BOOK REVIEW

Richard L. Epstein, *Propositional Logics: The Semantic Foundations of Logic*, 2nd Edition (Wadsworth, 2001), 525pp. ISBN 0-534-55847X

Richard L. Epstein, *Predicate Logic: The Semantic Foundations of Logic* (Wadsworth, 2001), 480pp. ISBN 0-534-558461.

Richard L. Epstein and Walter A. Carnielli, *Computability: Computable Functions, Logic, and the Foundations of Mathematics*, 2nd Edition, (Wadsworth, 2000), 299pp. ISBN 0-534-54644-7.

SIMON THOMPSON

Computing Lab
University of Kent
CANTERBURY CT2 7NF
UNITED KINGDOM

S.J.Thompson@ukc.ac.uk

The crisis in the foundations of mathematics that began in the late ninetheenth century led, in the first half of the last century, to an extraordinary burst of work in the mathematical analysis of logic and computability. This opened up the field of mathematical logic and at the same time solved many of the central problems that were identified: for example, Gödel showed the essential incompleteness of axiomatisations of arithmetic and Turing proved that the halting problem was unsolvable.

This body of work is of interest to many: mathematical logicians, for whom it forms the basis of their field; philosophers, to whom it presents a set of tools for understanding the foundations of mathematics; and theoretical computer scientists, who use logics to understand and describe the behaviour of computations. Moreover, it is the computer scientist who uses formal logic in practice; two examples follow. Logics may be used to describe the behaviour of computer hardware, and the remarkably efficient *model checking* decision procedures [1] are used to verify properties of chips with of the order of 10^{128} states. Logics can be implemented, and substantial libraries of fully formal proofs in mathematics and computing have been constructed using proof assistants such as HOL and Coq [2,3].

All three of the books under review give introductions to parts of the body of standard material. Epstein's *Propositional Logics* also has a research focus in presenting a unified semantics for a wide variety of propositional logics, modal, intuitionistic, paraconsistent and so on. The rest of the review will look at the books in turn after some general comments on them all.

STYLE AND CONTENTS

A student of mathematics is likely to learn the 'mechanics' of logic and computability theory as a sequence of definitions, lemmas and proofs. A philosopher would expect to analyse the reasoning underlying the definitions, and in the words of Epstein's preface to *Propositional Logic* she may well ask questions like 'if logic is the right way to reason, why are there so many logics?'. The books combine these approaches, and thus offer introductions that could be used either by mathematically inclined philosophers or by mathematicians with an interest in the philosophical roots of their subject. The material is also put into a historical context by extensive quotation from some of the seminal papers in the field, including Hilbert's "On the Infinite", Turing's *Entscheidungsproblem* paper, and numerous quotes from Gödel. On the whole these quotations are integrated into the narrative; someone wishing to read more of the contemporary material could consult van Heijenoort's [4].

Each of the books gives a wide variety of examples. In the logic texts many natural language sentences are given a formal rendering, and these examples are used to illustrate some of the difficulties of relating formal logics and natural language, most notably in the use of material implication to denote logical consequence. Without this discussion of the relationship between formalisation and the definition of logics, the translation from natural language to formal logic can raise more questions than it answers; it was this insight that motivated the development of Tarksi's world – an artificial, computer-implemented 'blocks world' – as used in the teaching text [5]. In *Computability* there are many examples of programs of various sorts: for Turing machines, for primitive recursion and for general recursion.

The authors' style is clear and approachable, and the books carry comprehensive indices of examples and appendices summarising major points. They also contain substantial numbers of exercises that range from the routine to the challenging.

PROPOSITIONAL LOGICS (Richard L. Epstein)

Propositional Logics serves two purposes. Its primary purpose is to cover a wide variety of propositional logics, including classical, intuitionistic, modal, paraconsistent, dependent, relatedness and many-valued logics. For each logic the semantics and proof theory are introduced, an extensive collection of examples is given, and the appropriate soundness and completeness theorems are proved. In this role the text succeeds admirably: the examples give the reader a gentle introduction to the logic as well as identifying cases in which the logical notions correspond only approximately to the concepts of natural language. Judiciously chosen quotations also help the reader to assess the impact and applicability of each logic.

The standard of presentation is strong and engaging, yet it has a few drawbacks.

- The semantics is first presented as the composition of two mappings: the first takes propositional variables to natural language propositions such as 'Simon is bald' which are then mapped to truth values (true in this case!). It is not clear that this two-stage process, which can be confusing to a naïve reader, adds anything to the exposition.
- In some proofs (e.g. Lemma 10 on page 72), many of the cases are left to the reader. In the early stages it would be useful to have more details given to help the student.
- It is notable that the text is almost silent on the subject of building proofs. After introducing the Hilbert-style axiomatisation of propositional logic, PC, on page 70, the only proofs constructed are those which are needed in the proof of the deduction theorem! A computer scientist intent on formalising some mathematics in a theorem prover might be disappointed to find no advice about how to go about building a proof of a particular proposition.
- A general introduction to propositional logic would be enhanced by a discussion of semantic tableaux. A tableau provides an intuitive mechanism for finding valuations satisfying formulas; a formula is shown to be valid by showing that its negation is unsatisfiable. A tableau also provides a systematic way of searching for counterexamples; the discussion on page 59 hints at this, without providing a completely satisfactory explanation of how it is done. Tableaux can also provide a unifying treatment of various logics: decision procedures for temporal logics, for instance, are based on tableaux, and this gives a link to work in model checking [1] discussed earlier.

Despite these particular drawbacks, the book presents a thorough treatment of the semantics and axiomatisation of a wide variety of propositional logics. We now discuss the general semantic approach.

A GENERAL SEMANTIC FRAMEWORK?

Propositional Logics also gives what is intended to be a general semantic framework for all propositional logics. Its founding premise is that in each of the logics a proposition is 'a written or uttered declarative sentence that we agree to view as being either true or false (but not both)' (p 128). To the reviewer this is by no means clear. For example, to a constructivist what is the status of the Goldbach conjecture: that every even natural number is the sum of two primes? It is a statement which has neither proof nor counterexample, and so to a constructivist it could not be deemed either true or false.

It is useful to unpack the quotation and to see it as expressing a statement about certain sorts of logical *judgement*. We can read the statement as asserting that the judgement

A <u>is true</u>

has a certain property: namely that in every case the judgement will hold of A or of $\neg A$ but not both. Of course, this assertion is effectively the law of the excluded middle, but at a *meta*-level. A constructivist would reject the judgement '… <u>is true</u>' as meaningless, but would accept the decidability of a judgement like

p <u>is a proof of</u> A

which can be established in a deductive system like Martin-Löf's [6]. Returning to the example of the Goldbach conjecture, it is clear that for no known (representation of a) proof p do we have

p <u>is a proof of</u> Goldbach conjecture

Using the terminology of judgements, a standard proof rule becomes a rule which establishes a judgement, so that *modus ponens* establishes the judgement B is provable from the judgements A is provable and $A \to B$ is provable.

The terminology of logical judgements comes from an alternative general foundation for logics: the notion of a logical framework. A *logical framework* is a variant of the typed λ–calculus, perhaps with dependent types, in which a formal deductive system can be presented [6,7]. In order to describe the operation of a deductive system it is necessary to formalise some meta-level aspects, including the various sorts of judgements the system encompasses. The motivation for logical frameworks has come from implementers of logics on computers who have sought a sufficiently strong mechanism to describe the variety of deductive logics that might be implemented. The notion of a logical framework has been shown to be robust and of applicable to a range of logics, including the substructural logics discussed below.

Under Epstein's approach – and especially in the quote from page 128 discussed earlier – there is an identification between the object and meta-level. To some degree a logical system can *reflect* aspects of its meta-language, but in general it is both clearer and more powerful to describe a logical system in a framework which can express separately both its object and meta properties.

Setting aside the debate over the founding premise, how does Epstein's general framework work? The meaning of a complex formula such as $A \to B$ is said to be determined by the truth values of A and B together with the (Boolean) value $\mathbb{B}(A,B)$ which measures the *relatedness* of A and B. For the case of implication we have the table:

A	B	$\mathbb{B}(A,B)$	$A \to B$
_	_	F	F
T	T	T	T
T	F	T	F
F	T	T	T
F	F	T	T

that shows in a logic of relatedness a material implication will be false unless the two propositions are related in some way. In some logics, such as a modal logic, the relatedness predicate will be defined with reference to a possible worlds semantics: two formulas will be related if the set of worlds satisfying A is a subset of those satisfying B. A similar analysis relates to the other connectives, but the author makes that valid point that the explanation of implication is central to all the logics he addresses.

The author manages successfully to explain the semantics of all the logics he discusses, but the reviewer is left in some doubt about its universal applicability. For instance, what is the status of the substructural logics [8], such as linear logic [9], in which the structural rules governing assumptions are weakened? To simplify, in linear logic each assumption must be used exactly once: the collection of assumptions in a proof is thus a list (or bag) rather than a set.

A final question is raised by the author's remark on page 298 to the effect that 'I can find no… explication of constructive mathematical content of a proposition which distinguishes $\forall x.P(x,2)$ from $\forall x.\forall y.P(x,y)$'. If one adopts the Curry-Howard interpretation of *propositions as types* [10] the difference is clear: a proof of the former is a one argument function, f say, so that $f(x)$ is a proof of $P(x,2)$ and the latter is a two argument function, g say, so that $G(x,y)$ is a proof of $P(x,y)$. One can construct a proof of $\forall y.\forall x.P(x,y)$ from an arbitrary such g – $\lambda(y,x).g(x,y)$ – but one cannot in general construct such a proof from an arbitrary *f*. It is perhaps a consequence of the author's semantic approach to the various logical systems that this difference is not apparent.

PREDICATE LOGIC (Richard L. Epstein)

This text covers the language of first-order predicate logic and its Tarskian, set-theoretic semantics. Together with the basic material the book covers identity, description operators, functions and various forms of quantifier. The penultimate chapter introduces second-order logic including a large collection of examples that show the enhanced expressive power of quantifying over predicates.

The approach of this volume is resolutely semantic: there is no discussion of proof (or tableaux) for first-order logics. This material is scheduled to appear in the projected volume *Classical Logic*, and the comparison of various predicate logics will be covered in *A General Framework for Semantics of Predicate Logics* (projected). Together with the current volume, these three will have the scope of the single volume *Propositional Logics*.

Predicate Logic gives a standard treatment of the material. The number and utility of examples provided is a real advantage of this over other texts, and as I remarked in the introduction to this review, the book's more 'philosophical' approach has advantages even for the mathematically minded. An example of this is the re-examination of earlier forms of logic – such as Aristotelian syllogism – in the light of formal predicate logic; another is motivation that the book provides from the research literature for the symbolic approach in logic.

COMPUTABILITY (Richard L. Epstein and Walter A. Carnielli)

This is the second edition of an existing textbook which 'confines its changes almost entirely to technical corrections'. It also includes the timeline on *Computability and Undecidability* in the form of a poster (approximately A1). The timeline covers the period 1834

(Babbage) to 1970 (Matiyasevich) and is supplemented by a 28-page commentary and 10-page bibliography. An adventurous teacher might use the timeline as the basis for a course rather than using the more conventional 'rational reconstruction' of the material presented in the body of the text itself.

As far as the text itself is concerned, it covers the material that one would expect of a standard course in computability plus somewhat more advanced material on the elementary functions and the Grzegorczyk hierarchy as well as the weak theory of arithmetic, '*Q*'. Preceding and following this technical material are sections introducing the philosophical context. The first section explores the paradoxes and their resolution; numbers, functions, proof and infinite collections. The final section (IV) looks in detail at various views of Church's thesis and the status of the constructivist approach to mathematics, and concludes with a lively quotation on strict finitism from Isles as well as very clear guidance on further reading.

The discussion is supported by extensive quotation from contemporary papers. One might argue that extensive quotation belongs more in a reader (such as Van Heijenoort [4]) than in a textbook like this, and in places – such as the full quotation of Hilbert's "On the Infinite" – one would have preferred more authorial guidance. On the other hand, an original paper is often fresher and better written, and certainly carries the overtones of real controversy.

Section II deals with Computability. The foundations are introduced with extensive quotes from Hermes and Mal'cev and the author draws the important distinction (attributed to Mostowski) between the semantic, intuitive notion of computability and the syntactic, purely formal idea of computation. Turing machines are then introduced, together with a plethora of example programs, and this leads into a discussion of Church's thesis. The formal treatment of recursion begins with primitive recursion (lots of examples again) which are followed by the elementary functions; the Ackermann function and general recursion;

minimisation and μ-computability. The meta-linguistic approach to computability then becomes the theme, with the introduction of Gödel numbering of programs, the universal computation predicate (which is elementary), the *s-m-n* theorem and so on. A final chapter proves the equivalence of Turing computable and recursive functions.

Section III examines Logic and Arithmetic. It begins with the elements of propositional logic, including its decidability and its axiomatisation. Next, chapter 20 is one of the highlights of the section and indeed the book itself. It consists of a five pages summary of the main elements of the Gödel incompleteness theorems and this gives the reader a very useful roadmap to the remainder of the section, which contains the most technical parts of the book. Various components are combined to give the Gödel theorems:

- arithmetic is formalised in the system *Q*, which is shown to be simpler than the standard first-order axiomatisation (*PA*) due to Peano;
- elementary coding of sequences of numbers uses the Chinese Remainder Theorem from number theory; this is necessary because of the lack of exponentiation in the elementary functions;
- the recursive functions are shown to be precisely those representable in *Q*;
- upon this foundation are proved the undecidability of *Q*, and the first incompleteness theorem. The second is proved informally, with an appeal to [11] for a fully formalised proof.

CONCLUSION

This set of books provides well-paced introductions to propositional logic, to predicate logic and its semantics and to computability theory. Their attention to the philosophical and foundational background of their subjects marks them out from the majority of similar books. The general semantic mechanism of *Propositional Logics* gives a unified view of a variety of semantic approaches, despite the misgivings aired earlier in the review about its universality and philosophical approach.

REFERENCES

[1] CLARKE, E. M. et al., *Model Checking* (MIT Press, 2000).

[2] GORDON, M. and MELHAM, T.F., *Introduction to HOL* (Cambridge University Press, 1993).

[3] CORNES C. *et al.*, *The Coq Proof Assistant Reference Manual* (Rapport Technique, INRIA, 1995).

[4] VAN HEIJENOORT, J. (ed.), *From Frege to Gödel, A Source Book in Mathematical Logic, 1879-1931* (Harvard University Press, 1967).

[5] BARWISE, K.J. and ETCHEMENDY, J., *Language, Truth and Logic* (CSLI Publications and Seven Bridges Press, 1999).

[6] NORDSTRÖM, B., PETERSSON, K. and SMITH J. M., *Programming in Martin-Löf's Type Theory – An Introduction* (Oxford University Press, 1990).

[7] HARPER, R., HONSELL, F. and PLOTKIN, G., *A Framework for Defining Logics*, in *Proceedings of the Symposium on Logic in Computer Science* (IEEE Press, 1987).

[8] SCHROEDER-HEISTER, P. and DOŠEN, K. (eds.), *Substructural Logics* (Oxford University Press, 1993).

[9] TROELSTRA, A.S., *Lectures on Linear Logic* (CSLI Press, 1992).

[10] HOWARD, W.A., *The Formulae-as-Types Notion of Construction*, in *To H. B. Curry: Essays on Combinatory Logic, λ-calculus and Formalism*, Seldin, J.P. and Hindley, J. R. (eds.) (Academic Press, 1980).

[11] SHOENFIELD, J. R., *Mathematical Logic* (Addison-Wesley, 1967).