Automating Prover Development for Non-Classical Logics

Renate A. Schmidt∗

Our interest in this talk is automation of prover development and other related tasks for non-classical logics, focussing on modal, description and intuitionistic logics. Developing and implementing a theorem prover from scratch is not straightforward and very time-consuming. There are however general principles for implementing theorem provers and many techniques and optimisations in provers are logic-independent. For automated prover generation it is therefore important to try and distil such general principles and techniques while at the same time exploiting as best as possible logic-specific properties. In an ongoing research project we have developed a tableau prover generator, called Mettel [1, 14, 15]. It takes as input the specification of the syntax of a logic and a set of inference rules defining a deduction calculus for the logic. The output is Java code of a prover that performs inferences in accordance with the specified calculus. Mettel is designed to make it very easy to obtain provers for various logics, including not previously studied logics. In this talk we demonstrate the use of Mettel, we discuss some of its internal workings, and highlight some practical experiences so far from various case studies [2, 4, 5, 6, 7]. Issues and results related to ensuring completeness, blocking to ensure termination [9, 10, 12], automatic rule refinements [13], automatic rule generation [11], and automating correspondence theory [3, 8] will be touched upon.

References


∗This research is supported by UK EPSRC research grant EP/H043748/1.


