Categoricity and Scott Families

Bakhadyr Khoussainov
Cornell University, Ithaca NY 14850 USA

Richard A. Shore *
Cornell University, Ithaca NY 14853 USA

1 Introduction

Effective model theory is an area of logic that analyzes the effective content of the typical notions and results of model theory and universal algebra. Typical notions in model theory and universal algebra are languages and structures, theories and models, models and their submodels, automorphisms and isomorphisms, embeddings and elementary embeddings. In this paper languages, structures, and models are assumed to be countable.

There are many ways to introduce considerations of effectiveness into the area of model theory or universal algebra. Here we will briefly explain considerations of effectiveness for theories and their models on the one hand, and for just structures on the other hand.

Let us begin by considering effectiveness for theories and their models. From the model theoretic point of view, given a first order theory, one is interested in finding models for the theory with specific algebraic or model-theoretic properties. In this sense theories are the basic objects in model theory. A natural way of introducing effectiveness is, therefore, to begin by considering decidable theories, i.e. ones whose theorems form a decidable (i.e. computable or recursive) set.

*Partially supported by NSF Grant DMS-9503503 and ARO through MSI, Cornell University, DAAL-03-C-0027.

Given a decidable complete theory, one can effectively carry out a Henkin type construction and build a model of the theory. This procedure guarantees that the satisfaction predicate for the model constructed is decidable. Thus we are led to the following definition.

Definition 1.1 *A structure $\mathcal{A}$ is* **decidable** *if there is a computable enumeration a_i of the elements of $\mathcal{A}$ such that $Th(\mathcal{A}, \mathbf{a}_i)$ is decidable.*

Just as model theory investigates the class of models of a given theory, effective model theory is concerned with decidable models of decidable theories. There have been a significant number of results about decidable models of decidable theories. These results typically discuss questions related to finding decidable prime, saturated, homogeneous models; omitting or realizing types by decidable models; the number of decidable models for decidable theories, etc. We refer the reader to [8] and [15] for survey articles in this area.

If we begin to introduce consideration of effectiveness just for the structures themselves, then we are essentially in the realm of general effective mathematics. Considerations of effectiveness for structures have been extensively developed since the early 60's beginning with Frölich and Shepherdson [5], Rabin [17], and Malcev [13]. However, it is worth to noting that even in the early 60's the idea of considering effectiveness in structures was not new. In the 30's Kleene and Church considered effectiveness in well-ordered sets and invented recursive ordinals. In the early 70's Nerode and his collaborators in the U. S., as well as Ershov and his colleagues in Novosibirsk, developed the powerful idea of combining model-theoretic and algebraic constructions with priority arguments from computability theory. This approach embodies the technical core of many results in effective model theory.

More recently, there have been many papers devoted to investigating effectiveness in structures. For example, Cenzer, Nerode and Remmel [1] have been developing the theory of p-time structures. Khoussainov and Nerode have begun the development of the theory of automatic structures [11]. These theories are based, respectively, on computations which can be performed in p-time and by finite automata. We will not discuss these topics but turn instead to structures in which the basic functions and relations can be computed by Turing machines.

Definition 1.2 *A structure $\mathcal{A}$ for a language $\mathcal{L}$ is* **computable** *if its domain A is a computable subset of ω and its functions and relations are uniformly computable. A structure isomorphic to a computable structure is called* **computably presentable***, and any such isomorphism is called a* **computable presentation***.*

The requirements of computability are significantly weaker than those for decidability. However, the definition captures what one normally means by an effective structure or presentation in mathematical discourse.

Identification of isomorphic structures is typical in model theory and universal algebra or, indeed, generally in classical mathematics. A typical model-theoretic or algebraic problem about isomorphisms can often be stated as follows: Find some invariants such that any two structures from the class are isomorphic if only if they have the same invariants.

Introducing effectiveness considerations into the area, we would like to understand the relationship between classical invariants and effective invariants; in particular, between isomorphism types and effective isomorphism types. Thus, while model theory identifies isomorphic structures, effective model theory is concerned with computable isomorphisms and finding characterizations for structures which have the same computable isomorphism type. A fundamental concept is therefore that of computable isomorphism type.

Definition 1.3 *Two computable structures $\mathcal{A}$ and $\mathcal{B}$ are of the same* **computable isomorphism type** *if there is computable isomorphism taking $\mathcal{A}$ to $\mathcal{B}$. The* **dimension** *of a computable structure $\mathcal{A}$ is the number of its computable isomorphism types.*

To what extent computable isomorphism types can differ from classical ones can be seen from the following result of Goncharov:

Theorem 1.4 ([7]) *For each $n \leq \omega$ there is a computable structure with computable dimension n.*

There has been significant interest in understanding the nature of the structures of dimension 1. The basic model theoretic notion which motivated this interest is that of ω–categoricity. A theory T is ω–categorical if all countable models of T are isomorphic. A structure $\mathcal{A}$ is ω–categorical if its theory is ω–categorical. The analogous concept for effective model theory deals only with computable structures and isomorphisms:

Definition 1.5 *A structure $\mathcal{A}$ is* **computably categorical** *if any two computable structures isomorphic to $\mathcal{A}$ are computably isomorphic.*

The result of Nurtazin [16] is one of the first about the nature of computably categorical structures. His theorem characterizes structures whose decidable presentations form one computable isomorphism type.

Theorem 1.6 ([16]) *For a structure $\mathcal{A}$ the following two conditions are equivalent:*

1. Any two decidable presentations of $\mathcal{A}$ are computably isomorphic.

2. There exists a finite sequence $\bar{c}$ of constants from A such that $(\mathcal{A}, \bar{c})$ is the prime model of the theory $Th(\mathcal{A}, \bar{c})$ and the set of atoms of this theory is computable.

In the late 70's Goncharov [6] and Remmel [18] independently gave an algebraic characterization for Boolean Algebras and Linear Orderings to be computably categorical.

Theorem 1.7 ([6] [18]) *1. A Boolean Algebra is computably categorical if and only if has finitely many atoms.*

2. A linear ordering is computably categorical if and only if the number of pairs of adjacent elements is finite.

2 Scott Families

Interestingly, all the structures which have been shown to be computably categorical have one common property. They all have Scott families.

Definition 2.1 *A* **Scott family** *for a structure $\mathcal{A}$ is a computable sequence $\phi_0(\bar{a}, x_1, \ldots, x_{n_0}), \phi_1(\bar{a}, x_1, \ldots, x_{n_1}), \ldots$ of $\exists$-formulas satisfiable in $\mathcal{A}$, where $\bar{a}$ is a fixed tuple of elements from $\mathcal{A}$, such that every tuple in $\mathcal{A}$ satisfies one these formulas and any two tuples satisfying the same formula from the sequence can be sent to each other via an automorphism of $\mathcal{A}$.*

The basic idea behind this definition is the following. If a computable structure $\mathcal{A}$ has a Scott family and $\mathcal{B}$ is a computable structure isomorphic to $\mathcal{A}$, then we can effectively carry out a back and forth argument to construct a computable isomorphism from $\mathcal{A}$ to $\mathcal{B}$.

Theorem 2.2 *If a structure $\mathcal{A}$ has a Scott family, then the structure is computably categorical. Moreover, for any n-tuple $(d_1, \dots, d_n)$ the expanded structure $(\mathcal{A}, d_1, \dots, d_n)$ also has a Scott Family.*

Proof. Let $\phi_0(\bar{a}, x_1, \dots, x_{n_0}), \phi_1(\bar{a}, x_1, \dots, x_{n_1}), \dots$ be a Scott family for $\mathcal{A}$, where $\bar{a} = (a_0, \dots, a_{m-1})$. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be computable presentations of $\mathcal{A}$. We define a mapping $f : A_1 \to A_2$ by stages. We can assume that for each $j \in \{0, \dots, m-1\}$, a_j^i is the element in A_i corresponding to the constant a_j. At even stages we define images of elements from $\mathcal{A}_1$, at odd stages we define preimages of elements from $\mathcal{A}_2$.

Stage 0. Set $f_1 = \{(a_0^1, a_0^2), \dots, (a_{m-1}^1, a_{m-1}^2)\}$.

Stage 2k. We can suppose that the function f_{2k-1} has been defined. Assume that $f_{2k-1} = \{(a_0^1, a_0^2), \dots, (a_{m-1}^1, a_{m-1}^2), (b_1, c_1), \dots, (b_s, c_s)\}$ and that f_{2k-1} can be extended to an isomorphism of $\mathcal{A}_1$ to $\mathcal{A}_2$. Let b be the first number not in the domain of f_{2k-1}. Consider the tuple $(b_1, \dots, b_s, b)$. Find an i such that $\phi_i(\bar{a}, b_1, \dots, b_s, b)$ holds in $\mathcal{A}_1$. Hence $\exists x \phi_i(\bar{a}, c_1, \dots, c_s, x)$ holds in $\mathcal{A}_2$. Find the first $c \in A_2$ for which $\phi_i(\bar{a}, c_1, \dots, c_s, c)$ holds. Extend f_{2k-1} by letting $f_{2k} = f_{2k-1} \bigcup \{(b, c)\}$.

Stage 2k+1. We define f_{2k+1} similarly so as to put the least element of A_2 not yet in the range of f_{2k} into that of f_{2k+1}.

Finally, let $f = \bigcup_{i \in \omega} f_i$. Thus, f is a computable isomorphism.

For the second part of the theorem, we slightly change the original Scott family. Namely, set $\psi_i = \phi_i(\bar{a}, x_1, \dots, x_{n_i}) \& \exists y_1 \dots \exists y_n (\&_j (d_j = y_j)$. Then, one can easily check that the sequence $\psi_0, \psi_1, \dots$ is a Scott family for the expanded structure $(\mathcal{A}, d_1, \dots, d_n)$. The theorem is proved.

Corollary 2.3 *If a structure $\mathcal{A}$ has a Scott family, then any expansion of $\mathcal{A}$ by finitely many constants is computably categorical.* $\Box$

At this point we would like to make the following two observations about the effect of expanding computably categorical structures by finitely many constants. First, as we have mentioned, all the known examples of computably categorical structures have Scott families. Thus, it is natural to ask whether there exists a computably categorical structure without a Scott family. By Corollary 2.3, one possible way to build a such structure is to provide an example of a computably categorical structure whose expansion by finitely many constants is not computably categorical. Second, as we mentioned, the notion of ω–categoricity is a basic model-theoretic motivation in the investigation of computably categorical structures. It is an easy consequence of the Ryll-Nardzewski theorem that if a structure $\mathcal{A}$ is ω-categorical then so is $(\mathcal{A}, \bar{a})$, the structure expanded by finitely many constants. It is the analogous situation in effective model theory that we wish to consider.

Millar [14] proved that a small amount of decidability is enough to guarantee that categoricity is preserved under such expansions. Informally his theorem states that if a structure $\mathcal{A}$ is computably categorical and we can effectively solve systems of algebraic equations and inequations over this structure, then computable categoricity is preserved under expansions by a finite number of constants.

Theorem 2.4 ([14]) *If a structure $\mathcal{A}$ is computably categorical and its existential theory is decidable, then the expansion of $\mathcal{A}$ by finitely many constants is also computably categorical.*

Without the assumption of the decidability of the existential diagram, the question (known as Ash-Goncharov problem) has been open:

Does there exist a computably categorical structure whose expansion by a finite number of constants is not computably categorical?

An answer to this question has recently been found:

Theorem 2.5 ([2]) *For each natural number n, there exists a computably categorical structure $\mathcal{A}$ such that, for every $a \in A$, the expanded structure $(\mathcal{A}, a)$ has dimension n.*

An immediate consequence of Corollary 2.3 is now the following result:

Corollary 2.6 *There exists a computably categorical structure without a Scott family.*

However, based on Corollary 2.3, one could suggest that the reason the structure constructed in Theorem 2.5 does not have a Scott family is that the structure has an expansion by a finite number of constants which is not computably categorical. We construct a counterexample to this suggestion in the next section.

3 Scott Sequences for Families of Computably Enumerable Sets

Our basic result is the following theorem.

Theorem 3.1 *There exists a structure without a Scott family such that every expansion of the structure by a finite number of constants is computably categorical.*

The structure required to establish the theorem is constructed by coding certain (uniformly) computably enumerable families of sets of natural numbers.

Definition 3.2 *A family S of sets of natural numbers has a* **one-to-one computable enumeration** *if there is a bijection $f : \omega \to S$ such that $\{(i,x)|x \in f(i)\}$ is computably enumerable. We then call f a* **(computable) one-to-one enumeration** *of S.*

We wish to consider a preordering on the one-to-one computable enumerations of S that naturally induces an equivalence relation corresponding to computable isomorphism:

Definition 3.3 *A computable enumeration f of S is* **reducible to** *g, $f \leq g$, if there is a computable Φ such that $f = g\Phi$. If $f \leq g$ and $g \leq f$, then we say that f and g are* **equivalent***.*

Note that if f is a one-to-one enumeration of S and $f = g\Phi$, then Φ is a permutation of ω and so $f \leq g$. Thus the equivalence classes of one-to-one enumerations are minimal elements in the induced partial ordering. These are the enumerations that we need to consider to define the family that supplies the structure required for Theorem 3.1. Informally, computable categoricity corresponds to there being a single such equivalence class and dimension corresponds to the number of such classes.

Definition 3.4 *A computable sequence $D_0, D_1, \ldots$ of (canonical indices for) finite sets is a* **Scott sequence** *for a family S if the following properties hold:*

1. *For each D_i there exists exactly one $M_i \in S$ such that $D_i \subset M$.*
2. *The set $S \setminus \{M_0, M_1, \ldots\}$ is finite.*

The reader can easily prove the following:

Theorem 3.5 *If S has a Scott sequence, then any two computable enumerations of S are equivalent.* $\Box$

For any given family S, we want to construct a structure $\mathcal{A}_S$ such that $\mathcal{A}_S$ has a Scott family if and only if S has a Scott sequence. Thus, let S be a family of sets and let f be a one-to-one computable enumeration of S. We assume that each set in S has at least two elements. Based on f, we will construct a computable structure, indeed a graph, $\mathcal{A}_f = (\omega, P_f)$, where P_f is a computable binary predicate on ω.

Consider a uniformly effective, possibly finite, sequence $a_{i,0}, a_{i,1}, a_{i,2}, \ldots$ without repetitions such that, for each $i \in \omega$, $f(i) = \{a_{i,0}, a_{i,1}, a_{i,2}, \ldots\}$.

For each $i \in \omega$, we can consider a computable structure $\mathcal{G}_i^f = (G_i, P_i^f)$ defined as follows. G_i has an element d_i such that for each $a_{i,j}$ the predicate P_i^f defines a unique cycle $C_{i,j}$ of length $a_{i,j}$ for which $d_i \in C_{i,j}$. In addition, for all j, k the cycles $C_{i,k}$ and $C_{i,j}$ have only one element in common which is d_i. Thus, we see that for all $j \neq k$ we have $d_i \in C_{i,k}$, $d_i \in C_{i,j}$ and $(C_{i,j} \setminus \{d_i\}) \bigcap (C_{i,k} \setminus \{d_i\}) = \emptyset$. We call the element d_i a **cluster point**. Informally, the structure $\mathcal{G}_i^f$ codes the set $f(i)$. The structure (graph) $\mathcal{G}_i^f$ is computable and satisfies the following properties:

1. For every number t, t belongs to $f(i)$ if and only if there exist distinct elements $x_0, \ldots, x_t$ of the structure $\mathcal{G}_i^f$ such that the formula

$$P_i^f(x_0, x_1) \,\&\, \ldots P_i^f(x_{t-1}, x_t) \,\&\, P_i^f(x_t, x_0)$$

holds in the structure $\mathcal{G}_i^f$.

2. Any two cycles in $\mathcal{G}_i^f$ have only one element in common.

By the construction of $\mathcal{G}_i^f$ and the computability of f, we can conclude that there exists a computable sequence $\mathcal{A}_0^f = (A_0^f, P_0)$, $\mathcal{A}_1^f = (A_1^f, P_1)$, $\mathcal{A}_2^f = (A_2^f, P_2), \ldots$ of computable structures such that:

1. For each i the structure $\mathcal{A}_i^f$ is isomorphic to the structure $\mathcal{G}_i^f$.
2. For each pair $i \neq j$, $A_i^f \bigcap A_j^f = \emptyset$ and $\omega = \bigcup_i A_i^f$.
3. The relation $P_f = \bigcup_i P_i$ is computable.

Consider the computable structure $\mathcal{A}_f$ defined in some canonical way so that $\mathcal{A}_f$ is isomorphic to (ω, P_f). Note that the set of all cluster points of $\mathcal{A}_f$ is recursive in every recursive presentation of $\mathcal{A}_f$. (We are here using the assumption that every set in the family S has at least two elements.) The following lemma describes the relationship between S and $\mathcal{A}_f$.

Lemma 3.6 *The structure $\mathcal{A}_f$ satisfies the following conditions.*

1. *If g is a one-to-one computable enumeration of S, then $\mathcal{A}_f$ is isomorphic to $\mathcal{A}_g$.*
2. *The structure $\mathcal{A}_f$ is rigid, that is it does not have any nontrivial automorphisms.*
3. *If g is a one-to-one computable enumeration of S, then $\mathcal{A}_f$ is computably isomorphic to $\mathcal{A}_g$ if and only if f and g are equivalent.*
4. *The dimension of the structure $\mathcal{A}_f$ is equal to the maximal number of nonequivalent one-to-one computable enumerations of S.*
5. *The structure $\mathcal{A}_f$ has a Scott family if and only if S has a Scott sequence.*

Proof. To prove 1, first, note that for any pair $i, j \in \omega$ the graphs $\mathcal{G}_i^f$ and G_j^g are isomorphic if and only if $f(i) = g(j)$. Hence, since f and g are one-to-one enumerations of S, we can conclude that $\mathcal{A}_f$ is isomorphic to $\mathcal{A}_g$.

Any automorphism α of $\mathcal{A}_f$ must be the identity by the construction of $\mathcal{A}_f$ and the fact that f is a one-to-one mapping. This proves 2.

Suppose that f and g are equivalent. There exists a recursive function Φ such that $f = g\Phi$. Hence the structure $\mathcal{G}_i^f$ is isomorphic to the structure $\mathcal{G}_{\Phi(i)}^g$. Hence $\mathcal{A}_f$ and $\mathcal{A}_g$ are computably isomorphic. Let $\mathcal{B}$ be a computable presentation of $\mathcal{A}_f$. Consider an effective sequence $e_0, e_1, e_2, \ldots$ without repetition of all cluster points in $\mathcal{B}$. We define a one-to-one computable enumeration $f_{\mathcal{B}}$ of S as follows:

$$f_{\mathcal{B}}(i) = \{n | e_i \text{ belongs to a circle of length } n\}.$$

It follows that $\mathcal{B}$ is computably isomorphic to $\mathcal{A}_g$ if and only if g is equivalent to $f_{\mathcal{B}}$. This proves 3.

4 follows from the proof of 3.

We are left to prove the last part of the lemma. Suppose that S has a Scott sequence $D_0, D_1, D_2, \ldots$. Without lost of generality we can suppose that $D_i \subset f(i)$. We have to prove that $\mathcal{A}_f$ has a Scott family. Take an $x \in A_f$. Find a d_i which is connected to x via P_f. Suppose that the length of a path which connects x with d_i is n. Define the following formula: $\psi(x) =$ [there exists a path of length n which connects x with a cluster point y such that for each $m \in D_i$ the element y belongs to a cycle of length m]. Now for every s–tuple $(x_1, \ldots, x_s)$ let $\phi_{(x_1,\ldots,x_s)}$ be $\psi(x_1)\&\ldots\&\psi(x_s)$. It is not hard to check that the sequence $\{\phi_{(x_1,\ldots,x_s)}\}$ is a Scott family for $\mathcal{A}_f$.

Now suppose for simplicity that $\mathcal{A}_f$ has a Scott family

$$\phi_0(x_1, \ldots, x_{n_0}), \phi_1(x_1, \ldots, x_{n_1}), \ldots$$

without parameters. The proof below will show that we do not lose any generality by making this assumption. Let $d_0, d_1, d_2 \ldots$ be an effective sequence of all cluster points from $\mathcal{A}_f$. Let

$$\phi_{i_0}(x_0), \phi_{i_1}(x_1), \ldots$$

be an effective subsequence of the original sequence such that $\phi_{i_k}(d_k)$ holds for each $k \in \omega$. Since the formulas are all existential and the structure is computable, we can effectively find a finite substructure $\mathcal{B}_i$ of $\mathcal{A}_f$ such that $d_i \in B_i$ and $\phi_{i_k}(d_k)$ holds in $\mathcal{B}_i$. Define

$$D_i = \{n | d_i \text{ belongs to a cricle of length } n \text{ in substructure } \mathcal{B}_i\}.$$

Since we have a Scott family for $\mathcal{A}_f$ and since the structure $\mathcal{A}_f$ is rigid, we can see that the sequence $D_0, D_1, \ldots$ is a Scott sequence for family S. □

Corollary 3.7 *Any two one-to-one computable enumerations of S are equivalent if and only if $\mathcal{A}_f$ is computably categorical.* □

Now, to prove Theorem 3.1 it suffices, by lemma 3.6, to build a computably enumerable family S of sets without a Scott sequence any two computable one-to-one enumerations of which are equivalent.

Lemma 3.8 *There is a computably enumerable family S of sets with no Scott sequence any two computable one-to-one enumerations of which are equivalent.*

In order to build a such family S and its one-to-one enumeration f, we need to satisfy the following requirements:

$$D_e : \quad F_e \text{ is not a Scott sequence for } S,$$

$$R_j : \quad g_j \equiv f \text{ or } g_j \text{ is not a one-to-one enumeration of } S,$$

where $\{g_j\}_j$ is a computable sequence of all potential one-to-one enumerations of a family of sets, and $\{F_e\}_e$ is a computable sequence of all potential Scott sequences for S. These requirements are similar to the requirements for constructing a computable structure as needed for Theorem 2.5 (see [2]). In fact, to construct such a family S, we essentially use the ideas from the proof of Theorem 2.5. A detailed proof of this lemma and similar results will appear in [12].

References

[1] D. Cenzer, J.Remmel, Polynomial Time vs Recursive Models, *Ann. Pure and Appl. Logic* **54**, 1991.

[2] , P.Cholak, S. Goncharov, B.Khoussainov and R. Shore, On Recursively Categorical Models, in preparation.

[3] J. N. Crossley, ed., *Aspects of Effective Algebra, Proceedings of a Conference at Monash University, 1979.*

[4] Yu. Ershov and S. S. Goncharov, eds., *Logic Notebook: Problems in Mathematical Logic*, Novosibirsk University, 1986.

[5] A. Frölich and J. C. Shepherdson, Effective Procedures in Field Theory, *Phil. Trans. Roy. Soc. London ser. A* **248** (1956), 407-432.

[6] V. D. Dzgoev and S. S. Goncharov, Autostability of Models, *Algebra i Logika* **19** (1980), 45-58.

[7] S. S. Goncharov, The Problem of the Number Of Non-Self-Equivalent Constructivizations, *Algebra i Logika* **19** (1980), 621-539.

[8] V. Harizanov, Pure Recursive Model Theory, preprint, to appear in *Handbook of Recursive Mathematics*, Y. Ershov, S. Goncharov, A. Nerode, J. Remmel eds.

[9] L. Harrington, Recursively Presentable Prime Models, *J. Symbolic Logic* **39** (1973), 305-309.

[10] N. Khissamiev, On Strongly Constructive Models of Decidable Theories, Izvestia AN Kaz. SSR (1974), 83-84, 94.

[11] B. Khoussainov and A. Nerode, Automata Representable Structures, in *Proceedings of Logic and Computational Complexity Conference, 1995*, Springer-Verlag Lecture Notes in Computer Science, to appear.

[12] B. Khoussainov and R. Shore, Recursive Categoricity and Degree Spectra of Relations, in preparation.

[13] A. I. Mal'cev , Constructive Algebras, *Uspekhi Matem. Nauk* **16** (1961), 3-60.

[14] T. Millar, Recursive Categoricity and Persistence, *J. Symbolic Logic* **51** (1986), 430-434.

[15] T. Millar, Abstract Recursive Model Theory, in *Handbook of Recursion Theory*, E. Griffor, ed., North-Holland, to appear.

[16] A. Nurtazin, Strong and Weak Constructivizations and Computable Families, *Algebra i Logika* **13** (1974), 311-323,364.

[17] M. O. Rabin, Computable Algebra, General Theory and Theory of Computable Fields, *Trans. Am. Math. Soc.* **95** (1960), 341-360.

[18] J. B. Remmel, Recursive Isomorphisms Types of Recursive Boolean Algebras, *J. Symbolic Logic* **46** (1981), 572-593.

[19] J. B. Remmel and J. N. Crossley, The Work of Anil Nerode: A Retrospective, in *Logical Methods*, J. N. Crossley, J. B. Remmel, R. A. Shore and M. E. Sweedler, eds., Birkhäuser, Boston, 1993, 1-91.