Research Statement

ANTONIO MONTALBÁN

November 17, 2008

My area of research centers around Computability Theory, a branch of Mathematical Logic. Inside computability theory, I have worked in various different areas. I have been particularly interested in the programs of Computable Mathematics, Reverse Mathematics and Turing Degree Theory. The former one studies the computability aspects of mathematical theorems and structures. The second one analyzes the complexity of mathematical theorems in terms of the complexity of the constructions needed for their proofs. The latter, which is considered Pure Computability Theory, studies the partial ordering induced by the relation "computable from".

I have also written a papers in other areas like Effective Randomness, automata theory, the structure of linear orderings, the lattice of Π^0_1-classes, and Borel structures. However, most of my work is on the three programs described before. The next section is dedicated to quickly describe these programs. The last section contains a few of my most interesting results, and a couple open questions I am working on.

BASIC CONCEPTS

The main concept in computability theory is the relation "computable from". A set $A \subseteq \mathbb{N}$ is said to be *computable from* a set $B \subseteq \mathbb{N}$, and we write $A \leq_T B$, if there is a computable procedure that can tell whether an element is in A or not using B as an *oracle*, that is, we let the procedure use the information of which elements are in B. A set A is said to be *computable* if it is computable without the use of any oracle. We chose to work with subsets of $\mathbb{N}$ because this is enough: every finite object can be encoded by a single number (using, for instance, the binary representation of the number). For instance, strings, graphs, trees, simplicial complexes, group presentations, etc., if they are finite, they can be coded by a single number. Here is an example: we can encode with a set of natural numbers, the set of finite triangulations of simply connected compact manifolds. It can be shown then that this set is not computable, implying that there is no algorithm to decide simply connectedness. But, on the other hand, this set is computable from the set of natural numbers which encodes the set of group presentations representing a non-trivial group (called the *word problem*).

Computable Mathematics. Effective mathematics is concerned with the computable aspects of mathematical objects and constructions. I have been working on questions like the following: How computationally complicated is common mathematical practice? When can a mathematical structure be represented computably? Can information be encoded into an isomorphism type of a structure? How can we measure the complexity of a mathematical proof?

In Computable and Reverse Mathematics, my research has concentrated on linear orderings, well-quasi-orderings and Boolean algebras. But I have also worked with other type of structures like torsion-free abelian groups, vector spaces, and on computable model theory where we consider general types of structures. I now have a student working on Artinian Rings. Results in Computable and Reverse Mathematics usually require a deeper understanding of the objects from classical mathematics. For instance I have obtained interesting results purely on the structure of the embeddability relation on linear orderings as explained below.

Reverse Mathematics. The questions of which axioms are necessary to do mathematics is of great importance in Foundations of Mathematics and is the main question behind the program of Reverse Mathematics. Reverse Mathematics deals with the system of second-order arithmetic which is rich enough to be able to express an important fragment of classical mathematics. This fragment includes number theory, calculus, countable algebra, real and complex analysis, differential equations, separable metric spaces and combinatorics among others. Almost all of mathematics that can be modeled with, or coded by, countable objects can be done in second-order arithmetic.

The idea of Reverse Mathematics goes as follows. We start by fixing a basic system of axioms. The most commonly used basic system is called RCA_0 that essentially says that computable sets exists. Now, given a theorem of "ordinary" mathematics, the question we ask is what axioms do we need to add to the basic system to prove this theorem. It is often the case in Reverse Mathematics that we can show that a certain set of axioms is necessary to prove a theorem by showing that the axioms follow from the theorem using the basic system. Because of this idea, this program is called Reverse Mathematics. Many different systems of axioms have been defined and studied. But a very interesting fact is that most of the theorems, whose proof-theoretic strength has been analyzed, have been proved equivalent over RCA_0 to one of five systems, that we will call the *main five*.

Effective Mathematics is closely related to Reverse Mathematics. The reason is the following: Many of the main axiom systems of second order arithmetic are equivalent to statements of the form "sets of a certain computational complexity exist", and also to statements of the form "constructions of a certain type are allowed". So, when we study the proof-theoretic strength of a theorem, many times we end up studying the complexity of the constructions in the proof of the theorem, and the complexity of the objects involved in the proof. As we said above, this is one of the interests of Effective Mathematics.

Turing Degree Theory. The Turing degree structure is a very natural structure defined as follows. The relation $\leq_T$ (defined above) is a quasi-ordering on $\mathcal{P}(\mathbb{N})$, the set of subsets of $\mathbb{N}$. It induces an equivalence relation ($A \equiv_T B \iff A \leq_T B \;\&\; B \leq_T A$) and a partial ordering on the equivalence classes. The equivalence classes are called *Turing degrees*. We use $(\mathbf{D}, \leq_T)$ to denote this partial ordering. With the intention of studying the relation $\leq_T$ in abstract, one of the main goals of Computability Theory is to understand the structure of $(\mathbf{D}, \leq_T)$.

The Turing degrees form an *upper semilattice*; that is, every pair of elements $\mathbf{a}$, $\mathbf{b}$ has a least upper bound $\mathbf{a} \vee \mathbf{b}$. Intuitively, $\mathbf{a} \vee \mathbf{b}$ contains all the information that $\mathbf{a}$ and $\mathbf{b}$ have toghether. There is another naturally defined operation called the *Turing jump* (or just *jump*). The jump of a degree $\mathbf{a}$, denoted $\mathbf{a}'$, is given by the degree of the *Halting Problem* relativized to some set in $\mathbf{a}$. (Given $A \subseteq \mathbb{N}$), the *Halting Problem relative to* A, denoted by A', is the set of codes for computer programs that, when run with oracle A, halt.) It can be shown that the jump operation is strictly increasing (i.e., $\forall \mathbf{a}(\mathbf{a} <_T \mathbf{a}')$) and monotonic (i.e., $\mathbf{a} <_T \mathbf{b} \implies \mathbf{a}' <_T \mathbf{a}'$). A *jump upper semilattice* is an upper semilattice together with a strictly increasing, monotonic function.

Various approaches have been taken to understanding the shape of the Turing Degree Structure. One is to study the algebraic properties of the structure. Once people realized the structure is a quite complicated one, methods from logic started being used to study the complexity of the structure. Another approach has been studying how algebraic properties of certain Turing degrees in this structure relate to properties about the computational power of the degree. There is a lot of interaction between these approaches and I have been interested in this program in general. I have written a survey paper (23) on the history of the study of the Turing Degree Structure via embeddability results where I mention my contributions to the area until 2006.

Some results

Computable presentations of structures. In 1955, Clifford Spector proved that every hyperarithmetic well ordering is isomorphic to a computable one. In less technical terms this says that if an ordinal has a representation of a certain complexity (hyperarithmetic, which is quite high) then it also has a very simple (computable) representation. This theorem is central in Hyperarithmetic theory. I proved the following surprising generalization to all countable linear orderings:

Theorem 0.1. *Every hyperarithmetic linear ordering is equimorphic with a computable one. (Two linear orderings are* equimorphic *if they can be embedded in each other.)*

The proof of this Theorem requires a deep analysis of the structure of the countable linear orderings modulo equimorphisms. This analysis led me to define equimorphism invariants for the class of scattered linear ordering of any size. These invariants are finite trees with nodes labeled with ordinals. They are equimorphism invariants in the sense that two linear orderings are equimorphic if and only if they are assigned the same invariant. The invariants provide a new description for the partial orderings induced by the embeddability relation on the class of scattered (and of all countable) linear orderings. I have written a survey paper (13) about my results on linear orderings before 2006.

Boolean Algebras. One of the questions I am the most interested in solving is the following well-known open question in Computable Mathematics.

Q1: Does every low_n Boolean algebra have a computable isomorphic copy?
(A set X is low_n is it has the same nth Turing jump as the empty set.) It has been proved that every low_4 Boolean algebra has a computable copy, but the general case reminds open. With K. Harris we have made considerable progress on the question. We achieved a very good understanding of the back-and-forth relations, and we believe this is a key step towards the solution of the problem, and will be a useful tool for other work with Boolean Algebras. I have also started to studying similar behaviors on other structures and analyzing structural reasons for that behavior.

Length of well-quasi-orderings. Here is a typical example of the effective-content analysis of a theorem: DeJongh and Parik 1977 proved that every wqo has a linearization that is maximal among all its possible linearization. (A *well-quasi-ordering*, or *wqo*, is a quasi-ordering without infinite descending sequences and without infinite antichains.) The order type of these maximal linearizations, called the *length* of the wqo, the is often used to measure well-quasi-orderness when used in applications in combinatorics, proof theory and re-writing systems in Computer Science. So far there was no uniform procedure to find maximal linearizations, and every proof computing them used some new idea or trick. I have shown that this can be done computably; every computable wqo has a computable linearization. However, there is no single computer program that is able to find these maximal linearizations, assuming program for the computable wqo is given as input (there is not even a hyperarithmetic procedure). My interest in understanding the computability aspect of this procedure came from the study of Fraïssé's conjecture (see below) and other results from Proof Theory.

Linear orderings. Fraïssé's conjecture (proved by Laver in 1971) is the statement that says that the countable linear orderings form a wqo with respect to embeddablity. It has interested logicians for many years because of the difficulty of its proof in terms of reverse mathematics; it uses constructions which are more computationally complicated than most of the theorems of mathematics. From my work, it follows that this statement has a *robustness property* in the sense that it is equivalent to many other statements talking about the same type of objects. It also follows that to assume Fraïssé's conjecture is sufficient and necessary to develop a reasonable theory linear orderings and the embeddability relation. So far, the only systems with this robustness property were the main five, but we do not know that Fraïssé's conjecture is equivalent to one of these five. It was conjectured by Clotein 1990 that it is equivalent to Friedman's system of Arithmetic Transfinite Recursion (ATR_0). This problem is still open and plan to work on it. One possible approach is to study the length of the wqo's involved, since this usually gives proof-theoretic information. Together with Marcone we have recently made some progress on this approach; we calculated the length of the wqo of linear orderings of finite Hausdorff rank, obtaining proof theoretic consequences, plus of course, a better understanding of the structure.

Turing Degrees. One approach to understanding the shape of the Turing Degree Structure has been by studying the structures that can be embedded into it. Kleene and Post, in the same paper where they introduced the Turing degree structure in 1955, proved that every finite upper semilattice can be embedded into $(\mathbf{D}, \leq_T)$. Since then, various other embeddablity results have been proved. For countable structures, the most general result proved so far is the following:

Theorem 0.2. *(Montalbán) Every countable jump upper semilattice can be embedded into the Turing Degrees* $(\mathbf{D}, \leq_T, \vee, ')$ *(of course, preserving join and jump).*

Another interesting result I proved along these lines is that the question of whether it is possible to embed every jump upper semilattice of size $\aleph_1$ satisfying the countable predecessor property into $(\mathbf{D}, \leq_T, \vee, ')$ is independent of ZFC.