TH∃OREM∀: A System for the Working Mathematician

Tudor Jebelean, Bruno Buchberger
RISC-Linz
www.theorema.org

The main goal of the TH∃OREM∀ project[1] is to deliver an integrated interactive environment which can assist the mathematician in all the phases of his scientific work: proving, computing and solving in various mathematical domains. The system is implemented on top of *Mathematica*, thus it is backed by the full algorithmic and computing power of the currently most popular computer algebra system, which is available on all the main computing platforms (Unix, Linux, Windows, and Apple). The current implementation is the result of several man-years of work by many people of the TH∃OREM∀ group at RISC (see `www.theorema.org`), under the direction of Bruno Buchberger. Until now, we already built into the system the main features concerning *proving* and *computing*, while the *solving* features are in the design phase. The main features of the system will be demonstrated live during this presentation.

The system interacts with the user in the language of *predicate logic*, which is the natural language for expressing mathematical properties and algorithms. Few intuitive commands allow the user to enter mathematical formulae (in natural two-dimensional notation) and to compose them into mathematical theories, and also to use some basic domains (numbers, tuples, sets) which are already provided in TH∃OREM∀. Moreover, the system provides the implementation of the powerfull concept of *functor*, which allows the build-up of sophisticated domains on top of simpler ones. The mathematician has the possibility to experiment with the algorithms expressed in this way by directly running them using the TH∃OREM∀ *computing* engine, and he can also study their formal properties (e.g. correctness) using the *provers* of TH∃OREM∀. It is an unique feature of TH∃OREM∀ that these two phases of the mathematical activity can be performed in the *same integrated system* and using the *same language*.

Computation is performed under full control of the user, which means being able to trace the reason (definition, formula) for each computing step – if necessary, but most important with full control of the *knowledge* which is used in the computing process. For instance, in a certain situation the mathematician wants to give to the symbols **, +, 0, Successor* the axiomatic meaning as defined by induction over natural numbers, while in another situation the same symbols should be interpreted using the full power of the underlying computational engine (positional notation, long arithmetic, etc.)

Proving is done with specific methods for several mathematical domains: propositional logic, general predicate logic, induction over integers and over lists, set theory, boolean combinations of polynomial [in]equalities (using Groebner Bases), combinatorial summation (using Paule–Schorn–Zeilberger), and a novel technique for proving in higher-order logic with equality: PCS (proving–computing–solving), introduced by Buchberger. TH∃OREM∀ departs from the methods mostly used in automatic provers today, because it uses a *natural proving style*: the formulae are expressed in their original form (two-dimensional, non-clausal), the inference steps are expressed in natural style and in human language, and – most importantly – the proving methods are similar to the ones which are used by the working mathematicians. Therefore, the user has the possibility to inspect and easily understand the proofs, to verify any inference, and to interact with the proof by modifying certain assumptions, etc.

[1]Supported by Austrian Forschungsförderungsfonds (FWF), project P10002-PHY.