Executive Summary

Franz Baader[1], Peter Baumgartner[2],
Robert Nieuwenhuis[3] and Andrei Voronkov[4]

[1] TU Dresden, Institute for Theoretical Computer Science
Dresden, Germany
baader@tcs.inf.tu-dresden.de
[2] Max-Planck-Institut für Informatik
Saarbrücken, Germany
baumgart@mpi-inf.mpg.de
[3] Technical University of Catalonia, Department LSI
Barcelona, Spain
roberto@lsi.upc.edu
[4] The University of Manchester, Department of Computer Science
Manchester, United Kingdom
voronkov@at cs.man.ac.uk

Abstract. Formal logic provides a mathematical foundation for many areas of computer science. Logical languages are used as specification language within, e.g., program development and verification, hardware design and verification, relational databases, and many subfields of Artificial Intelligence. Automated Deduction is concerned with the design and implementation of algorithms based on logical deduction for solving problems in these areas.
The last years have seen considerable improvements concerning both basic automated deduction technology and its (real-world) applications. Accordingly, the goal of the seminar was to bring together researchers from both sides in order to get an overview of the state of the art, and also to get ideas how to advance automated deduction from an application oriented point of view.

Keywords. Formal Logic, Deduction, Artificial Intelligence

1 Motivation

Formal logic provides a mathematical foundation for many areas of computer science, including problem specification, program development, transformation and verification, hardware design and verification, relational databases, knowledge engineering, theorem proving, computer algebra, logic programming, and artificial intelligence.

Using computers to solving problems in these areas, therefore, requires the design and implementation of algorithms based on logical deduction. It remains one of the great challenges in informatics to make computers capable of doing non-trivial logical reasoning, be it fully automatic, or in interaction with humans. Some progress, however, has been made in the past ten years:

Dagstuhl Seminar Proceedings 05431
Deduction and Applications
http://drops.dagstuhl.de/opus/volltexte/2006/510

- Automated theorem provers and finite model building programs solved various open mathematical problems of combinatorial nature.
- Model checking, a form of theorem proving over finite models, has become a very successful push-button method for verifying nontrivial safety properties of hardware and software.
- Automated deduction, in particular for so-called description logics, is widely assessed as a core enabling technology for the *Semantic Web*.
- Methods of interactive theorem proving have helped in formally verifying semantic (type) safety aspects of programming languages such as Java. The "Schwerpunktprogramm *Deduktion*" funded by the Deutsche Forschungsgemeinschaft together with previous Dagstuhl seminars on "Deduction" have been instrumental in obtaining these successes.

The conviction that mathematical logic is a unifying principle in computer science and that methods from different theoretical areas as well as application domains should be brought together as a means to fight fragmentation has lead to successful new conferences like FLoC and IJCAR, and to IFCoLoG, the recently established International Federation for Computational Logic.

This interdisciplinary view of logic in computer science motivated the Dagstuhl seminar. Specifically, we considered several application areas:

> Software verification — Hardware verification — Cryptographic protocols — Programming languages — Formal methods — Semantic Web — Large knowledge bases.

2 Participants and Program

In total we had 64 participants, mostly from Europe, but also from USA and Australia. A good balance between more senior and junior participants was maintained.

As said above, the seminar was motivated by an interdisciplinary view of logic in computer science. In the seminar we thus invited deduction-oriented researchers from a broader range of applications. This helped to get a (selective) overview of the state of the art concerning deduction in their areas of expertise.

The application areas best represented at the seminar (in terms of number of talks held) concern various forms of "verification". Furthermore, it became clear that the integration of theory specific reasoners, in particular decision procedures, into a core general purpose verification environment and the flexible and semantically well-founded combination of such reasoners are extremely important in many applications of verification tools. This shows that the confluence between application-driven approaches and the foundational research in these areas is now paying off, but these applications also pose new research problems both on the foundational and the practical side, of which interpolation modulo theories is only one example.

3 Conclusion

The seminar consisted of a full, but not overly loaded program. We left enough time for discussions. The atmosphere we felt to be a productive one, characterized by many discussions, both on-line during the talks and off-line. Altogether, we perceived the seminar as a very successful one.