Book Reviews

N. Shankar
Metamathematics, Machines, and Gödel's Proof. Cambridge University Press. 1994. ISBN 0-521-42027-X. £25.00, 202 pp. hardbound.

Gödel's incompleteness proof must be, after Einstein's $E = mc^2$, the most famous theoretical result of the century, even if it does not reduce to such an elegantly occult epigram. However, the fact that professionals have not provided a one sentence summary has not stopped amateurs, who have précised it as argument for everything from God's existence to the *a priori* impossibility of AI. The latter is, I imagine, responsible for the title, which suggests popular philosophy. But the content documents a purely technical project; philosophy does not appear.

What we really get is a detailed description of the formal statement and proof, using a computer proof checker (the Boyer–Moore theorem prover, 'Nqthm'), of two results in metamathematics: Gödel's first incompleteness theorem and the Church–Rosser theorem for untyped Lambda Calculus. Shankar claims rightly that these are major 'real' results directly relevant to research in mechanical proof. They are also well suited for machine checking, being, even in their informal versions, particularly syntactic. The work described is impressive, both for the scale of the results and the fact that they were obtained using a distinctly ideosyncratic system; he clearly deserved the PhD he got for it. That this is so, however, is not altogether reassuring about the state of the art: Nqthm is one of the most successful systems around, and if people still get PhDs for using it, then much research remains to be done before we can think of mechanical proof as a means to other ends, rather than a research end in itself. Still, progress is being made—the thesis topics are certainly growing more ambitious.

I have just two criticisms. The first is that it reads too much like the thesis it once was: full of signposting, and that careful drawing of conclusions every 30 pages that is intended to reassure examiners that all the right things have been done and the right thoughts have been thought; this does not affect the content, but is tedious and clumsy. The second is more significant, because it does affect the content: whoever produced the final copy forgot to recompile the index, which looks to be for a text of about 255 pages—70 pages more than there are, a problem for the reader of a book containing a couple of hundred formal definitions, and a pity.

So who would be interested? Anyone seriously interested in mechanical proof in general, or Nqathm in particular; but it is not for the faint hearted.

Seán Matthews
Max-Planck-Institut für Informatik

T. F. Melham
Higher Order Logic and Hardware Verification. Cambridge University Press. 1993. ISBN 0 521 41718 X £24.95. 165 pp. hardbound.

The correctness of designs for digital systems are generally 'verified' by simulation, i.e. by computing the response of the system to sample excitations. Simulation has the advantage that it can be performed entirely automatically but it has the disadvantage that, since only an infinitesimal proportion of possible excitations can be examined, many design errors will go undetected.

An alternative to simulation is hardware verification. This relies upon the use of formal reasoning to establish that an implementation satisfies its specification. The starting point is a collection of the behavioural specifications of the components of the implementation and a set of rules for combining these behaviours. This gives a specification which the overall implementation satisfies and it is then up to the verifier to establish that this specification logically implies the desired specification. Since such proofs tend to be lengthy, often running to many thousands (sometimes millions) of applications of primitive inference rules, computation support is used both to help generate the proofs and to guarantee their freedom from logical error.

There are many different species that can be used for hardware verification; this book focusses on the use of 'HOL'—a widely used dialect of higher-order logic having polymorphic types. As the title of the book suggests, two themes are addressed; the HOL logic itself and the principles of hardware verification using HOL.

The HOL logic is distinguished by its simplicity and economy, having only a handful of primitive type constructors, constants and axioms. Other elements of logic (e.g. Cartesian product, numbers, induction) are defined in terms of these primitives. This has the advantage that the logical soundness of the logic is relatively transparent, as also is the correctness of any computational implementation of it.

An early chapter is devoted to an account of HOL. There is, when all the derived forms are taken into account, much ground to cover and parts of this chapter are relatively terse. Whilst a reader familiar with elementary logic and with a polymorphically-typed functional language will have little difficulty, any reader without this background is likely to find this chapter hard going.

The second main theme of the book, hardware verification, is covered comprehensively. The approach, its aims, its methods, and its limitations, are outlined in general terms in Chapter 1. Then, after the HOL logic has been described, successive chapters fill in the details. Much space is, rightly, devoted to a comprehensive