\{e : \mathrm{card}(\mathrm{lfp}(\Gamma_e)) > k\}$ and, hence, $\{e : \mathrm{card}(\mathrm{lfp}(\Gamma_e)) > k\}$ is Σ_1^0 complete.

(b) Clearly, $\{e : \mathrm{card}(\mathrm{lfp}(\Gamma_e)) = k\} = \{e : \mathrm{card}(\mathrm{lfp}(\Gamma_e)) \leqslant k\} - \{e : \mathrm{card}(\mathrm{lfp}(\Gamma_e)) \leqslant k - 1\}$. For completeness, we need only show that for any c.e. sets C and D with $D \subseteq C$, there is 1:1 computable function g such that $e \in C - D \iff g(e) \in \{e : \mathrm{card}(\mathrm{lfp}(\Gamma_e) = k\}$. So let C and D be c.e. sets where $D \subseteq C$ and define g so that

$$m \in \Gamma_{g(e)}(A) \iff [(m < k \;\&\; e \in C) \;\vee\; (m = k \;\&\; k - 1 \in A \;\&\; e \in D)].$$

If $e \notin C$, then $\mathrm{lfp}(\Gamma_{g(e)}) = \emptyset$. If $e \in C - D$, then $|\Gamma_{g(e)}| = k$ and $\mathrm{lfp}(\Gamma_{g(e)}) = \{0, 1, \ldots, k-1\}$. If $e \in C \cap D$, then $|\Gamma_{g(e)}| = 2$ and $\mathrm{lfp}(\Gamma_{g(e)}) = \{0, 1, \ldots, k\}$. □

Theorem 3.7.

(a) $\{e : \mathrm{lfp}(\Gamma_e) \text{ is finite}\}$ is Σ_2^0 complete and $\{e : \mathrm{lfp}(\Gamma_e) \text{ is infinite}\}$ is Π_2^0 complete.

(b) $\{e : \mathrm{lfp}(\Gamma_e) \text{ is cofinite}\}$ is Σ_3^0 complete and $\{e : \mathrm{lfp}(\Gamma_e) \text{ is coinfinite}\}$ is Π_3^0 complete.

Proof. The statements follow easily from the fact that $\{e : W_e \text{ is finite}\}$ is Σ_2^0 complete and that $\{e : W_e \text{ is cofinite}\}$ is Σ_3^0 complete by letting $\Gamma_{f(e)}(A) = W_e$ for all A. □

The corresponding result for computable monotone operators is a corollary.

Theorem 3.8.

(a) $\{e : \mathrm{lfp}(\Delta_e) \text{ is infinite}\}$ is Π_2^0 complete.

(b) $\{e : \mathrm{lfp}(\Delta_e) \text{ is cofinite}\}$ is Σ_3^0 complete and $\{e : \mathrm{lfp}(\Delta_e) \text{ is coinfinite}\}$ is Π_3^0 complete.

Next we consider the closure ordinal of a monotone inductive operator.

Theorem 3.9. For any natural number $t \geqslant 1$:

(a) $\{e : |\Gamma_e| > t\}$ is Σ_2^0 complete and $\{e : |\Gamma_e| \leqslant t\}$ is Π_2^0 complete.

(b) $\{e : |\Gamma_e| = 1\}$ is Π_2^0 complete.

(c) $\{e : |\Gamma_e| = t + 1\}$ is D_2^0 complete.

Proof. We will use the fact that $\mathrm{Fin} = \{e : W_e \text{ is finite}\}$ is a Σ_2^0 complete set. We can define a 1:1 computable function f such that

$$m \in \Gamma_{f(e)}(A) \iff m = 0 \;\vee\; (\exists n \leqslant m)(n \in A) \vee\; (\exists n \geqslant m)(n \in W_e).$$

If W_e is infinite, $\Gamma_{f(e)}^1 = \mathbb{N}$ and $|\Gamma_e| = 1$. If W_e is finite, let M be the largest element of $W_e \cup \{0\}$. Then $\Gamma_{f(e)}^1 = \{0, 1, \ldots, M\}$, $\Gamma_{f(e)}^2 = \mathbb{N}$ and, therefore, $|\Gamma_{f(e)}| = 2$. Thus $e \in \mathrm{Fin} \iff f(e) \in \{e : |\Gamma_e| > 1\}$, which establishes completeness for part (a) when $t = 1$ and the completeness of part (b).

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

For the completeness in part (a), fix $t \geqslant 1$ and define a 1:1 computable function g such that $m \in \Gamma_{g(e)}(A)$ if and only if

$$m = 0 \ \vee \ (m < t \ \& \ m - 1 \in A) \vee \ (m \geqslant t \ \& \ [(\exists n \geqslant m)(n \in W_e) \vee \ (t - 1 \in A)]).$$

Then it is easy to see that if W_e is infinite, for all i,

$$\Gamma^i_{g(e)} = \{x : x < i \vee x \geqslant t\},$$

so $|\Gamma_{g(e)}| = t$ and $\text{lfp}(\Gamma_{g(e)}) = \mathbb{N}$. However, if W_e is finite and M is the largest element of W_e, we have, for $i \leqslant t$,

$$\Gamma^i_{g(e)} = \{x : x < i \ \vee \ t \leqslant x \leqslant M\}$$

and

$$\Gamma^{t+1}_{g(e)} = \mathbb{N},$$

so $|\Gamma_{g(e)}| = t + 1$. Thus $e \in \text{Fin} \iff g(e) \in \{e : |\Gamma_e| > t\}$.

For the completeness in part (c) in the case where $t = 1$, it suffices to define a computable function h such that $|\Gamma_{h(a,b)}| = 2$ if and only if W_a is finite and W_b is infinite. Let Ev denote the set of even numbers and Od denote the set of odd numbers. First define $h(a, b)$ so that

$$2m \in \Gamma_{h(a,b)}(A) \iff m = 0 \ \vee \ (\exists n \leqslant m)(n \in A \vee \ (\exists n \geqslant m)(n \in W_a)).$$

Then, by our argument for case (a), $Ev \subseteq \Gamma^1_{h(a,b)}$ if W_a is infinite. If W_a is finite and M is the greatest element of $W_a \cup \{0\}$, then $\Gamma^1_{h(a,b)} \cap Ev = \{2x : x \leqslant M\}$ and $Ev \subseteq \Gamma^2_{h(a,b)}$. We then complete the definition of h so that

$$\begin{aligned} 2m + 1 \in \Gamma_{h(a,b)}(A) \iff & [m = 0 \vee (\exists n \geqslant m)(n \in W_a)] \\ & \vee[0 \in A \ \& \ (\exists n \geqslant m)(n \in W_b)] \\ & \vee[0 \in A \ \& \ \ m > 0 \ \& \ (2m - 1 \in A)]. \end{aligned}$$

Now if W_a is infinite, $0d \subseteq \Gamma^1_{h(a,b)}$, so $\Gamma^1_{h(a,b)} = \mathbb{N}$ and $|\Gamma_{h(a,b)}| = 1$. Next suppose that W_a is finite and M is the greatest element of $W_a \cup \{0\}$. Then our definition of h ensures that $\Gamma^1_{h(a,b)} \cap 0d = \{2x + 1 : x \leqslant M\}$ since $0 \notin \Gamma^0_{h(a,b)}$. Now, if W_b is infinite, $Od \subseteq \Gamma^2_{h(a,b)}$, so $\Gamma^2_{h(a,b)} = \mathbb{N}$ and $|\Gamma_{h(a,b)}| = 2$. Finally, if W_b is finite and B is the largest element of $W_a \cup W_b \cup \{0\}$, we have $\Gamma^2_{h(a,b)} \cap 0d = \{2x+1 : x \leqslant B\}$ and $2B + 3 \in \Gamma^3_{h(a,b)}$, so $|\Gamma_{h(a,b)}| \geqslant 3$. This shows that $\{e : |\Gamma_e| = 2\}$ is D^0_2 complete.

For the general case of part (c), fix $t > 1$ and define h so that

$$\begin{aligned} 2m \in \Gamma_{h(a,b)}(A) \iff & m = 0 \vee (m < t \ \& \ m - 1 \in A) \\ & \vee \ (m \geqslant t \ \& \ [(\exists n \geqslant m)(n \in W_e \vee \ 2(t - 1) \in A)]). \end{aligned}$$

Then we can argue as in case (a) that $Ev \subseteq \Gamma^t_{h(a,b)}$ if W_a if infinite. On the other hand, if W_a is finite and M is the largest element of $W_a \cup \{0\}$, then

$$\begin{aligned} \Gamma^t_{h(a,b)} \cap Ev \ &= \ \{0, \ldots, 2(t - 1)\} \cup \{2x : M \geqslant x \geqslant t\} \text{ and} \\ \Gamma^{t+1}_{h(a,b)} \cap Ev \ &= \ Ev. \end{aligned}$$

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

We now complete the definition of h so that

$$2m+1 \in \Gamma_{h(a,b)}(A) \iff [m = 0 \vee (\exists n \geqslant m)(n \in W_a)] \\ \vee[2(t-1) \in A \;\&\; (\exists n \geqslant m)(n \in W_a)] \\ \vee\; [m > 0 \;\&\; 2(t-1) \in A \;\&\; 2m-1 \in A].$$

It can then be verified that W_a is finite and W_b is infinite if and only if $\Gamma_{h(a,b)}| = t+1$. □

Theorem 3.10. $\{e : |\Gamma_e| = \omega\}$ is Π^0_3 complete and $\{e : |\Gamma_e| < \omega\}$ is Σ^0_3 complete.

Proof. We use the Σ^0_3 completeness of Cof $= \{e : W_e$ is cofinite$\}$. We define a 1:1 computable function f so that W_e is cofinite if and only if $|\Gamma_{f(e)}| < \omega$. Define f so that

$$2n \in \Gamma_{f(e)}(A) \iff n = 0 \vee 2n-2 \in A \;\vee\; 2n+1 \in A;$$
$$2n+1 \in \Gamma_{f(e)}(A) \iff (\exists m > n)(2m+1 \in A) \\ \vee\; (\exists m < n)[2m \in A \;\&\; (\forall i \leqslant n)(m \leqslant i \longrightarrow i \in W_e)].$$

We make the following observations. First, $\Gamma^1_{f(e)} = \{0\}$ for all e. Next, it is easy to see by the first of our two conditions defining f that we certainly have $2n \in \Gamma^{n+1}_{f(e)}$ for all n and e and, moreover, $2n \in \Gamma^t_{f(e)}$ for $n \geqslant t$ if and only if $2n+1 \in \Gamma^{t-1}_{f(e)}$. Thus, if Ev is the set of even numbers, $Ev \subseteq \mathrm{lfp}(\Gamma_{f(e)})$ for all e.

Now fix e and let $\Gamma = \Gamma_{f(e)}$. First suppose that W_e is cofinite and M is the smallest natural number such that $i \in W_e$ for all $i \geqslant M$. It follows from our second condition defining f that, since $2M \in \Gamma^{M+1}$, we have $2n+1 \in \Gamma^{M+2}$ for all $n \geqslant M$. But then it is easy to see that $2n+1 \in \Gamma^{M+3}$ for all n and $2n \in \Gamma^{M+4}$ for all n. Thus $\mathrm{lfp}(\Gamma) = \mathbb{N}$ and $|\Gamma| \leqslant M+4$. On the other hand, suppose that $|\Gamma| = k$ is finite. It follows that $2n \in \Gamma^k$ for all n. Let $t \leqslant k$ be the least value such that $\{n : 2n \in \Gamma^t\}$ is infinite. By our observations above, $t > 1$, so let M be the maximum of $\{m : 2m \in \Gamma^{t-1}\}$. Thus, for infinitely many $n \geqslant t$, we have $2n \in \Gamma^t$, so $2n+1 \in \Gamma^{t-1}$. Now let s be the least $k \leqslant t-1$ such that $\{n : 2n+1 \in \Gamma^k\}$ is infinite. Again it must be the case that $s > 1$, so Γ^{s-1} must be finite. Now let p be the largest element such that $2p \in \Gamma^{s-1}$. Because $\{n : 2n+1 \in \Gamma^s\}$ is infinite, it must be the case that for arbitrarily large n, there is an $m \leqslant p$ such that $2m \in \Gamma^{s-1}$ and $i \in W_e$ for $m \leqslant i \leqslant n$. But this implies that W_e is coinfinite. □

The operator $\Gamma_{f(e)}$ defined in the proof of Theorem 3.10 does not define a computable monotone operator, so we cannot conclude that $\{e : |\Delta_e| = \omega\}$ is Π^0_3 complete. In fact, $\{e : |\Delta_e| = \omega\}$ is Π^0_2 complete, as our next result shows.

Theorem 3.11. $\{e : e \in \mathrm{ICM} \;\&\; |\Delta_e| = \omega\}$ is Π^0_2 complete.

Proof. We define a 1:1 computable function f such that for all e, we have $f(e) \in ICM$ and W_e is finite if and only if $|\Delta_{f(e)}| < \omega$. The desired f is the function defined in the proof of Theorem 3.3 where

$$\langle m,s\rangle \in \Delta_{f(e)}(A) \iff [m \in W_{a,s} \;\vee\; \langle m, s+1\rangle \in A].$$

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

Suppose first that W_e is infinite. Then there are arbitrarily large m and s such that $m \in W_{e,s+1} - W_{e,s}$, and, therefore, $\langle m, 0\rangle \in \Delta^{s+2}_{f(e)} - \Delta^{s+1}_{f(e)}$. Thus $|\Delta_{f(e)}| = \omega$. On the other hand, if W_e is finite, there is a finite s such that $m \in W_e$ implies $m \in W_{e,s}$ for all m. It follows that $|\Delta_{f(e)}| \leqslant s+1$, and is finite. □

4. Weakly finitary monotone operators

It follows from Lemma 2.1 that any Σ^0_1 monotone inductive operator Γ is *finitary*, that is, for any x and any set A, we have $x \in \Gamma(A)$ if and only if there is a finite subset D of A such that $x \in \Gamma(D)$. The idea of a weakly finitary operator is to have a finite set $m_1, \ldots, m_k$ of *exceptional* numbers that may be put into $\Gamma(A)$ when an *infinite* set is included in A. If there are exactly k exceptional numbers, the operator Γ will be called k-weakly finitary. For example, we might allow some finite number of consequences of the ω-rule in a subsystem of Peano arithmetic and still obtain a c.e. theory.

Definition 4.1.

(1) We say that a monotone inductive operator $\Gamma : \mathscr{P}(\mathbb{N}) \to \mathscr{P}(\mathbb{N})$ is *weakly finitary* if there is a finite set S_Γ such that for all A:

  (a) $x \notin S_\Gamma$ and $x \in \Gamma(A)$ implies there exists a finite set $F \subseteq A$ such that $x \in \Gamma(F)$.

  (b) $x \in S_\Gamma$ and there is a family $\mathscr{F}_{\Gamma,x}$ of subsets of $\mathbb{N}$ that includes at least one infinite subset of $\mathbb{N}$ such that $x \in \Gamma(A)$ implies there exists an $F \subseteq A$ such that $x \in \Gamma(F)$ for some $F \in \mathscr{F}_{\Gamma,x}$.

  If $|S_\Gamma| = k$, we say that Γ is *k-weakly finitary*.

(2) We say $\Gamma = \Lambda_{k,e}$ is a k-weakly finitary Σ^0_1 monotone inductive operator with index $\langle k, e\rangle = \langle k, \langle d, \langle m_1, e_1, \ldots, m_k, e_k\rangle\rangle$ if:

  (i) Γ is a weakly finitary monotone operator with $S_\Gamma = \{m_1 < \cdots < m_k\}$.

  (ii) For all $m_i \in S_\Gamma$, $\mathscr{F}_{\Gamma,m_i} = \{W_a : a \in W_{e_i}\}$.

  (iii) For all $A \in \mathscr{P}(\mathbb{N})$ and $m \in \mathbb{N}$, we have $m \in \Lambda_{k,e}(A)$ if and only if either:

    (a) $m \in \Gamma_d(A)$; or

    (b) for some i, we have $m = m_i$ and $(\exists a \in W_{e_i})(W_a \subseteq A)$.

Example 4.1. One example of this type of operator comes from the attempts described in Cenzer *et al.* (2005) to extend logic programming for reasoning about infinite sets. they defined an extension of logic programming, which they call *extended set-based programming* (esb). In this example, we shall give the formal definitions of ESB constraints, clauses and programs, and define the analogue of Horn programs and stable models for ESB programs. The basic idea is to incorporate constraints involving infinite sets into logic programming clauses by using various types of indexing schemes.

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

To describe the constraints used by Cenzer, Marek and Remmel, we first need to describe three types of indices for subsets of the natural numbers:

1 **Explicit indices of finite sets**. Recall that $D_n = \{x_1 < \ldots < x_k\}$ where $n = \sum_{i=1^k} 2^{x_i}$.
2 **Computable indices of computable sets**. Let $\phi_0, \phi_1, \ldots$ be an effective list of all partial computable functions. By a computable index of a computable set R, we mean an e such that ϕ_e is the characteristic function of R. If ϕ_e is a total $\{0,1\}$-valued function, we will use R_e to denote the set $\{x \in \mathbb{N} : \phi_e(x) = 1\}$.
3 **C.e. indices of c.e. sets**. By a c.e. index of a c.e. set W, we mean an e such that W equals the domain of ϕ_e, that is, $W_e = \{x \in \mathbb{N} : \phi_e(x) \text{ converges}\}$.

No matter what type of indices we use, we shall always consider two types of constraints based on X and a finite set of indices $\mathscr{F}$, namely, $\langle X, \mathscr{F}\rangle^=$ and $\langle X, \mathscr{F}\rangle^\subseteq$. For any subset $M \subseteq \omega$, we say that M *is a model of* $\langle X, \mathscr{F}\rangle^=$, written $M \models \langle X, \mathscr{F}\rangle^=$, if there exists an $e \in \mathscr{F}$ such that $M \cap X$ equals the set with index e. Similarly, we say that M *is a model of* $\langle X, \mathscr{F}\rangle^\subseteq$, written $M \models \langle X, \mathscr{F}\rangle^\subseteq$, if there exists an $e \in \mathscr{F}$ such that $M \cap X$ contains the set with index e.

Cenzer, Marek and Remmel then consider three different types of constraints:

(A) **Finite constraints**. Here we assume that we are given an explicit index x of a finite set X and a finite family $\mathscr{F}$ of explicit indices of finite subsets of X. We identify the finite constraints $\langle X, \mathscr{F}\rangle^=$ and $\langle X, \mathscr{F}\rangle^\subseteq$ with their codes, $\langle 0, 0, x, n\rangle$ and $\langle 0, 1, x, n\rangle$, respectively, where $\mathscr{F} = D_n$. Here the first coordinate 0 says that the constraint is finite; the second coordinate is 0 or 1 depending on whether the constraint is $\langle X, \mathscr{F}\rangle^=$ or $\langle X, \mathscr{F}\rangle^\subseteq$; and the third and fourth coordinates are the codes for X and $\mathscr{F}$, respectively.

(B) **Computable constraints**. Here we assume that we are given a computable index x of a computable set X and a finite family $\mathscr{R}$ of computable indices of computable subsets of X. Again we identify the computable constraints $\langle X, \mathscr{R}\rangle^=$ and $\langle X, \mathscr{R}\rangle^\subseteq$ with their codes, $\langle 1, 0, x, n\rangle$ and $\langle 1, 1, x, n\rangle$, respectively, where $\mathscr{R} = D_n$. Here the first coordinate 1 says that the constraint is computable; the second coordinate is 0 or 1 depending on whether the constraint is $\langle X, \mathscr{R}\rangle^=$ or $\langle X, \mathscr{R}\rangle^\subseteq$; and the third and fourth coordinates are the codes for X and $\mathscr{R}$, respectively.

(C) **C.e. constraints**. Here we are given a c.e. index x of a c.e. set X and a finite family $\mathscr{W}$ of c.e. indices of c.e. subsets of X. Again we identify the finite constraints $\langle X, \mathscr{W}\rangle^=$ and $\langle X, \mathscr{W}\rangle^\subseteq$ with their codes, $\langle 2, 0, x, n\rangle$ and $\langle 2, 1, x, n\rangle$, respectively, where $\mathscr{W} = D_n$. The first coordinate 2 says that the constraint is c.e.; the second coordinate is 0 or 1 depending on whether the constraint is $\langle X, \mathscr{W}\rangle^=$ or $\langle X, \mathscr{W}\rangle^\subseteq$; and the third and fourth coordinates are the codes for X and $\mathscr{W}$.

An *extended set-based clause* is defined to be a clause of the form

$$\langle X, \mathscr{A}\rangle^* \leftarrow \langle Y_1, \mathscr{B}_1\rangle^\subseteq, \ldots, \langle Y_k, \mathscr{B}_k\rangle^\subseteq, \langle Z_1, \mathscr{C}_1\rangle^=, \ldots, \langle Z_l, \mathscr{C}_l\rangle^=, \tag{5}$$

where $*$ is either $=$ or $\subseteq$. We refer to $\langle X, \mathscr{A}\rangle^*$ as the head of C, written head(C), and $\langle Y_1, \mathscr{B}_1\rangle^\subseteq, \ldots, \langle Y_k, \mathscr{B}_k\rangle^\subseteq, \langle Z_1, \mathscr{C}_1\rangle^=, \ldots, \langle Z_l, \mathscr{C}_l\rangle^=$ as the body of C, written body(C). Here, either k or l may be 0. M is said to be a model of C if either M does not model every constraint in body(C) or $M \models$ head(C).

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

Again, we consider three different types of clauses:

(a) **Finite clauses**. These are clauses in which all of the constraints are finite constraints.
(b) **Computable clauses**. These are clauses where all the constraints appearing in the clause are finite or computable constraints and at least one constraint is a computable constraint.
(c) **C.e. clauses**: These are clauses where all the constraints appearing in the clause are finite, computable or c.e. constraints and there is at least one c.e. constraint.

An extended set-based (ESB) program P is a set of clauses of the form of (1). We say that an ESB program P is computable if the set of codes of the clauses of P is a computable set. Here the code of a clause C of the form of (1) is $\langle c, e_1, \ldots, e_k, f_1, \ldots, f_l \rangle$ where c is the code of $\langle X, \mathscr{A} \rangle^*$, e_i is the code for $\langle Y_i, \mathscr{B}_i \rangle^{\subseteq}$ for $i = 1, \ldots, k$ and f_j is the code for $\langle Z_j, \mathscr{C}_j \rangle^{=}$ for $j = 1, \ldots, l$.

Given a program P, we use Fin(P) (respectively, Comp(P), CE(P)) to denote the set of all finite (respectively, computable, c.e.) clauses in P. It is easy to see from our coding of clauses that if P is a computable ESB program, then Fin(P), Comp(P) and CE(P) are also computable ESB programs.

Let P be a computable *ESB* program. We say that P is *computable with finite constraints* if $P = \text{Fin}(P)$. Similarly, we say that P is *computable with computable constraints* if $P = \text{Fin}(P) \cup \text{Comp}(P)$ and $\text{Comp}(P) \neq \emptyset$, and P is *computable with c.e. constraints* if $\text{CE}(P) \neq \emptyset$. Finally, we say that P is *weakly finite with computable constraints* if P is computable with computable constraints and the set of heads of clauses in Comp(P) is finite, and P is *weakly finite with c.e. constraints* if P is computable with c.e. constraints and the set of heads of clauses in $\text{Comp}(P) \cup \text{CE}(P)$ is finite.

Next we define the analogue of Horn programs for ESB programs. A Horn program P is a set of clauses of the form

$$\langle X, \mathscr{A} \rangle^{\subseteq} \leftarrow \langle Y_1, \mathscr{B}_1 \rangle^{\subseteq}, \ldots, \langle Y_k, \mathscr{B}_k \rangle^{\subseteq} \tag{6}$$

where $\mathscr{A}$ is a singleton. We define the *one-step provability operator*, $T_P : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$, so that for any $S \subseteq \mathbb{N}$, we have $T_P(S)$ is the union of the set of all F_e such that there exists a clause $C \in P$ with $S \models \text{body}(C)$, $\text{head}(C) = \langle X, \mathscr{A} \rangle^{\subseteq}$ and $A = \{e\}$ where $F_e = D_e$ if head(C) is a finite constraint, $F_e = R_e$ if head(C) is a computable constraint, and F_e is W_e if head(C) is a c.e. constraint. It is easy to see that T_P is a monotone operator, and hence there is a least fixed point, which we denote by M^P. Moreover, it is easy to check that M^P is a model of P.

If P is an ESB Horn program in which the body of every clause consists of *finite* constraints, then one can easily show that the least fixed point of T_P is reached in ω-steps, that is, $M^P = T_P \uparrow^{\omega} (\emptyset)$. However, if we allow clauses whose bodies contain either computable or c.e. constraints, we can no longer guarantee that we reach the least fixed point of T_P in ω steps. Here is an example.

Example 4.2. Let e_n be the explicit index of the set $\{n\}$ for all $n \geqslant 0$, let w be a computable index of $\mathbb{N}$ and f be a computable index of the set of even numbers E. Consider the

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

following program:

$$\begin{aligned}\langle\{0\},\{e_0\}\rangle^{\subseteq} &\leftarrow \\ \langle\{2x+2\},\{e_{2x+2}\}\rangle^{\subseteq} &\leftarrow \langle\{2x\},\{e_{2x}\}\rangle^{\subseteq} \quad \text{(for every number } x\text{)} \\ \langle\omega,\{w\}\rangle^{\subseteq} &\leftarrow \langle E,\{f\}\rangle^{\subseteq}.\end{aligned}$$

Clearly, $\mathbb{N}$ is the least model of P, but it takes $\omega+1$ steps to reach the fixed point. That is, it is easy to check that $T_P\uparrow^{\omega}= E$ and that $T_P\uparrow^{\omega+1}=\mathbb{N}$

Several results for ESB and weakly ESB programs were proved in Cenzer *et al.* (2005). Their basic result for ESB Horn programs is given by the following theorem.

Theorem 4.1.

(a) If P is a computable ESB Horn Program with finite constraints, then the least fixed point of the one-step provability operator T_P is c.e..

(b) If P is a weakly finite ESB Horn program with computable constraints such that $\mathrm{Fin}(P)$ is computable, then the least fixed point of the one step provability operator T_P is c.e..

(c) If P is a weakly finite ESB Horn program with c.e. constraints such that $\mathrm{Fin}(P)$ is computable, then the least fixed point of the one-step provability operator T_P is c.e..

In fact, a similar result to Theorem 4.1 holds for k-weakly Σ^0_1 monotone operators.

Theorem 4.2. Let Λ be a k-weakly Σ^0_1 monotone operator with index $\langle k,e\rangle=\langle k,\langle d,\langle m_1,e_1,\dots,m_k,e_k\rangle\rangle$. Then:

(a) $|\Lambda|\leqslant\omega\cdot(k+1)$.

(b) $\mathrm{lfp}(\Lambda)$ is Σ^0_1.

Proof. We will present an informal procedure that constructs the closure in $\leqslant k+1$ rounds where each round may consist of as many as ω steps.

Round (1). First let $U_0=\mathrm{lfp}(\Gamma_d)$. Since Γ_d is a Σ^0_1 monotone inductive operator, U_0 is c.e. by Theorem 2.1. Next consider the finite set

$$F_0=\{m_i:(\exists a\in W_{e_i})(W_a\subseteq U_0)\}.$$

We cannot necessarily find F_0 effectively, but, nevertheless, F_0 is a finite set, so $A_1=U_0\cup F_0$ will be a c.e. set. If $F_0=\emptyset$, we have $\mathrm{lfp}(\Lambda)=U_0$ and $|\Lambda|\leqslant\omega$. Otherwise, go on to Round 2.

We now present the description of Round $n+1$, for $n\geqslant 1$, assuming that A_n is the result of step n.

Round $(n+1)$**.** Consider the set $U_n=\Gamma_d^{\omega}(A_n)$. It is easy to see that since A_n is c.e., U_n is also c.e.. Next consider the finite set

$$F_n=\{m_i:(\exists a\in W_{e_i})(W_a\subseteq U_n)\}.$$

Again, we cannot necessarily find F_n effectively, but, nevertheless, $A_{n+1}=U_n\cup F_n$ is a c.e. set. Now, if $F_n\subseteq U_n$, we have $\mathrm{lfp}(\Lambda)=U_n$ and $|\Lambda|\leqslant\omega\cdot(n+1)$. Otherwise, go on to Round $(n+2)$.

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

It is clear that this process must be completed after at most $k+1$ rounds, so $|\Lambda| \leqslant \omega \cdot (k+1)$ and $\mathrm{lfp}(\Lambda)$ is always a c.e. set. □

Example 4.3. It is easy to construct an example Λ of a k-weakly finitary Σ_1^0 monotone operator with $|\Lambda| = \omega \cdot (k+1)$, as follows. Let $A_0, \ldots, A_k$ be a set of infinite computable sets that partition $\mathbb{N}$. Let $A_i = \{a_{0,i} < a_{1,i} < \ldots\}$ for $i = 0, \ldots, k$. Now define a Σ_1^0 monotone operator Γ such that for all $A \subseteq \mathbb{N}$:

(i) $a_{0,0} \in \Gamma(A)$.
(ii) For all $j \geqslant 0$. $a_{j+1,0} \in \Gamma(A)$ if and only if $a_{j,0} \in A$.
(iii) For all $i \geqslant 1$, $a_{1,i} \in \Gamma(A)$ if and only if $a_{0,i} \in A$.
(iv) For all $i \geqslant 1$ and $j \geqslant 1$, $a_{j+1,i} \in \Gamma(A)$ if and only if $a_{j,i} \in A$.

Finally, we complete the definition of Λ by adding the following rules, which govern when the elements $a_{0,1}, \ldots, a_{0,k}$ can be in $\Lambda(A)$.

For all $i > 0$, $a_{0,i} \in \Lambda(A)$ if and only if $A_{i-1} \subseteq A$.

It is easy to see that Λ is a k-weakly finitary Σ_1^0 monotone operator and that

$$\begin{aligned} \Lambda^{\omega} &= A_0, \\ \Lambda^{\omega+1} &= A_0 \cup \{a_{0,1}\}, \\ \Lambda^{2\omega} &= A_0 \cup A_1, \\ \Lambda^{2\omega+1} &= A_0 \cup A_1 \cup \{a_{0,2}\}, \\ &\vdots \\ \Lambda^{k\omega} &= A_0 \cup A_1 \cup \cdots \cup A_{k-1}, \\ \Lambda^{k\omega+1} &= A_0 \cup A_1 \cup \ldots \cup A_{k-1} \cup \{a_{0,k}\}, \text{ and} \\ \Lambda^{(k+1)\omega} &= A_0 \cup A_1 \cup \cdots \cup A_k = \mathbb{N}. \end{aligned}$$

Thus $|\Lambda| = \omega(k+1)$.

The following lemma gives an alternate approach to proving part (b) of Theorem 4.2, and will be needed below.

Lemma 4.1. Let Λ be a k-weakly finitary Σ_1^0 monotone operator with index

$$\langle k, e \rangle = \langle k, \langle d, \langle m_1, e_1, \ldots, m_k, e_k \rangle \rangle \rangle.$$

Then

(a) for some finite subset F of $\{m_1, \ldots, m_k\}$, $\mathrm{lfp}(\Lambda) = \Gamma_d^{\omega}(F)$, and
(b) for some finite subset G of $\{m_1, \ldots, m_k\}$, $\Lambda^{\omega} = \Gamma_d^{\omega}(G)$.

Proof.

(a) Let $F = \{m_i : m_i \in \mathrm{lfp}(\Lambda)\}$. Then, certainly, $\Gamma_d^{\omega}(F) \subseteq \Lambda^{\omega}(F) \subseteq \mathrm{lfp}(\Lambda)$. For the reverse inclusion, it suffices to show that $C = \Gamma_d^{\omega}(F)$ is closed under Λ. If $\Lambda(C) - C \neq \emptyset$, then either:

(i) there is some $y \notin S_{\Gamma} = \{m_1, \ldots, m_k\}$ such that $y \in \Gamma_d(C) - C$; or

(ii) there is some $m_i \notin C$ such $W_a \subseteq C$ for some $a \in W_{e_i}$.

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

Note that (i) is impossible. That is, $\Gamma_d(C) \subseteq C$ because Γ_d is a Σ^0_1 monotone operator, and thus $\Gamma_d(\Gamma^\omega(F)) = \Gamma^\omega(F)$. But (ii) is impossible also since otherwise $m_i \in F$ and $F \subseteq C$. Thus it must be the case that $\Lambda(C) = \Gamma_d(C)$.

(b) Let $G = \{m_i : m_i \in \Lambda^\omega\}$. Since G is a finite set, there is some finite t such that $G \subseteq \Lambda^t$. Then, certainly, $\Gamma^\omega_d(G) \subseteq \Lambda^\omega(G) \subseteq \Lambda^\omega(\Lambda^t) = \Lambda^\omega$. For the reverse inclusion, suppose $D = \Gamma^\omega_d(G)$ and $\Lambda^\omega - D \neq \varnothing$. Then let s be the least stage such that there is an $x \in \Lambda^s - D$. Then either:

(I) $x \notin S_\Gamma = \{m_1, \ldots, m_k\}$ and hence, there is some finite set $F \subseteq \Lambda^{t-1}$ such that $x \in \Gamma(F)$; or

(II) $x = m_i \notin G$ and $W_a \subseteq \Lambda^{t-1}$ for some $a \in W_{e_i}$.

Note that in case (I), $F \subseteq D$ by our choice of s. But since F is finite, there must be some finite t such $F \subseteq \Gamma^t(G)$, so $x \in \Gamma(F) \subseteq \Gamma(\Gamma^t(G)) \subseteq \Gamma^\omega(G) = D$. Thus case (I) cannot hold. But Case (II) is impossible since otherwise $m_i \in G$ and $G \subseteq D$. Thus it must be the case that $\Lambda^\omega = \Gamma^\omega_d(G)$. □

It is possible to develop a theory of index sets for weakly finitary Σ^0_1 inductive operators. In general, this theory is more subtle than the corresponding theory of Σ^0_1 inductive operators. We will not attempt in this paper to prove analogues of all the index set results given in Section 3. Instead, we will give a couple of examples of index set results for weakly finitary Σ^0_1 inductive operators where there is a contrast between the index set result for weakly finitary Σ^0_1 inductive operators and the corresponding index set result for Σ^0_1 inductive operators.

Clearly, $\{e : |\Gamma_e| \leqslant \omega\} = \mathbb{N}$ and is thus computable since for any Σ^0_1 inductive operator Γ, we have $\Gamma^\omega = \mathrm{lfp}(\Gamma)$. By contrast, we have the following theorem for weakly finitary Σ^0_1 inductive operators.

Theorem 4.3.

(a) For all $k \geqslant 1$, the set of e such that $\langle k, e\rangle = \langle k, \langle d, \langle m_1, e_1, \ldots, m_k, e_k\rangle\rangle\rangle$ and $\{m_1, \ldots, m_k\} \cap \mathrm{cl}(\Lambda_{k,e}) = \varnothing$ (in which case $\mathrm{cl}(\Lambda_{k,e}) = \Gamma^\omega_d$) is a complete Π^0_3 set.

(b) For all $k \geqslant 1$, $\{e : |\Lambda_{k,e}| \leqslant \omega \ \&\ \{m_1, \ldots, m_k\} \subseteq \Lambda^\omega_{k,e}\}$ is Σ^0_3 complete.

(c) For all $k \geqslant 2$, $\{e : |\Lambda_{k,e}| \leqslant \omega\}$ is D^0_3 complete.

Proof. For the upper bound for part (a), suppose $\langle k, e\rangle = \langle k, \langle d, \langle m_1, e_1, \ldots, m_k, e_k\rangle\rangle\rangle$. Then it is easy to see from our construction in Theorem 4.2 that $\{m_1, \ldots, m_k\} \cap \mathrm{cl}(\Lambda_{k,e}) = \varnothing$ only if there is no i and $a \in W_{e_i}$ such that $W_a \subseteq \Gamma^\omega_d$. Since Γ_d is a Σ^0_1 inductive operator, Γ^ω_d is a c.e. set. Thus $\{m_1, \ldots, m_k\} \cap \mathrm{cl}(\Lambda_{k,e}) = \varnothing$ if and only if

$$(\forall i \in \{1, \ldots, k\})(\forall a \in W_{e_i})(\exists c)(c \in W_a \ \&\ c \notin \Gamma^\omega_d),$$

which is a Π^0_3 predicate.

Next we consider the upper bounds for parts (b) and (c). Fix a set $F \subseteq \{1, \ldots, k\}$. For each index $\langle k, e\rangle = \langle k, \langle d, \langle m_1, e_1, \ldots, m_k, e_k\rangle\rangle$, let $M_{F,k,e} = \Gamma^\omega_d(\{m_i : i \in F\})$.

Now fix $\langle k, e\rangle$. By Lemma 4.1, we know there there is some F such that $M_{F,k,e} = \Lambda^\omega_{k,e}$. We are interested in analysing the predicate that

$$Q(F, k, e) : \ M_{F,k,e} = \Lambda^\omega_{k,e}. \tag{7}$$

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

First suppose that $F, G \subseteq \{1, \ldots, k\}$ and $M_{F,k,e}, M_{G,k,e} \subseteq \Lambda^{\omega}_{k,e}$. Then it is easy to see that there must be some finite stage t such that $G \cup F \subseteq \Lambda^{t}_{k,e}$. But then

$$\begin{aligned} M_{F\cup G,k,e} &= \Gamma^{\omega}_{d}(G \cup F) \\ &\subseteq \Gamma^{\omega}_{d}(\Lambda^{t}_{k,e}) \\ &\subseteq \Lambda^{\omega}_{k,e}(\Lambda^{t}_{k,e}) \\ &= \Lambda^{\omega}_{k,e}. \end{aligned}$$

It thus follows that a particular F such that $M_{F,k,e} = \Lambda^{\omega}_{k,e}$ is the maximal G such that $M_{G,k,e} \subseteq \Lambda^{\omega}_{k,e}$.

Now if $M_{F,k,e} = \Lambda^{\omega}_{k,e}$, we can list the elements of F in the order in which they appear in the sequence $\{\Lambda^{t}_{k,e}\}_{t \geqslant 0}$. That is, there is listing of $F = \{f_1, \ldots, f_s\}$, $1 \leqslant i_1 < \ldots i_p < s$ and $t_1 < t_2 < \cdots < t_{p+1}$ such that

$$\begin{aligned} f_1, \ldots, f_{i_1} &\in \Lambda^{t_1}_{k,e} - \Lambda^{t_1 - 1}_{k,e}, \\ f_{i_1+1}, \ldots, f_{i_2} &\in \Lambda^{t_2}_{k,e} - \Lambda^{t_2 - 1}_{k,e}, \\ &\vdots \\ f_{i_{p-1}+1}, \ldots, f_{i_p} &\in \Lambda^{t_p}_{k,e} - \Lambda^{t_p - 1}_{k,e}, \text{ and} \\ f_{i_p+1}, \ldots, f_{s} &\in \Lambda^{t_{p+1}}_{k,e} - \Lambda^{t_{p+1} - 1}_{k,e}. \end{aligned}$$

But in such circumstances it is easy to see that

$$\begin{aligned} \Lambda^{t_1-1}_{k,e} &= \Gamma^{t_1-1}_{d} \\ \Lambda^{t_1}_{k,e} &= \Gamma_d(\Gamma^{t_1-1}_{d}) \cup \{f_1, \ldots, f_{i_1}\} = \Gamma^{t_1}_{d} \cup \{f_1, \ldots, f_{i_1}\}. \end{aligned}$$

Now we can effectively find an index q_1 such that $W_{q_1} = \Gamma^{t_1}_{d} \cup \{f_1, \ldots, f_{i_1}\} = \Lambda^{t_1}_{k,e}$ from t_1 and $f_1, \ldots, f_{i_1}$. This gives

$$\begin{aligned} \Lambda^{t_2-1}_{k,e} &= \Gamma^{t_2-1-t_1}_{d}(W_{q_1}) \text{ and} \\ \Lambda^{t_2}_{k,e} &= \Gamma_d(\Gamma^{t_2-1-t_1}_{d}(W_{q_1})) \cup \{f_{i_1+1}, \ldots, f_{i_2}\} \\ &= \Gamma^{t_2-t_1}_{d}(W_{q_1}) \cup \{f_{i_1+1}, \ldots, f_{i_2}\}. \end{aligned}$$

Now we can effectively find an index q_2 such that $W_{q_2} = \Gamma^{t_2-t_1}_{d}(W_{q_1}) \cup \{f_{i_1+1}, \ldots, f_{i_2}\} = \Lambda^{t_2}_{k,e}$ from q_1, t_2, and $f_{i_1+1}, \ldots, f_{i_2}$. Continuing in this way, if we have found an index q_r such that $W_{q_{r-1}} = \Lambda^{t_{r-1}}_{k,e}$, then

$$\begin{aligned} \Lambda^{t_r-1}_{k,e} &= \Gamma^{t_r-1-t_{r-1}}_{d}(W_{q_{r-1}}) \text{ and} \\ \Lambda^{t_r}_{k,e} &= \Gamma_d(\Gamma^{t_r-1-t_{r-1}}_{d}(W_{q_{r-1}})) \cup \{f_{i_{r-1}+1}, \ldots, f_{i_r}\} \\ &= \Gamma^{t_r-t_{r-1}}_{d}(W_{q_{r-1}}) \cup \{f_{i_{r-1}+1}, \ldots, f_{i_r}\}. \end{aligned}$$

Again, we can effectively find an index q_r such that

$$W_{q_r} = \Gamma^{t_r-t_{r-1}}_{d}(W_{q_{r-1}}) \cup \{f_{i_{r-1}+1}, \ldots, f_{i_r}\} = \Lambda^{t_r}_{k,e}$$

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

from q_{r-1}, t_r, and $f_{i_{r-1}+1},\ldots,f_{i_r}$. Finally, to verify that each stage works properly, we must check for each r that

$$\{f_{i_{r-1}+1},\ldots,f_{i_r}\} \subseteq \Lambda_{k,e}(\Gamma_d^{t_r-1-t_{r-1}}(W_{q_{r-1}}))$$

or that for each $m_j \in \{i_{r-1}+1,\ldots,i_r\}$, we have

$$(\exists a)(a \in W_{e_i} \ \&\ W_a \subseteq \Gamma_d^{t_r-1-t_{r-1}}(W_{q_{r-1}}).$$

Again, we can effectively find an index v_r for $\Gamma_d^{t_r-1-t_{r-1}}(W_{q_{r-1}})$ so that the predicate that $W_a \subseteq W_{v_r} = \Gamma_d^{t_r-1-t_{r-1}}(W_{q_{r-1}})$ is a Π_2^0 predicate. It follows that for each $m_j \in \{i_{r-1}+1,\ldots,i_r\}$,

$$(\exists a)(a \in W_{e_i} \ \&\ W_a \subseteq \Gamma_d^{t_r-1-t_{r-1}}(W_{q_{r-1}}))$$

is a Σ_3^0 predicate. Thus the existence of sequences $F = \{f_1,\ldots,f_s\}$, $1 \leqslant i_1 < \ldots i_p < s$, $t_1 < t_2 \leqslant t_{p+1}$, $q_1,\ldots,q_{p+1}$ satisfying all the properties above is a Σ_3^0 predicate. It then follows that $M_{G,k,e} \subseteq \Lambda_{k,e}^{\omega}$ is a Σ_3^0 predicate, since it is equivalent to saying that there exists an $F \subseteq \{1,\ldots,k\}$ such that $G \subseteq F$ and there exist sequences $F = \{f_1,\ldots,f_s\}$, $1 \leqslant i_1 < \ldots i_p < s$, $t_1 < t_2 \leqslant t_{p+1}$, $q_1,\ldots,q_{p+1}$ satisfying all the properties above. Thus the predicate that $M_{G,k,e} \nsubseteq \Lambda_{k,e}^{\omega}$ is Π_3^0. Now, for any $F \neq \{1,\ldots,k\}$, the predicate that F is the maximal G such that $M_{G,k,e} \subseteq \Lambda_{k,e}^{\omega}$ is the conjunction of Σ_3^0 and Π_3^0 predicates. If $F = \{1,\ldots,k\}$, the predicate that F is the maximal G such that $M_{G,k,e} \subseteq \Lambda_{k,e}^{\omega}$ is just a Σ_3^0 predicate. Note that if $\{m_1,\ldots,m_k\} \subseteq \Lambda^{\omega}$, it must be the case that $\Lambda_{k,e}^{\omega} = M_{\{1,\ldots,k\},k,e}$ Finally, to say that $|\Lambda_{k,e}| > \omega$, we need only say that there exists an $F \neq \{1,\ldots,k\}$ such that F is the maximal G such that $M_{G,k,e} \subseteq \Lambda_{k,e}^{\omega}$ and $M_{F,k,e}$ is not closed under $\Lambda_{k,e}$. Now if $M_{F,k,e} = \Lambda_{k,e}^{\omega}$, then, clearly, $M_{F,k,e}$ is closed under Γ_d, so $M_{F,k,e}$ is not closed under $\Lambda_{k,e}$ if and only if

$$(\exists m_i \notin F)(\exists a \in W_{e_i})[W_a \subseteq M_{F,k,e}],$$

which is a Σ_3^0 predicate. Thus the predicate $|\Lambda_{k,e}| > \omega$ is a conjunction of Σ_3^0 and Π_3^0 predicates. Thus we have established the upper bounds for parts (b) and (c).

For the completeness of parts (a),(b) and (c), we will use the Σ_3^0 complete set Cof $= \{e : W_e \text{ is cofinite}\}$. Let $P = \{p_0 < p_1 < \cdots\}$ denote the set of primes.

For completeness for part (a), fix k and let $W_{f_i} = \{2^n p_m : n \geqslant 0 \ \&\ m \geqslant i\}$ for $i \geqslant 0$. Then define a 1-1 computable function g so that $\langle k, g(e)\rangle = \langle k, \langle d, \langle m_1, e_1, \ldots, m_k, e_k\rangle\rangle\rangle$ where $m_i = i - 1$ and $W_{e_i} = \{f_0, f_1, \ldots\}$, for $i = 1,\ldots,k$, and Γ_d is defined so that for all $A \subseteq N$:

(1) for all $m \geqslant k$, $p_m \in \Gamma_d(A) \iff m \in W_e$; and
(2) for all $n \geqslant 1$ and $m \geqslant k$, $2^n p_m \in \Gamma_d(A) \iff 2^{n-1} p_m \in A$.

It is then easy to see that $\Gamma_d^1 = \{p_m : m \in W_e \ \&\ m \geqslant k\}$, $\Gamma_d^{\omega} = \{2^n p_m : m \in W_e \ \&\ m \geqslant k \ \&\ n \geqslant 0\}$, and there is no finite t such that $W_{f_i} \subseteq \Gamma_d^t$ for some i. Thus, if W_e is cofinite, there will be an i such $W_{f_i} \subseteq \Gamma_d^{\omega}$ and, hence, $\{0,\ldots,k-1\} \subseteq \Gamma_d^{\omega+1} - \Gamma_d^{\omega}$. However, if W_e is not cofinite, there will be no i such that $W_{f_i} \subseteq \Gamma_d^{\omega}$. Hence $\Gamma_d^{\omega} = \mathrm{cl}(\Lambda_{k,g(e)})$ and $\{0,\ldots,k-1\} \cap \mathrm{cl}(\Lambda_{k,g(e)}) = \varnothing$. Thus

$$g(e) \in \{e : \langle k, e\rangle = \langle k, \langle d, \langle m_1, e_1, \ldots, m_k, e_k\rangle\rangle\rangle \ \&\ \{m_1,\ldots,m_k\} \cap \mathrm{cl}(\Lambda_{k,e} = \varnothing\}$$

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

if and only if W_e is not cofinite. It follows that

$$\{e : \langle k,e\rangle = \langle k,\langle d,\langle m_1,e_1,\ldots,m_k,e_k\rangle\rangle\rangle \;\&\; \{m_1,\ldots,m_k\} \cap \mathrm{cl}(\Lambda_{k,e} = \emptyset\}$$

is Π^0_3 complete.

For the completness for part (b), fix k and for $i = 1,\ldots,k$ let $m_i = i-1$ and let $W_{e_i} = \{b_0,b_1,b_2,\ldots\}$ where for each n, $W_{b_n} = \mathbb{N}-\{0,\ldots,n\}$. Then define the 1:1 computable function f by

$$f(a) = \langle k,e\rangle = \langle k,\langle d,\langle m_1,e_1,\ldots,m_k,e_k\rangle\rangle$$

where Γ_d is defined by:

For all $A \subseteq \mathbb{N}$:

(1) $k \in \Gamma_d(A)$.

(2) For all $x \geqslant 1$, $x+k \in \Gamma_d(A) \iff x \in W_a \;\vee\; (\forall y < x) y+k \in A$.

Now, if W_a is cofinite, it is easy to see that $\Lambda^1_{k,e}$ is cofinite and hence $\{0,\ldots,k-1\} \subseteq \Lambda^2_{k,e}$. It then easily follows that $\Lambda^{\omega}_{k,e} = \mathbb{N}$ and hence $|\Lambda_{k,e}| \leqslant \omega$. However, if W_e is not cofinite, it is easy to see that there is no $t \geqslant 0$ such that $\Lambda^t_{k,e}$ is cofinite. However, it will be the case that $\Lambda^{\omega}_{k,e} \supseteq \{x : k \leqslant x\}$, so $\Lambda^{\omega+1}_{k,e} = \mathbb{N}$. Thus

$$a \in \mathrm{Cof} \iff f(a) \in \{e : |\Lambda_{k,e}| \leqslant \omega \;\&\; \{m_1,\ldots,m_k\} \subseteq \Lambda^{\omega}_{k,e}\}$$

so

$$\{e : |\Lambda_{k,e}| \leqslant \omega \;\&\; \{m_1,\ldots,m_k\} \subseteq \Lambda^{\omega}_{k,e}\}$$

is Σ^0_3 complete.

For the completeness of part (c), fix $k \geqslant 2$. Then we need only show that there is a 1:1 computable function h such that $h(a,b) \in \{e : |\Lambda_{k,e}| \leqslant \omega\}$ if and only if W_a is cofinite and W_b is not cofinite. Let $P = \{p_0 < p_1 < \ldots\}$ be the set of prime numbers. For each i, let $W_{c_i} = \{2^n p_i : n \geqslant 1\}$. Then let h be the computable function such that $\langle k,h(a,b)\rangle = \langle k,\langle d,\langle m_1,e_1,\ldots,m_k,e_k\rangle\rangle$ where $m_i = 2(i-1)+1$ for $i = 1,\ldots k$, $W_{e_1} = \{b_0,b_1,b_2,\ldots\}$ where $W_{b_i} = \{2x+1 : x \in \mathbb{N}\} - \{1,3,\ldots,2i+1\}$, $W_{e_j} = \{c_0,c_1,c_2,\ldots\}$ for $j = 2,\ldots,k$, where $W_{c_i} = \{2^n p_m : n \geqslant 0 \;\&\; m \geqslant i\}$, and Γ_d is defined so that for all $A \subseteq \mathbb{N}$:

(1) $2k+1 \in \Gamma_d(A)$.

(2) For all $x \geqslant 1$, $2(x+k)+1 \in \Gamma_d(A) \iff x \in W_a \;\vee\; (\forall y < x)(2(y+k)+1 \in A$.

(3) For all $m \geqslant 0$. $2p_m \in \Gamma_d(A) \iff m \in W_b$.

(4) For all $m \geqslant 0$ and $n \geqslant 2$, $2^n p_m \in \Gamma_d(A) \iff 2^{n-1} p_m \in A$.

We can use the same analysis as we used in part (a) to conclude $\{2p_m : m \in W_b\} \subseteq \Gamma^1_d$, $\{2^n p_m : m \in W_b \;\&\; n \geqslant 1\} \subseteq \Gamma^{\omega}_d$, and there is no finite t such that $W_{c_i} \subseteq \Gamma^t_d$ for some i. Moreover, $\{3,\ldots,2k-1\} \subseteq \Gamma^{\omega+1}_d - \Gamma^{\omega}_d$ if W_b is cofinite, and $\{3,\ldots,2k-1\} \cap \Gamma^{\omega}_d = \emptyset$ otherwise. Next we can use our analysis from part (b) to conclude that if W_a is cofinite, $1 \in \Lambda^1_{k,e}$, and hence $\{1\} \cup \{2s+1 : s \geqslant k\} \subseteq \Lambda^{\omega}_{k,e}$. However, if W_a is not cofinite, there is no stage t such that $1 \in \Lambda^t_{k,e}$, so $\{2s+1 : s \geqslant k\} \subseteq \Lambda^{\omega}_{k,e}$ and $1 \in \Lambda^{\omega+1}_{k,e}$. It follows that $|\Lambda_{k,h(a,b)}| \leqslant \omega$ if and only if W_a is cofinite and W_b is not cofinite. Hence, for $k \geqslant 2$, $\{e : |\Lambda_{k,e}| \leqslant \omega\}$ is D^0_3 complete. □

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

Next we need to define the family of difference sets of Σ^0_3 sets. For two Σ^0_3 sets A and B, the difference $A - B$ is the intersection of a Σ^0_3 set and a Π^0_3 set and is said to be a 2-Σ^0_3 set. For $n > 0$, we say that a set C is $2n$-Σ^0_3 if and only if A is the union of n 2-Σ^0_3 sets and is $2n+1$-Σ^0_3 if and only if A is the union of a Σ^0_3 set with a $2n$-Σ^0_3 set. We say that A is an n-Π^0_3 set if the complement of A is an n-Σ^0_3 set.

We can then prove the following theorem.

Theorem 4.4. Fix any computable set R_t. Then for each k, we have

$$\{e : \mathrm{lfp}(\Lambda_{k,e}) \cap R_t \text{ is computable}\}$$

is a $(2^{k+1}-1)$-Σ^0_3 set.

Proof. Fix a set $F \subseteq \{1,\ldots,k\}$. Let $M_{F,k,e} = \Gamma^\omega_d(\{m_i : i \in F\})$ for each index $\langle k,e\rangle = \langle k, \langle d, \langle m_1, e_1, \ldots, m_k, e_k\rangle\rangle$. We are interested in analysing the predicate that

$$P(F,k,e): \; M_{F,k,e} = \mathrm{lfp}(\Lambda_{k,e}) \;\&\; R_t \cap M_{F,k,e} \text{ is computable.} \qquad (8)$$

It follows from Lemma 4.1 that $\mathrm{lfp}(\Lambda_{k,e}) = M_{F,k,e}$ if and only if:

1 $\{m_i : (\exists a \in W_{e_i})(W_a \subseteq M_{F,k,e})\} \subseteq \{m_i : i \in F\}$; and
2 for all $G \subsetneq F$, $\{m_i : (\exists a \in W_{e_i})(W_a \subseteq M_{G,k,e})\} \not\subseteq \{m_i : i \in G\}$.

The predicate that $\{m_i : (\exists a \in W_{e_i})(W_a \subseteq M_{G,k,e})\} \not\subseteq \{m_i : i \in G\}$ is Σ^0_3 since it holds if and only if there is an $i \in \{1,\ldots,k\} - G$ such that $(\exists a)(a \in W_{e_i} \;\&\; W_a \subseteq M_{G,k,e})$. Since $M_{G,k,e}$ is uniformly c.e., the predicate $W_a \subseteq M_{G,k,e}$ is Π^0_2, and hence the predicate $(\exists a)(a \in W_{e_i} \;\&\; W_a \subseteq M_{G,k,e})$ is Σ^0_3. It follows that the predicate $\{m_i : (\exists a \in W_{e_i})(W_a \subseteq M_{F,k,e})\} \subseteq \{m_i : i \in F\}$ is Π^0_3 if $F \neq \{1,\ldots,k\}$. Finally, the predicate '$M_{F,k,e} \cap R_t$ is computable' is Σ^0_3. Thus, if $F \neq \{1,\ldots,k\}$, the predicate $P(F,k,e)$ is the conjunction of a Σ^0_3 and Π^0_3 predicate and hence is a 2-Σ^0_3 predicate. If $F = \{1,\ldots,k\}$, we may omit the Π^0_3 predicate so that $P(F,k,e)$ is a Σ^0_3 predicate.

It follows that the predicate that $\{e : \mathrm{lfp}(\Gamma_{k,e}) \cap R_t$ is computable$\}$ is a disjunction of $2^k - 1$ 2-Σ^0_3 sets and one Σ^0_3 set and hence a $2^{k+1}-1$ set. □

It is important to note that the set of all $\langle k,e\rangle$ such that $\mathrm{lfp}(\Lambda_{k,e})$ *itself* is computable is just Σ^0_3. (In fact, if the set R_t in Theorem 4.4 is finite or cofinite, then $\{e : \mathrm{lfp}(\Lambda_{k,e}) \cap R_t$ is computable$\}$ is Σ^0_3.) That is, for each finite $F \subseteq \{1,\ldots,k\}$ and each computable set R, the question of whether $R = M_{F,k,e}$ is a Π^0_2 question since $M_{F,k,e}$ is uniformly c.e.. If there is an F such that $R = M_{F,k,e}$, then the question of whether $\{m_i : (\exists a \in W_{e_i})(W_a \subseteq R)\} \subseteq \{m_i : i \in F\}$ is a Π^0_2 question. That is, the question whether $W_a \subseteq R$ is a Π^0_1 question, so the question of whether $(\exists i \in \{1,\ldots,k\} - F)(\exists a)(a \in W_{e_i} \;\&\; W_a \subseteq R)$ is a Σ^0_2 question. Thus $\mathrm{lfp}(\Lambda_{k,e})$ is computable if and only if there is an s and there exists an $F \subseteq \{1,\ldots,k\}$ such that W_s is computable, $M_{F,k,e} = W_s$, $\{m_i : (\exists a \in W_{e_i})(W_a \subseteq W_s)\} \subseteq \{m_i : i \in F\}$, and for all $G \subsetneq F$, $\{m_i : (\exists a \in W_{e_i})(W_a \subseteq M_{G,k,e})\} \not\subseteq \{m_i : i \in G\}$. Since the predicate W_s is computable, $M_{F,k,e} = W_s$ and $\{m_i : (\exists a \in W_{e_i})(W_a \subseteq W_s)\} \subseteq \{m_i : i \in F\}$ are all Π^0_2 and the predicates $\{m_i : (\exists a \in W_{e_i})(W_a \subseteq M_{G,k,e})\} \not\subseteq \{m_i : i \in G\}$ are Σ^0_3, we have the predicate that $\mathrm{lfp}(\Lambda_{k,e})$ is computable is Σ^0_3. We can then proceed as in the proof of Theorem 4.2 to prove $\{\langle k,e\rangle : \mathrm{lfp}(\Lambda_{k,e})$ is computable$\}$ is Σ^0_3-complete. Thus we have the following theorem.

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

Theorem 4.5. $\{\langle k, e\rangle : \text{lfp}(\Lambda_{k,e})$ is computable$\}$ is Σ^0_3-complete.

Finally, we give a completeness result for Theorem 4.4 for the case $k = 1$.

Theorem 4.6. Let R_t be a fixed infinite coinfinite computable set. Then

$$\{e : \text{lfp}(\Lambda_{1,e}) \cap R_t \text{ is computable}\}$$

is 3-Σ^0_3-complete.

Proof. The upper bound on the complexity is given by the proof of Theorem 4.4. For the other direction, fix $R_t = \{2n : n \in \mathbb{N}\}$ without loss of generality. Let $C = \{e : \text{lfp}(\Lambda_{1,e}) \cap R_t$ is computable$\}$. Note that it is proved in Soare (1987) that Rec $= \{e : W_e$ is computable$\}$ and Cof $= \{e : W_e$ is cofinite$\}$ are Σ^0_3 complete.

For the completeness, first we claim that

$$D = \{\langle a, b, c\rangle : (W_a \text{ is not cofinite } \& \ W_b \text{ is computable}) \ \vee \ W_c \text{ is computable}\}$$

is 3-Σ^0_3 complete. Let $S = (B - A) \cup C$, where A,B,C are Σ^0_3. Then there are functions f, g, h such that $a \in A \iff f(a) \in$ Cof, $b \in B \iff g(b) \in$ Rec, and $c \in C \iff h(c) \in$ Rec. Thus, $s = \langle a, b, c\rangle \in S$ iff $[(f(a) \notin$ Cof$)$ and $g(b) \in$ Rec$)$ or $h(c) \in$ Rec$]$ iff $\phi(s) = \langle f(a), g(b), h(c)\rangle \in D$. Thus it suffices to reduce D to C. So we will define a 1-weakly finitary Σ^0_1 monotone operator $\Lambda_{f(a,b,c)}$ such that lfp$(\Lambda_{f(a,b,c)}) \cap R_t$ is computable if and only if $\langle a, b, c\rangle \in D$. Since Rec and Cof are Σ^0_3 complete, it follows that there exists a computable function g such that W_c is computable or W_a is cofinite if and only if $W_{g(a,c)}$ is cofinite. Let h be a computable function such that for each n, we have $W_{h(n)} = \{8i + 3 : i > n\}$. The 1-weakly finitary inductive operator $\Lambda = \Lambda_{f(a,b,c)}$ is defined by the following clauses:

(1) $0 \in \Lambda(A)$ if $W_{h(n)} \subseteq A$ for some n.
(2) $8\langle i, s\rangle + 1 \in \Lambda(A)$ if $i \in W_{g(a,c),s}$ or $8\langle i, s + 1\rangle + 1 \in A$.
(3) $8i + 3 \in \Lambda(A)$ if $8\langle i, 0\rangle + 1 \in A$.
(4) $8\langle i, s\rangle + 5 \in \Lambda(A)$ if $i \in W_{b,s}$ or $8\langle i, s + 1\rangle + 5 \in A$.
(5) $8i + 2 \in \Lambda(A)$ if $8\langle i, 0\rangle + 5 \in A$.
(6) $8\langle i, s\rangle + 7 \in \Lambda(A)$ if $0 \in A$ and either $i \in W_{c,s}$ or $8\langle i, s + 1\rangle + 7 \in A$.
(7) $8i + 4 \in \Lambda(A)$ if $8\langle i, 0\rangle + 7 \in A$.
(8) $8i + 2 \in \Lambda(A)$ if $0 \in A$.

It is easy to see that clauses (2)–(8) define a computable monotone inductive operator, so Λ is a 1-weakly finitary Σ^0_1 operator with $S_\Lambda = \{0\}$.

Clauses of type (2) and (3) ensure that lfp(Λ) must include $\{8i + 3 : i \in W_{g(a,c)}\}$, and clauses of type (4) and (5) ensure that lfp(Λ) must include $\{8i + 2 : i \in W_b\}$.

Let $M =$ lfp(Λ). If $W_{g(a,c)}$ is cofinite, one of the clauses of type (1) will apply and then the clauses of type (6), (7) and (8) will ensure that $M \cap R_t$ equals $\{0\} \cup \{8i + 2 : i < \omega\} \cup \{8i + 4 : i \in W_c\}$, so $M \cap R_t$ will be computable if and only if W_c is computable. If $W_{g(a,c)}$ is not cofinite, $M \cap R_t$ will consist of $\{8i + 2 : i \in W_b\}$, so $M \cap R_t$ will be computable if and only if W_b is computable.

If $\langle a, b, c\rangle \in D$, there are two cases. First suppose that W_c is computable. Then $W_{g(a,c)}$ is cofinite, so $M \cap R_t$ is computable, as desired. Next suppose that W_c is not computable.

https://doi.org/10.1017/S0960129506005585 Published online by Cambridge University Press

Then we must have that W_a is not cofinite and W_b is computable. This means that $W_{g(a,c)}$ is not cofinite and $M \cap R_t$ is again computable.

If $\langle a, b, c \rangle \notin D$, then W_c is not computable and either W_a is cofinite or W_b is not computable. Again there are two cases. First suppose that W_a is cofinite. Then $W_{g(a,c)}$ is cofinite, so $M \cap R_t$ is not computable, as desired. If W_a is not cofinite, $W_{g(a,c)}$ is not cofinite and W_b is not computable. Thus again $M \cap R_t$ is not computable. □

We conjecture that a similar completeness result will hold for k-weakly Σ_1^0 operators. Finally, we note that k-weakly computable monotone operators may be defined, and corresponding versions of Theorems 4.4, 4.5 and 4.6 can be shown.

References

Baldwin, J. (1982) Recursion theory and abstract dependence. In: Metakides, G. (ed.) *Patras Logic Symposion*, North Holland 67–76.

Baldwin, J. (1984) First order theories of abstract dependence relations. *Ann. Pure and Applied Logic* **26** 215–243.

Cenzer, D. (1978) Non-generable formal languages. *Fundamenta Informaticae* **3** 95–104.

Cenzer, D., Marek, W. and Remmel, J. B. (2005) Logic programming with infinite sets. *Annals of Mathematics and Artificial Intelligence* **44** 309–339.

Cenzer, D. and Remmel, J. B. (1999) Index sets in computable analysis. *Theoretical Computer Science* **219** 111–150.

Dekker, J. C. E. (1971) Two notes on vector spaces with recursive operations. *Notre Dame Journal of Formal Logic* **12** 329–334.

Downey, R. G. (1983) Nowhere simplicity in matroids. *J. Austral. Math. Soc.* (Series A) **35** 28–45.

Downey, R. G. (1983) On a question of A. Retzlaff. *Z. Math. Logik Grund. Math.* **29** 379–384.

Downey, R. G. (1984) Some remarks on a theorem of Iraj Kalantari concerning convexity and recursion theory. *Z. Math. Logik Grund. Math.* **30** 295–302.

Downey, R. G. and Remmel, J. B. (1998) Effective Algebras and Closure Systems: Coding Properties. In: Ershov, Y., Goncharov, Y., Nerode, Y. and Remmel, J. B. (eds.) Handbook of Recursive Mathematics. Volume 2. *Studies in Logic and the Foundations of Mathematics* **139**, Elsevier 997–1040.

Gelfond, M. and Lifschitz, V. (1988) The stable semantics for logic programs. In: Kowalski, R. and Bowen, K. A. (eds.) *ICLP88* 1070–1080.

Ershov, Y., Goncharov, S., Nerode, A. and Remmel, J. B. (eds.) (1998a) Handbook of Recursive Mathematics, Volume 1: Recursive Model Theory. *Studies in Logic and the Foundations of Mathematics* **138**, Elsevier.

Ershov, Y., Goncharov, S., Nerode, A. and Remmel, J. B. (eds.) (1998a) Handbook of Recursive Mathematics, Volume 2: Recursive Algebra, Analysis and Combinatorics. *Studies in Logic and the Foundations of Mathematics* **139**, Elsevier.

Hinman, P. G. (1978) *Recursion-Theoretic Hierarchies*, Springer-Verlag.

Kalantari, I. (1981) Effective content of a theorem of M. H. Stone. In: Crossley, J. N. (ed.) *Aspects of effective algebra*, U.D.A. Book Company, Steels Greek, 128–146.

Nerode, A. and Remmel, J. B. (1982) Recursion theory on matroids. In: Metakides, G. (ed.) *Patras Logic Symposion*, North Holland 41–65.

Nerode, A. and Remmel, J. B. (1983) Recursion theory on matroids, II. In: Chong, C. T. and Wicks, M. J. (eds.) *Southeast Asian Conference on Logic*, North Holland 133–184.

Nerode, A. and Remmel, J. B. (1985) A survey of the lattices of r.e. substructures. In: Nerode, A. and Shore, R. A. (eds.) Recursion Theory. *Proceedings of the Symposium in Pure Mathematics* **42**, Amer. Math. Soc. 323–375.

Remmel, J. B. (1978) Recursively enumerable Boolean algebras. *Ann. Math. Logic* **14** 75–107.

Remmel, J. B. (1980) Recursion theory on algebraic structures with an independent set. *Ann. Math. Logic* **18** 153–191.

Soare, R. E. (1987) *Recursively Enumerable Sets and Degrees*, Springer-Verlag.