Manuel Lerman. ***A framework for priority arguments.*** Lecture Notes in Logic, vol. 34. Cambridge University Press, NY, 2010, xvi + 176 pp.

This is a book about one of the main methods in the area of computability (or recursion) theory. The *priority method* was invented by Friedberg and independently Muchnik for the solution of one of the main problems in the area at the time, namely Post's problem (whether there are computably enumerable degrees other than zero and the degree of the halting problem). Since then, the priority method has been developed considerably and is the main type of argument employed for the investigation of the structure of computably enumerable degrees. Moreover, it has found applications in other areas like computable mathematics, but it is ultimately connected to theorems that involve computably enumerable objects.

Since the Friedberg/Muchnik papers, priority arguments have become very complex and sophisticated. There have been various attempts to make them more systematic or even create a sort of framework for these arguments. For example Harrington gave his own account of a notoriously hard priority argument of Lachlan (the Lachlan non-splitting theorem) which was highly influential in understanding and presenting such proofs. This and other more systematic approaches (like one by Ash and one by Knight for constructions in computable model theory as well as unpublished work by Groszek and Slaman) have influenced the framework presented in this book.

Frameworks for priority arguments are usually needed when a theorem involves many levels (typically, infinitely many) of an underlying inductive structure. For example, the framework presented in this book is based on a framework developed by the author and Steffen Lempp, which was ultimately used in to prove the decidability of the existential theory of the poset of computably enumerable degrees in a language containing predicates representing the relations $\mathbf{a}^{(n)} \leq \mathbf{b}^{(n)}$ for all integers n. The same holds for Ash, who developed a framework in order to prove a theorem in computable model theory that referred to all levels of the hyperarithmetical hierarchy.

Apart from such cases where a framework was developed out of the need to prove a theorem, it is arguably desirable to have a uniform approach that encompasses all priority arguments. Indeed, on the one hand priority proofs tend to require lengthy presentations and on the other hand a priority proof seems to have a lot in common with other priority proofs. The point is that if we could isolate the general common principles that are used in these arguments (and are repeatedly demonstrated in each of these proofs) we could adopt a more generic approach in which proofs become shorter by making references to general 'framework theorems'. Such an approach is adopted in set theory with forcing arguments. It is fair to say, however, that in computability theory only a very small proportion of proofs are done in a formal framework. Even forcing arguments in computability theory are most often presented in a self-contained manner, without appealing to a forcing framework. It is arguable that presenting a priority proof with reference to a general framework tends to hide the intuition and the main ideas that underly the argument. It definitely separates the proof from the theorem it proves.

The book starts with a brief chapter giving the definition of the computably enumerable degrees along with notation and related notions. It also briefly discusses the notion of *sequences of trees* which plays a central role in the framework for priority arguments. Chapter 2 lays down the basics of the framework: *systems of trees of strategies*. As usual in priority arguments, we start by writing out a set of requirements which guarantee the theorem we wish to prove. Then for each requirement we introduce a basic module, which describes the way the requirement is going to be satisfied. According to the framework, basic modules are finite binary trees whose edges and non-terminal nodes are labeled with sentences. The *directing sentence* is the one that directs the action taken. The framework involves a decomposition of requirements that starts at some high level tree T^n and continues successively to lower levels until we reach the computable level T_0 at which the construction takes place. Further details are illustrated by means of examples, like the Friedberg–Muchnik theorem.

It has to be said that for someone who is familiar with the standard proofs of such basic theorems, the framework looks quite involved. It basically amounts to learning how to do something we already know, but in a different way. In any case, it seems necessary to spend considerable time learning how to prove well known theorems in this manner, before one can benefit from the framework. Moreover this presentation of priority arguments may be quite intimidating for someone starting out learning computability theory.

Various devices associated with the framework, like *links* and *blocks* are discussed before the *framework theorem* is presented at the end of Chapter 2. Various additional framework lemmas are needed for the proof of various theorems. These are stated throughout the book (in the proofs of the various theorems) but their proof is deferred until Chapters 8 and 9.

One of the criticisms about this framework (or the early versions of it) is that it needs a lot of patches and extensions in order to accommodate various theorems whose proof does not exactly fit the basic setting. This however seems unavoidable since, despite the apparent overlap in many priority proofs, new theorems often require new ideas and non-trivial modifications and devices in their proof. *No other attempt to organize priority arguments in a generic way has been carried out in such a detailed manner, covering most of the fundamental priority constructions in recursion theory.*

Chapters 3–7 present various representative priority arguments within the framework. These are organized in a familiar way, roughly speaking according to the number of quantifiers needed in order to fully describe how the underlying requirements are satisfied. Chapter 3 deals with Σ_1 constructions like the low computably enumerable non-computable degree and the properly d.c.e. degree. Such constructions are sometimes called 'finite injury of bounded type' since the number of injuries of a requirement can be bounded by a computable function on the indices of the requirements. These arguments are perhaps too simple to demonstrate advantages of the framework approach.

Chapter 4 deals with Δ_2 constructions. These are sometimes called 'finite injury of unbounded type' arguments. The Sacks splitting theorem is a typical example. Another example is the construction of a computably enumerable degree which is incomparable to a given incomplete non-computable computably enumerable degree (also due to Sacks). In these arguments there is no recursive bound on the (finite) number of injuries that a strategy may endure.

Chapter 5 deals with Π_2 constructions. These are often called *infinite injury arguments* since strategies may endure an infinite number of injuries (but in a controlled manner which ensures their satisfaction). Various typical examples of such theorems are presented within the framework, like the construction of a high incomplete computably enumerable degree and the jump inversion theorem. A special representative of this class of arguments is the minimal pair construction, which is also presented in this chapter along with the embedding of the pentagon.

Chapter 6 deals with Δ_3 constructions, which are a type of infinite injury argument. The Sacks density theorem (asserting that the structure of the computably enumerable degrees is dense) is presented in the framework. Chapter 7 deals with Σ_3 constructions. The first example of such a construction was Lachlan's non-splitting theorem which was dubbed the *monster theorem* due to its complexity. The example chosen to demonstrate this type of construction in the framework is a curious one. It is the construction of a 'strong minimal pair' or a 'super minimal pair' as it has often been called in the literature. This is a pair of degrees $\mathbf{a}, \mathbf{b}$ such that $\mathbf{a} \not\leq \mathbf{b}$ and the least upper bound of every $\mathbf{z}$ with $\mathbf{0} < \mathbf{z} < \mathbf{a}$ with $\mathbf{b}$ is $\geq \mathbf{a}$. This theorem is attributed to Slaman, but no written proof has appeared until now. It has been discussed by various researchers, including myself, and it plays a role in one of the main open problems in the computably enumerable degrees: whether the 2-quantifier theory is decidable. I have attempted to prove it myself in the past, but found it quite tricky. The account given in Chapter 7 (within the framework) is impressively short. However (having not studied the framework sufficiently) I can not get an impression as to how the author deals with various obstacles one encounters in a standard attempt to prove it.

Chapter 7 concludes with various reflective remarks on the framework. The author conjectures that any priority argument proof of properties of the computably enumerable degrees can be presented in the framework. On the other hand he admits that in certain cases (like the embedding of the 1-3-1 lattice) the proof using the framework may be more cumbersome.

Chapters 8 and 9 are devoted to the proof of the various framework lemmas that are used in the proofs that we just discussed. They are quite technical and require considerable effort to read. Finally in Chapter 10 the author discusses higher level priority arguments, which are really the type of arguments that seem to require a formal framework.

In conclusion, this is an impressive technical piece of work. It is hard to say how influential it is going to be in the future, but as of today most theorems in the computably enumerable degrees are proved without reference to a formal framework. Having said that, higher order priority arguments that refer to some underlying inductive structure (like the jump hierarchy) seem to require a more organized and formal approach, making them perfect candidates for this framework. The value of this approach will ultimately depend on the new theorems that are proved using it.

George Barmpalias

State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, People's Republic of China. barmpalias@gmail.com.

Entry for the Table of Contents: