Research Statement

Asher Kach

1. Introduction

My research in the field of effective algebra (a subfield of mathematical logic) has focused on trying to understand how algebraic structure imposes constraints on how complicated an algebraic object can be or must be. Much of my work has been studying which isomorphism types of algebraic structures (particularly linear orders and Boolean algebras) have effective presentations. Other work of mine has studied the complexity of external relations on algebraic structures, the complexity of embeddings and isomorphisms between algebraic structures, and the effectiveness of algebraic constructions.

The tools and techniques of computability theory are pervasive throughout this study. The robust paradigm of Turing machines (an idealized computer with infinite memory) provides a formalization of the notions of complexity and relative complexity. Specifically, a set $S \subseteq \omega$ of natural numbers is *computable* in this paradigm if there is a Turing machine that decides the membership question for S (i.e., answers whether an arbitrary integer n is in S); a set $S \subseteq \omega$ is *computably enumerable* if there is a Turing machine that enumerates the elements of S. Relative computability is then studied via the notion of Turing reducibility, where a set A is *Turing below* a set B (denoted $A \leq_T B$) if there is a Turing machine that computes the characteristic function χ_A of A when provided the characteristic function χ_B of B as an oracle.

With these notions in place, it is possible to formalize the requirements for an algebraic structure to be effective.

Definition 1.1. A countable algebraic structure having finitely many functions and relations acting on it is *computable* if its universe can be identified with the natural numbers ω in such a way that the functions and relations become computable operations on ω.

For example, the structure $(\mathbb{Z} : +)$ (i.e., the integers with addition) is computable by identifying a nonnegative integer n with the natural number $2n$ and a negative integer n with the natural number $-2n-1$. Addition of natural numbers a and b (coding integers m and n) is then a computable function defined in cases depending on whether a and b are even or odd.

Though the motivation is to better understand the interplay between algebraic structure and effectiveness, my research has uncovered this relationship in a number of ways. Besides having directly answered questions in effective algebra (see Sections 2 and 3), I have established results in classical computability theory that yield results in effective algebra as corollaries (see Section 4), and studied the reverse mathematical strength of statements regarding algebraic structures (see Section 5). Plans for future work are also described (see Section 6).

Several notions from computability theory recur throughout this work. Limitwise monotonic functions (see [9]) describe functions that are not computable but that can be approximated from below in a computable manner.

Definition 1.2. A set $S \subseteq \omega + 1$ is $\mathbf{a}$-*limitwise monotonic* if there is a total $\mathbf{a}$-computable function $f : \omega \times \omega \to \omega$ satisfying $f(x, s) \leq f(x, s+1)$ for all x and s such that S is the range of the function $F(x)$ given by

$$F(x) = \lim_s f(x, s).$$

We use the convention that $F(x) = \omega$ if $\lim_s f(x, s) = \infty$.

The Feiner hierarchy (see [5]) refines the Turing degree $\mathbf{0}^{(\omega)}$ for sets where the number of jumps necessary to compute the membership question for n is linear in n.

Definition 1.3. A set $S \subseteq \omega$ is $\Delta^0_{(bn+a+1)}(X)$ in the Feiner Δ-hierarchy if there is a Turing functional Φ_e such that $\chi_S(n) = \Phi_e^{X^{(bn+a)}}(n)$ for all n.

A set $S \subseteq \omega$ is $\Sigma^0_{(bn+a+1)}(X)$ in the Feiner Σ-hierarchy if there is a Turing functional Φ_e such that $n \in S$ if and only if $n \in W_e^{X^{(bn+a)}}$.

2. Computable Isomorphism Types

A fundamental problem of effective algebra and a major component of my research is determining which algeraic structures are computable. Much of my research in this area has been on linear orders and Boolean algebras.

2.1. **Linear Orders.**

Classically, countable linear orders are quite varied. Largely as a consequence, particular classes of linear orders are considered when attempting to understand the connection between the structure of a linear order and whether or not it is computable. Shuffle sums and η-representations are two common classes that code sets of natural numbers into linear orders.

Definition 2.1. The *shuffle sum* of a countable set of linear orders $S = \{L_i\}_{i \in I}$, denoted $\sigma(S)$, is the (unique) linear order obtained by partitioning the rationals into dense sets $\{Q_i\}_{i \in I}$ and replacing each rational of Q_i by the linear order L_i.

Definition 2.2. Let $S \subseteq \omega$ be enumerated in increasing order as $S = \{a_0 < a_1 < a_2 < \dots\}$ and let η be the order type of the rational numbers. If $\pi : \omega \to \omega$ is a surjective function, then the linear order

$$\mathcal{L}_\pi := \eta + a_{\pi(0)} + \eta + a_{\pi(1)} + \eta + a_{\pi(2)} + \dots$$

is a *weak η-representation for S*. If π is also injective, then $\mathcal{L}_\pi$ is a *unique η-representation for S*; if π is also nondecreasing, then $\mathcal{L}_\pi$ is a *increasing η-representation for S*; and if π is the identity, then $\mathcal{L}_\pi$ is a *strong η-representation for S*.

Roughly speaking, the shuffle sum of a set S is the linear order having infinitely many copies of each $L_i \in S$, with a copy of every $L_j \in S$ between two distinct copies of x. In contrast, η-representations have infinitely many copies of the integers ζ with the elements of S between them.

In both cases, it is intuitive that the complexity of the set S (viewed as a collection of ordinals) and the complexity of the linear order encoding S should be related. Indeed, this is the case. With shuffle sums, there is a direct connection with limitwise monotonic functions.

Theorem 2.3 (Kach [10])**.** *For sets $S \subseteq \omega + 1$, the shuffle sum $\sigma(S)$ is computable if and only if the set S is $\mathbf{0}'$-limitwise monotonic.*

Kenneth Harris (see [6]) connected the sets $S \subseteq \omega$ with computable weak η-representations (computable unique η-representations) with the $\mathbf{0}'$-limitwise monotonic functions using techniques similar to those in Theorem 2.3. Making use of a generalization of limitwise monotonic functions, Daniel Turetsky and I characterized the sets S with computable increasing η-representations.

Theorem 2.4 (Kach and Turetsky [14])**.** *For sets $S \subseteq \omega$, an increasing η-representation of S is computable if and only if the set S is pseudo-increasing $\mathbf{0}'$-limitwise monotonic on $\mathbb{Q}$.*

It is noteworthy that η-representations (which encode the set S at a local level) require the domain of the limitwise monotonic limit function to be large (i.e., the rationals), whereas for shuffle sums (which encode the set S at the global level) the domain can be small (i.e., the natural numbers).

Rather than characterize the isomorphism types with computable copies, it is also sometimes possible to demonstrate that if a structure is almost computable, then it is in fact computable. The notion of being low$_n$ is a means of characterizing almost computable.

Definition 2.5. For a positive integer n, a set $X \subseteq \omega$ is low$_n$ if $X^{(n)} \equiv_T \emptyset^{((n))}$.

With Miller and Montalbán, I showed that as the complexity of the order type of the descending cuts decreases, the extent to which being almost computable implies being computable increases.

Definition 2.6. A *cut* of a linear order $\mathcal{L}$ is a partition $(\mathcal{I}, \mathcal{J})$ of $\mathcal{L}$ where $\mathcal{I}$ is an initial segment of $\mathcal{L}$ and $\mathcal{J}$ is an end segment of $\mathcal{L}$.

A *descending cut* in a linear order $\mathcal{L}$ is a cut $(\mathcal{I}, \mathcal{J})$ such that $\mathcal{J}$ is nonempty and has no least element.

Theorem 2.7 (Kach, Miller, and Montalbán [13])**.** *Let $\mathcal{L}$ be a countable linear order.*

(1) *[Folklore] If $\mathcal{L}$ has no descending cuts, then the existence of a hyperarithmetic copy implies the existence of a computable copy.*

(2) *If $\mathcal{L}$ has finitely many descending cuts, then the existence of a low_n copy (for some n) implies the existence of a computable copy.*
(3) *If $\mathcal{L}$ has infinitely many descending cuts whose order type is scattered, then the existence of a low_2 copy implies the existence of a computable copy.*

Moreover, none of these statements can be strengthened.

It is well known that there are low (i.e., low_1) linear orders $\mathcal{L}$ that are not computable (see [3]), so there are linear orders that are almost computable in a very strong sense but not computable.

2.2. **Boolean Algebras.** Unlike with linear orders, every Boolean algebra exhibits some of the behavior of Theorem 2.7.

Theorem 2.8 (Knight and Stob [16])**.** *If $\mathcal{B}$ is a Boolean algebra with a low_4 copy, then $\mathcal{B}$ is computable.*

It is conjectured that Theorem 2.8 can be strengthened from low_4 to low_n for any n. Rather than attempting to strengthen Theorem 2.8, my work has provided the first characterization of which Boolean algebras have computable presentations for a nontrivial class (namely, the depth zero, rank ω Boolean algebras). Before being able to do so, an algebraic understanding of this class was necessary. Making explicit work of Heindorf (see [7]), I provided a constructive definition of the measures for depth zero Boolean algebras using the isomorphism invariants developed by Ketonen (see [15] or [17])). Afterwords, the computable depth zero Boolean algebras were connected with the Feiner hierarchy.

Theorem 2.9 (Kach [11])**.** *For sets $S \subseteq \omega + 1$ with maximal element, the following are equivalent:*
(1) *The depth zero Boolean algebra $\mathcal{B}_{u(S)}$ is computable.*
(2) *The depth zero Boolean algebra $\mathcal{B}_{v(S)}$ is computable.*
(3) *The set S is $\Sigma^0_{(2n+3)}$ in the Feiner Σ-hierarchy.*

3. Embeddings of Computable Structures

In addition to studying which algebraic structures are computable, my research has also studied the effectiveness of embeddings between structures. It is well known that there are fixed presentations of computable structures $\mathcal{S}_1$ and $\mathcal{S}_2$ for which there is a classical embedding of $\mathcal{S}_1$ into $\mathcal{S}_2$ but no computable (indeeed, even hyperarithmetic) embedding between the presentations. It is natural to ask whether this is possible only because unnecessary complexity is being introduced into the computable presentation of $\mathcal{S}_1$ and $\mathcal{S}_2$. The answer, as my research has demonstrated, depends largely on the class of algebraic structures considered. For linear orders, the phenomena happens for many choices of the smaller structure.

Theorem 3.1 (Kach and J. Miller [12])**.** *There is a computable non-scattered linear order $\mathcal{L}_\eta$ that is intrinsically hyperarithmetically scattered.*

For each computable ordinal α, there is a computable non-well-ordered linear order $\mathcal{L}_{\omega^*}$ that is intrinsically $\mathbf{0}^{(\alpha)}$ well-ordered.

Indeed, the same phenomena happens in many other classes of algebraic structures.

Theorem 3.2 (Calvert, Kach, Levin, and Solomon [2]; Kach and J. Miller [12])**.** *If $\mathcal{C}$ is the class of directed graphs, the class of undirected graphs, the class of commutative rings, the class of two-step nilpotent groups, the class of integral domains, the class of commutative semigroups, or the class of ordered fields, there are computable structures $\mathcal{S}_1$ and $\mathcal{S}_2$ in $\mathcal{C}$ such that $\mathcal{S}_1$ classically embeds into $\mathcal{S}_2$ but for no hyperarithmetic presentations of $\mathcal{S}_1$ and $\mathcal{S}_2$ does $\mathcal{S}_1$ hyperarithmetically embed into $\mathcal{S}_2$.*

However, it does not happen in all classes of algebraic structures.

Theorem 3.3 (Calvert, Kach, Levin, and Solomon [2])**.** *If $\mathcal{C}$ is the class of equivalence structures or Boolean algebras, then if $\mathcal{S}_1$ classically embeds into $\mathcal{S}_2$, there is a computable embedding of $\mathcal{S}_1$ into $\mathcal{S}_2$ for some computable presentations of $\mathcal{S}_1$ and $\mathcal{S}_2$.*

Again this highlights the interplay between algebraic structure and complexity. The reason Boolean algebras behave differently is that they have no local structure that isn't reflected at the global scale; the reason equivalence structures behave differently is that they have no local structure which isn't duplicated at the global scale.

4. Classical Computability Theory

Besides proving results in effective algebra directly, it is often possible to transfer results in pure computability theory to results in effective algebra. For example, my studies of the limitwise monotonic sets and the Feiner hierarchy have yielded results about linear orders, Boolean algebras, and other structures.

Theorem 4.1 (Hirschfeldt, Kach, Montalbán [8])**.** *There is a computably enumerable set A of intermediate Turing degree such that A is low for Σ-Feiner.*

Noting that Theorem 2.9 relativizes, Theorem 4.1 yields the following corollary.

Corollary 4.2. *There is a (depth zero, rank ω) Boolean algebra of intermediate degree that is not computable.*

Other results on η-representations followed from classical computability theory results.

Theorem 4.3 (Kach and Turetsky [14])**.** *Every Δ^0_2 degree is support increasing limitwise monotonic on $\mathbb{Q}$.*

Corollary 4.4. *Every Δ^0_3 degree has a computable increasing η-representation. Hence there are sets (degrees) with computable increasing η-representations but not a computable strong η-representation.*

5. Reverse Mathematics

Effective algebra also has connections to reverse mathematics as there is often a connection between the proof theoretic strength of asserting an algebraic object exists and a standard system of second order arithmetic. For example, there is a strong connection between the proof theoretic strength of asserting certain vector subspaces exist and the standard systems.

Theorem 5.1 (Downey, Hirschfeldt, Kach, Lempp, Mileti, Montalbán [4])**.** *Over RCA_0, WKL_0 is equivalent to the statement "Every vector space (over an infinite field) of dimension greater than one has a proper subspace."*

Over RCA_0, ACA_0 is equivalent to the statement "Every vector space (over an infinite field) of dimension greater than one has a proper finite-dimensional subspace."

6. Future Work

My plans for future work center on the theme of trying to understand the connection between algebraic structure and effectiveness. Several natural questions arise from previous results.

Question 6.1. For which sets $S \subseteq \omega$ is the strong η-representation for S computable?

Question 6.2. Are there computable abelian p-groups $\mathcal{G}_1$ and $\mathcal{G}_2$ such that $\mathcal{G}_1$ classically embeds into $\mathcal{G}_2$ but for no computable presentations of $\mathcal{G}_1$ and $\mathcal{G}_2$ does $\mathcal{G}_1$ computably (hyperarithmetically) embed into $\mathcal{G}_2$?

Question 6.3. Are there computable trees (viewed as posets) $\mathcal{T}_1$ and $\mathcal{T}_2$ such that $\mathcal{T}_1$ classically embeds into $\mathcal{T}_2$ but for no computable presentations of $\mathcal{T}_1$ and $\mathcal{T}_2$ does $\mathcal{T}_1$ hyperarithmetically embed into $\mathcal{T}_2$.

As a consequence of work by Binns, Kjos-Hanssen, Lerman, Schmerl, and Solomon, Theorem 3.2 is known to be true for the class $\mathcal{C}$ of trees (viewed as posets) if the word "hyperarithmetic" is replaced by "computable" (see [1]).

Another interest of mine is to better understand the countable Boolean algebras, both classically and effectively. For example, the finite depth, finite rank Boolean algebras are well understood classically and effectively (see [7]). My work has clarified the depth zero, rank ω Boolean algebras. It is natural to explore when the depth is allowed to be infinite and the rank is kept small.

Question 6.4. What are the depth ω, rank one Boolean algebras? Which of these are computable?

Lastly, I'm interested in countable Boolean algebras from the viewpoint of Vaught's Conjecture.

Question 6.5. If φ is a sentence in the language of $L_{\omega_1,\omega}$ whose models are Boolean algebras, does φ having uncountably many countable models necessitate continuum many countable models?

Ketonen invariants again seem like a useful tool to answer this question.

References

[1] Stephen Binns, Bjorn Kjos-Hanssen, Manuel Lerman, James H. Schmerl, and Reed Solomon. Self embeddings of computable trees. *Trans. Amer. Math. Soc.*, 49:1–37, 2008.

[2] Wesley Calvert, Asher M. Kach, Oscar Levin, and D.R̃eed Solomon. Embeddings of computable structures. In preparation.

[3] R. G. Downey. Computability theory and linear orderings. In *Handbook of recursive mathematics, Vol. 2*, volume 139 of *Studies in Logic and the Foundations of Mathematics*, pages 823–976. North-Holland, Amsterdam, 1998.

[4] Rodney G. Downey, Denis R. Hirschfeldt, Asher M. Kach, Steffen Lempp, Joseph R. Mileti, and Antonio Montalbán. Subspaces of computable vector spaces. *Journal of Algebra*, 314:888–894, 2007.

[5] Lawrence Feiner. Hierarchies of Boolean algebras. *J. Symbolic Logic*, 35:365–374, 1970.

[6] Kenneth Harris. η-representations of sets and degrees. *J. Symbolic Logic*, to appear.

[7] Lutz Heindorf. Alternative characterizations of finitary and well-founded Boolean algebras. *Algebra Universalis*, 29(1):109–135, 1992.

[8] Denis R. Hirschfeldt, Asher M. Kach, and Antonio Montalbán. A Feiner look at the intermediate degrees. In preparation.

[9] N. G. Hisamiev. Criterion for constructivizability of a direct sum of cyclic p-groups. *Izv. Akad. Nauk Kazakh. SSR Ser. Fiz.-Mat.*, (1):51–55, 86, 1981.

[10] Asher M. Kach. Computable shuffle sums of ordinals. *Archive for Mathematical Logic*, 47(3):211–219, 2008.

[11] Asher M. Kach. Depth zero Boolean algebras. Submitted.

[12] Asher M. Kach and Joseph S. Miller. Embeddings of computable linear orders. In preparation.

[13] Asher M. Kach, Joseph S. Miller, and Antonio Montalbán. Linear orders and descending cuts. In preparation.

[14] Asher M. Kach and Daniel Turetsky. Limitwise monotonic functions, sets, and degrees on computable domains. Submitted.

[15] Jussi Ketonen. The structure of countable Boolean algebras. *Ann. of Math. (2)*, 108(1):41–89, 1978.

[16] Julia F. Knight and Michael Stob. Computable Boolean algebras. *J. Symbolic Logic*, 65(4):1605–1623, 2000.

[17] R. S. Pierce. Countable Boolean algebras. In *Handbook of Boolean algebras, Vol. 3*, pages 775–876. North-Holland, Amsterdam, 1989.