Interactive Theorem Proving and Verification

FOREWORD

Research in the area of automated reasoning is largely concentrated around two major themes – Automated Theorem Proving and Interactive Theorem Proving. The goal of Automated Theorem Proving, as the name suggests, is to try to prove a wide range of mathematical theorems using a computer in an automatic fashion. On the other hand, Interactive Theorem Proving tries to achieve similar goals, but in the form of a collaborative effort between human beings and computers. This special issue of *Sādhanā* is concerned with the discipline of Interactive Theorem Proving.

The focus of Interactive Theorem Proving is to present in a fully formalised way, all the axioms, definitions, computations and proofs within any mathematical subject. While the proofs themselves come from humans, the formalisations are meant to be done in such a way that a computer can verify the correctness of the claims. At present there are two major areas of applications of Interactive Theorem Proving. One of them is formalization of mathematics and its proofs, while the other is formal verification of computer systems. The papers in this special issue provide an introduction and overview of research on some of the basic issues of interactive theorem proving and its applications.

The paper by Herman Geuvers provides an overview of the history and state of the art of *proof assistants* which are systems that provide the interface between humans and computers for the purposes of interactive theorem proving. The paper by Gerwin Klein gives a high-level overview of machine-checked software verification in general and the verification of operating systems code in particular. It also provides a comprehensive coverage of various operating system verification projects over the past several years.

The paper by Andrea Asperti *et al* deals with issues in the implementation of the new kernel of *Matita*, which is a compact interactive proof assistant based on a logical formalism called the Calculus of Inductive Constructions. The paper by Eyad Alkassar *et al* discusses and develops the necessary formalisms for proving the correctness of client-server software, modelling all the layers of the implementation stack from instruction set architectures till the remote procedure call layer.

Finally the paper by Freek Wiedijk presents an overview of the process of formalizing a proof of Arrow's theorem, a well-known result from the field of economics. In addition to giving a real feel for the formalization process, it also addresses some of the central questions about the value of such an enterprise, and demonstrates how formalizing may be useful outside mathematics and computer science.

I would like to express my thanks to a number of people whose help was indispensable in making this special issue possible – Gérard Huet and Henk Barendregt for introducing me to this fascinating area of research; Vivek Borkar for encouraging me to edit this special issue; Andrea Asperti, Freek Wiedijk, Gerwin Klein, Herman Geuvers and Wolfgang Paul for readily accepting the invitation to write papers for this special issue; Riki Krishnan and T D Mahabaleswara for their help in preparing this issue for publication; Adam Naumowicz, Conor McBride, Deepak Kapur, K Gopinath, Hendrik Tews, Hugo Herbelin, John Crossley,

John Harrison, Josef Urban, Purandar Bhaduri, Thierry Coquand, and Tobias Nipkow for their timely and ready help in dealing with various aspects of this special issue. I am grateful to all of them.

February 2009

Raja Natarajan
Guest Editor

School of Technology and Computer Science,
Tata Institute of Fundamental Research, Mumbai 400 005
e-mail: raja@tifr.res.in