Special Issue on User Interfaces in Theorem Proving: Preface

David Aspinall (david.aspinall@ed.ac.uk)
LFCS, School of Informatics, The University of Edinburgh, U.K.

Christoph Lüth (christoph.lueth@dfki.de)
Deutsches Forschungszentrum für Künstliche Intelligenz, 28359 Bremen, Germany

Theorem proving is coming of age. While its foundations predate the first computers, and systems have been built since the 1950s, dramatic improvements of proving power and expressivity have been made over the last fifteen years. And like other parts of computing, the complex symbolic manipulations of theorem provers have greatly benefited from the rapid increase in computing power. This means that applications of theorem proving and related methods have now become large, diverse and mature, ranging from real-world hardware and software verification to the formalisation of complex and deep mathematical proofs.

One area that is still very much in need of improvement is interfaces for theorem provers. There is no broad agreement about what makes a good user interface in this area, and little is known about how to bridge the gap between imprecise human interaction and the highly stringent demands of fully formal mathematics. Yet there is universal recognition that interfaces must improve if theorem proving is to become more accessible and productive. Many present systems are unbearably complicated to use and can take months to learn because their interfaces are inadequate for beginning users. Experts are held back too: many interface operations which could significantly enhance productivity are not supported, although they are commonplace in other modern applications.

An example operation is *searching*: instead of carefully structuring data in order to later efficiently retrieve them, nowadays the emphasis has shifted towards flexible and speedy searching in a large shared body of knowledge. Theorem provers ought to provide such search facilities, for theorems, definitions, and proofs. Second, we have the critical question of how proofs are shown to the user: *presentations* of proof should be readily comprehensible to both author and subsequent readers. Finally, there is the issue of the *interaction* itself, taking place between the user and prover to help direct proof construction. Here, there is much scope for graphically oriented interfaces with intuitive gestures to supplement or even replace textual interaction. This seems especially possible in restricted problem domains.

This special issue on *User Interfaces in Theorem Proving* contains five papers reporting progress in this area, addressing the themes outlined above. The first two papers address, inter alia, the problem of searching and sharing knowledge. The paper *User Interaction with the Matita Proof Assistant* by Andrea Asperti *et al.* describes a new theorem prover, Matita, based on the proven Coq technology, but offering distinctive features for user interaction, such as a searchable shared library, and a flexible, sophisticated and web-aware syntax machinery. The paper *Integrating Searching and Authoring in Mizar* by Paul Cairns and Jeremy Gow describes the Alcor system, an interface to Mizar and its Mathematical Library geared particular towards searching and authoring assistance. The next two papers take up the problem of authoring, in particular how to provide an interface which allows users to interact with a theorem prover in most natural (for the user), and hence most productive, way. *A Graphical User Interface for Formal Proofs in Geometry* by Julien Narboux presents an approach to present and author proofs in geometry, by combining the intuitive graphical presentation with an automated theorem prover to check simple facts, and a powerful interactive prover (*viz*, Coq) to check the proofs built manually using the graphical presentation. By restricting the problem domain to geometry a more appropriate user interface can be provided. The paper *An Interface for Student Proof Exercises Using MathsTiles and Isabelle/HOL in an Intelligent Book* by William Billingsley and Peter Robinson describes a simplified graphical way of constructing proofs in an otherwise highly complex theorem prover. Searching is also addressed here, as the system allows the user to ask and search for help. Finally, the paper *Visualizing SAT Instances and Runs of the DPLL Algorithm* by Carsten Sinz shows how appropriate visualisation of solutions of the SAT problem gives new insights into the problem structure. Again, restricting to a specific domain is a key to success.

These five papers are representative of recent research, and indicate that interfaces for theorem provers and related tools is a rich topic for investigation. We would like to thank the authors, and in particular the anonymous referees for the effort they put in to make this special issue a success. We are encouraged by the strong interest in meetings whose scope covers this topic, including the *User Interfaces for Theorem Provers* and *MathUI* workshops, and the *Mathematical Knowledge Management* conference series. The outlook is optimistic, but much more remains to be achieved.

David Aspinall and Christoph Lüth.
Edinburgh and Bremen, March, 2007.