In memoriam

Annette Schumann

1959 – 2001

Preface

This volume is a self-contained introduction to interactive proof in higher-order logic (HOL), using the proof assistant Isabelle 2002. Compared with existing Isabelle documentation, it provides a direct route into higher-order logic, which most people prefer these days. It bypasses first-order logic and minimizes discussion of meta-theory. It is written for potential users rather than for our colleagues in the research world.

Another departure from previous documentation is that we describe Markus Wenzel's proof script notation instead of ML tactic scripts. The latter make it easier to introduce new tactics on the fly, but hardly anybody does that. Wenzel's dedicated syntax is elegant, replacing for example eight simplification tactics with a single method, namely *simp*, with associated options.

The book has three parts.

- The first part, **Elementary Techniques**, shows how to model functional programs in higher-order logic. Early examples involve lists and the natural numbers. Most proofs are two steps long, consisting of induction on a chosen variable followed by the *auto* tactic. But even this elementary part covers such advanced topics as nested and mutual recursion.
- The second part, **Logic and Sets**, presents a collection of lower-level tactics that you can use to apply rules selectively. It also describes Isabelle/HOL's treatment of sets, functions, and relations and explains how to define sets inductively. One of the examples concerns the theory of model checking, and another is drawn from a classic textbook on formal languages.
- The third part, **Advanced Material**, describes a variety of other topics. Among these are the real numbers, records, and overloading. Esoteric techniques are described involving induction and recursion. A whole chapter is devoted to an extended example: the verification of a security protocol.

The typesetting relies on Wenzel's theory presentation tools. An annotated source file is run, typesetting the theory in the form of a LaTeX source file. This book is derived almost entirely from output generated in this way. The final chapter of Part I explains how users may produce their own formal documents in a similar fashion.

Isabelle's web site[1] contains links to the download area and to documentation and other information. Most Isabelle sessions are now run from within David Aspinall's wonderful user interface, Proof General[2], even together with the X-Symbol[3] package for XEmacs. This book says very little about Proof General, which has its own documentation. In order to run Isabelle, you will need a Standard ML compiler. We recommend Poly/ML[4], which is free and gives the best performance. The other fully supported compiler is Standard ML of New Jersey[5].

This tutorial owes a lot to the constant discussions with and the valuable feedback from the Isabelle group at Munich: Stefan Berghofer, Olaf Müller, Wolfgang Naraschewski, David von Oheimb, Leonor Prensa Nieto, Cornelia Pusch, Norbert Schirmer, and Martin Strecker. Stephan Merz was also kind enough to read and comment on a draft version. We received comments from Stefano Bistarelli, Gergely Buday, and Tanja Vos.

The research has been funded by many sources, including the DFG grants Ni 491/2, Ni 491/3, Ni 491/4 and the EPSRC grants GR/K57381, GR/K77051, GR/M75440, GR/R01156/01, and by the ESPRIT working groups 21900 and IST-1999-29001 (the *Types* project).

[1] http://isabelle.in.tum.de/

[2] http://www.proofgeneral.org/

[3] http://www.fmi.uni-passau.de/~wedler/x-symbol/

[4] http://www.polyml.org/

[5] http://cm.bell-labs.com/cm/cs/what/smlnj/index.html