REFERENCES

1. S. Abiteboul and R. Hull. IFO: a formal semantic database model. *ACM Transactions on Database Systems*, **12**(4) 525–565 (1987).
2. P. van Bommel, A. H. M. ter Hofstede and Th.P. van der Weide. Semantics and verification of object-role models. Technical Report 90–13, Department of Information Systems, University of Nijmegen, The Netherlands. Submitted to Information Systems. June 1990.
3. F. Bry and R. Manthey. Checking consistency of database constraints: a logical basis. In *Proceedings of the Twelfth International Conference on Very Large Data Bases*, pp. 13–20 (1986).
4. P. P. Chen. The entity-relationship model: toward a unified view of data. *ACM Transactions on Database Systems*, **1** (1) 9–36 (1976).
5. Index Technology Corporation. *EXCELERATOR*. Index Technology Corporation, One Main Street, Cambridge, M.A. 02142–9990 (1989).
6. P. N. Creasy. A formalisation of the NIAM conceptual Information Model. Department of Computer Science, Australian National University (1983).
7. C. J. Date. *An Introduction to Data Base Systems*. Addison Wesley (1986).
8. T. A. Halpin. *A logical analysis of information systems: static aspects of the data-oriented perspective*. PhD thesis, Department of Computer Science, University of Queensland, Australia (1989).
9. KnowledgeWare. *Information Engineering Workbench*. KnowledgeWare, Inc., 3340 Peachtree Road, N.E., Suite 1100, Atlanta, GA 30026 (1987).
10. C. H. Kung. *A temporal framework for information systems specification and verification*. PhD thesis, University of Trondheim, Norway (1984).
11. D. Maier. *The theory of relational databases*. Computer Science Press, 1988.
12. E. Mendelson. *Introduction to Mathematical Logic*. Wadsworth and Brooks (1987).
13. G. M. Nijssen and T. A. Halpin. *Conceptual schema and Relational Database Design: A fact oriented approach*. Prentice Hall of Australia Pty Ltd (1989).
14. Pandata. *The System Development Workbench*. Pandata B.V., Postbus 1923, 2280 DX Rijswijk, The Netherlands (1988).
15. S. Abiteboul, P. C. Fischer and H. J. Schek (Eds.). *Nested Relations and Complex Objects in Databases*. Springer Verlag (1987).
16. C. Sernadas, J. Fiadeiro, R. Meersman and A. Sernadas. Proof-theoretic conceptual modelling: The NIAM case study. In *Proceedings of the IFIP TC8/WG 8.1 Working Conference on Information System Concepts: An In-depth Analysis*, pp. 1–30 (1989).
17. O. M. F. de Troyer. On rule-based generation of conceptual database updates. In *Proceedings of the IFIP TC2 Working Conference on Knowledge and Data* (1986).
18. O. M. F. de Troyer. RIDL*: A Tool for the Computer-Assisted Engineering of Large Databases in the Presence of Integrity Constraints. Technical report, Institute for Language Technology and Artificial Intelligence ITK, Tilburg University, ITK Research Report No. 3 (1989).
19. G. M. A. Verheijen and J. van Bekkum. NIAM: an Information Analysis Method. In T. W. Olle, H. G. Sol and A. A. Verrijn Stuart, editors, *Information System Design Methodologies: A Comparative Review*, pp. 537–590. North-Holland (1982).
20. T. F. Verhoef. Structuring Yourdon's modern structured analysis. Technical report, Software Engineering Research Centre (SERC), Utrecht, The Netherlands (1991).
21. J. J. V. R. Wintraecken. *Informatie-analyse volgens NIAM*. Academic Service. In Dutch. (1985).

Book Review

GÉRARD HUET and GORDON PLOTKIN
Logical Frameworks, Cambridge University Press. ISBN 0-521-41300-1. £35.00.

Anyone who has followed developments in Formal Methods will have observed the proliferation of logics for reasoning about programs. These include Hoare logics, weakest-precondition logics, process algebras, temporal logics and domain theories. How can all these logics, each evolving seemingly endless variations, receive machine support? For this purpose, logical frameworks were introduced. These were universal calculi that could represent other logics in a uniform manner. A machine implementation of a logical framework would support formal reasoning in many other logics.

Ironically, there is now a proliferation of logical frameworks: the Automath languages, Martin-Löf's framework, the Edinburgh Logical Framework (ELF), intuitionistic higher-order logic, the Calculus of Constructions. Many of these are no longer regarded as universal calculi, but as logics for reasoning about programs. They require machine support; several computer implementations have been built.

This does not mean that we have fruitlessly gone in a circle. Logical frameworks tell us much about the formalisation of abstraction in mathematics, and its connection with abstraction in programming. They are equally at home with general mathematical constructions and computational reasoning.

The ESPRIT Basic Research Action devoted to this area held a major workshop in May 1990. The book *Logical Frameworks*, which appeared sixteen months later, is a refereed selection of the papers presented there. The speed of the editorial process has left quite a few typos, but none of the papers has become obsolete. The book is divided into six parts, such as Implementations, Type Theory and Logical Issues. There is not enough space to discuss all the articles that I found interesting; here are a few of them.

N. G. de Bruijn writes 'A plea for weaker frameworks', based upon his two decades of experience with the Automath project. Weaker frameworks are simpler – easier to learn and to implement. In contrast, Philippe de Groote puts forward a strong framework extended with a formal notion of context, for the purpose of theory structuring. It is a pity that these articles are not presented in the form of a debate; neither cites the other.

David Basin and Matt Kaufmann present a detailed comparison between two well-known systems, the Boyer/Moore Prover and Nuprl. They prove Ramsey's Theorem, which concerns finite graphs, in both systems. Neither comes out as clearly better; each has strengths and weaknesses. As more people start using theorem provers, this sort of comparison will become increasingly necessary.

Amy Felty proves an equivalence in expressive power between two logical frameworks, the ELF and intuitionistic higher-order logic. The latter is implemented by the logic programming language λProlog (and also in my system, Isabelle). Felty's result shows that these systems can support any logic expressible in the ELF.

Because the ELF has certain advantages, such as compactness and explicit proof objects, Frank Pfenning is developing a logic programming language based upon it. His paper details the formidable difficulties and how he is overcoming them. The paper by David Pym and Lincoln Wallen is also concerned with methods for implementing the ELF efficiently.

This is a fascinating book and practically the only source for this material. An alternative is *Logic in Computer Science*, edited by P. Odifreddi, which presents expository articles on various aspects of the subject, and is more accessible for non-specialists. *Logical Frameworks* is at the cutting edge of research, and is correspondingly more difficult for the uninitiated. Most of the articles pack a heavy dose of formalism. Despite this, the book demands consideration by any computer scientist whose work is concerned with logic.

L. PAULSON
Cambridge