Research Statement

Oscar Levin

When I first encountered proofs in an undergraduate discrete math course, I was immediately drawn by the apparent power the techniques provided. Surely this tool would be able to provide answers to all of life's questions. But of course I soon learned that not all mathematical truths can be proved easily and that in fact there are some for which no proof exists, nor ever will. More strangely still, the complexity or complete lack of proof seemed to have little connection to the complexity of the theorem. I became fascinated by the question as to why some theorems are easier to prove than others. Thus my interest in mathematical logic began.

There are various ways to investigate this phenomenon but my favored approach is computability theory. I am especially interested in applying this technique to algebra and combinatorics, so the research program I am involved in is often referred to as *effective algebra*. Combining computability theory and algebra has applications beyond the provability questions posed above - the last century has seen techniques from computability theory lead to the discovery that, for instance, the word problem for groups is undecidable, as well as the undecidability of Hilbert's 10th problem. Investigating algebraic and combinatorial structures from the computability viewpoint can improve our understanding of the algebra and combinatorics themselves. In the next few pages, I will describe how all this works using some examples from my own research. We begin with...

The Basics

When investigation the complexity of a theorem or proof, the basic idea is to place some sort of restriction on either the proof techniques or the types of objects considered. The goal is to find a restriction which invalidates the theorem, as this will illustrate where the complexity lies. If you were using set theory, you might ask which theorems fail when we remove this or that axiom (for example, the Axiom of Choice). Using computability theory, we instead place the restriction on the types of objects considered: *computable* ones. We often find that when we require the underlying structures to be described effectively, the theorems are not *effectively* true. We can then ask to what extent the theorem fails - how much extra computational power is required. This lends insight into how complicated a theorem might be: the hypothesis for the theorem might only mention very simple structures, but the conclusion requires much more complex ones.

"Computable" and "effective" are precisely defined in [5] and elsewhere, but can be safely thought of as that which a computer can (theoretically) describe. For example, a set of natural numbers is computable just in case there is a computer program that can determine membership. A function is computable if there is some computer program which, given the input to the function, can compute the output. A relation is computable if there is some computer program which decides whether the relation holds on given inputs. These computer programs are theoretical in that they can use an arbitrary amount of memory, and take arbitrarily long to give an answer - we are not concerned here with efficiency, only with whether or not an algorithm exists. And there are sets of natural numbers which are not computable. Notice first that we can list out (in some order) every computer algorithm which accepts exactly one input: $\{\varphi_1, \varphi_2, \varphi_3, \ldots\}$. These algorithms need not *halt* on all inputs; some inputs could cause the algorithm to enter an infinite loop. Now consider the set of numbers e for which the eth algorithm halts on the input e. That is, $K = \{e \in \mathbb{N} \mid \varphi_e(e)\downarrow\}$. This set (called the *halting problem*) is not computable. (In fact most every set of natural numbers is non-computable, as there are only countably many computer programs.)

Now consider an algebraic or combinatorial structure, such as a graph. For us a graph is a collection of vertices, some of which are related (i.e., there is an edge between them). For a graph to be computable, the set of vertices must be a computable set and the edge relation needs to be computable as well. Notice that with this definition, we can ask, given any two vertices, whether they are connected by an edge and in some finite amount of time receive an answer. However we may not be able to find all vertices connected to a given vertex, nor even the degree of that vertex. We could start checking each vertex, but we will never know when we have found the last one.

In my thesis I focused on ordered fields. We say an ordered field is computable if the field operations ($+$ and $\cdot$) are computable functions, and the $\leq$ relation is computable. Since formal definitions for computability apply to $\mathbb{N}$, we always take the natural numbers to be the elements of the field - think of "coding" the field into a language a computer can understand. Notice this implies that computable ordered fields are necessarily countable. Most countable ordered fields you can think of (the rationals, all finite extensions of $\mathbb{Q}$, both algebraic and transcendental) can be presented in a computable way.

Breaking Theorems

Consider a classical result about the relationship between ordered fields and formally real fields (one for which -1 is not a sum of squares). Artin and Schrier proved that every formally real field is orderable. To understand the complexity of this theorem, we ask whether it is effectively true: if we start with a *computable* formally real field, must it be a computably orderable? The answer is no. This result is a consequence of the the work in [3], which classifies the effective strength of the space of all orderings of a field. The idea is to build a computable real field (the domain is computable, and

+ and · are computable) so that any choice of positive elements (which determines the ordering) can compute a non-computable set.

This failure in the computable case seems to be intrinsic to the problem of extending orders. In my dissertation I considered some related theorems about order extension. Under certain conditions, both partial orders as well as pre-orders can be extended to full orders on the field, or to full orders on an extension field (see [1] and [4], respectively). I showed that each of these theorems fail to be effective - that there are fields with computable partial orders extending to full orders, but not extending to computable full orders.

These examples suggest that there is something complicated about extending orders on fields. However, in each case, the proof proceeded by finding an example of a computable field for which the orders cannot be effectively extended. One might wonder whether this is simply an artifact of the way in which the fields was presented instead of some intrinsic property of the fields. Perhaps there is a more straight forward way to describe the same field for which it is easy to extend the order. To understand this, we need to investigate the relationship between various codings of a structure.

Isolating Complexity

There are many ways to represent (code) a structure computably. We wish to distinguish between complexity in the coding and complexity in the structure itself. One way to know whether we need worry about this is to know the computable dimension of the structure. The computable dimension counts how many computable copies there are of a structure up to *computable* isomorphism. If the computable dimension is 1, we say the structure is *computably categorical*: for every computable copy $\mathcal{B}$ of a structure $\mathcal{A}$, there is a computable isomorphism from $\mathcal{A}$ to $\mathcal{B}$. Notice that this means that any computable-theoretic property of one computable copy of the structure must be held by *all* computable copies of the structure. The particular coding of a computably categorical structure does not matter. On the other hand, if there are computable copies of the structure which are, while necessarily isomorphic, not isomorphic by a computable isomorphism (the computable dimension is greater than 1), then the coding can be very important.

As it turns out, while computable dimension has been sorted out for almost all general classes of structures, determining computable dimension for fields is a highly non-trivial problem. One of the goals of my dissertation was to see if inroads could be made by considering *ordered* fields as a special case. I was able to make some progress by showing that all computable ordered fields with finite transcendence degree are computably categorical. This is not true of fields in general, as there are non-computably categorical fields with finite transcendence degree. Here, the order helps. However, when you look at fields with infinite transcendence degree, problems present themselves. In my thesis I was able to show that in same basic cases, the computable dimension had to be infinite. Since then, I have discovered that the same problems that plague the general fields case are present here to: there are fields (ordered and not) with infinite transcendence degree which are computably categorical. The order is no help. The natural division of finite vs. infinite transcendence degree (which works for algebraically and real closed fields) does not determine the computable dimension for (ordered) fields in general. Finding where that division is is part of my current research moving forward.

More Computability Questions

Given a computable ordered field, there are subsets, additional relations and functions which may or may not be computable. For example, I have shown that there is a computable ordered field such that the set of elements which are sums of squares fails to be computable. The field constructed is computably categorical, so the particular presentation is not to blame for our inability to determine whether an element is a sum of squares.

Or consider the transcendence basis of a computable field. I have shown that they need not be computable. In fact, they can in some sense be as complicated as possible: there is a computable ordered field such that every transcendence basis computes the halting problem. I would like to extend this result to say something about *pure* transcendence bases, which are needed in the construction of non-computably isomorphic copies of certain ordered fields of infinite transcendence degree. It appears that finding a pure transcendence basis is more difficult (from a computability standpoint) than finding just any transcendence basis, but so far, this question is open.

Of course these techniques can be applied to a wide variety of algebraic and combinatorial structures. Indeed it is often interesting to consider how a particular computable-theoretic property applies across different structures. Suppose you have a structure $\mathcal{A}$ which embeds into another structure $\mathcal{B}$. You might wonder whether $\mathcal{A}$ and $\mathcal{B}$ being computable implies that the embedding is a computable function. The answer depends on the structures, as there are classes of computable structures which always admit computable embeddings as well as classes of computable structures for which no embedding is computable. It can also happen that either $\mathcal{A}$ or $\mathcal{B}$ has a computable copy which admits a computable embedding, even if the original structures do not. In a recent paper with Asher Kach and Reed Solomon, we were able to separate each possible case by consider a variety of computable structures including Boolean algebras, partially ordered sets, trees, ordered groups, equivalence structures and algebraically closed fields (see [2]).

Future Research

There is still plenty left to investigate about the effective content of ordered fields. In addition to the questions left open in the above discussion, I would like to look at the relationships between ordered fields and valuation theory in the context of computability. Beyond ordered fields, I plan to expand my work to include other algebraic and combinatorial structures. I have recently started a research project on computability and planar graphs, working with Matt Jura and Tyler Markannen. Much is already known about effective graph theory, but little has been done looking at planar graphs, so we hope this will lead to a few publications.

In addition to continuing my own research into these topics, I am very much looking forward to working with students on research topics. Because effective algebra builds off relatively simple algebraic concepts, the pre-requisites for working in this area are minimal. With an advanced course in abstract algebra or combinatorics, a talented undergraduate student could make a good study of this subject for an undergraduate thesis or summer research project. As an undergraduate, I was lucky enough to participate in an REU (Research Experience for Undergraduates) and also have an adviser who encouraged research, leading to a talk at the regional MAA meeting. I would very much like to provide similar opportunities for the next generation of mathematicians, and my particular research interests are well suited for this endeavor.

One final thought about my approach to research: I enjoy doing research primarily because I enjoy working on hard problems, especially when working together with other mathematicians. I fully intend to continue working in the area of effective algebra, but would also welcome to opportunity to branch out. I particularly enjoy problems in algebra and combinatorics, as well as theoretical computer science - computability theory can be thought of as the infinite version of computational complexity. And although the research methods in math education are quite distinct from my specialty, I am very interested in improving math education and would thoroughly enjoy collaborating on a project with an expert in that field. I find the questions in effective algebra outlined above very interesting, but I realize that it is a bit of a niche subject. The true joy of mathematics is sharing it with others, so I place higher importance on the camaraderie of collaboration than on the specific questions we study.

References

[1] Fuchs, L., *Partially Ordered Algebraic Systems*, Pergamon Press, New York, 1963.

[2] Kach, A.M., Levin, O., Solomon, R., *Embeddings of Computable Structures*, Notre Dame J. Formal Logic, Volume 51, Number 1 (2010), 55-68.

[3] Metakides, G., Nerode, A., *Effective Content of Field Theory*, Annals of Mathematical Logic 17 (1997) 286-320.

[4] Prestel, A., *Lectures on Formally Real Fields*, Lecture Notes in Mathematics 1093, Springer-Verlag, 1984.

[5] Soare, R.I., *Recursively Enumerable Sets and Degrees,* Perspectives in Mathematical Logic, Springer-Verlag, New York, 1987.