Gödel's God on the Computer (Abstract)*

Christoph Benzmüller[1] and Bruno Woltzenlogel Paleo[2]

[1] Freie Universität Berlin, Germany
c.benzmueller@fu-berlin.de
[2] Vienna University of Technology, Austria
bruno@logic.at

Attempts to prove the existence (or non-existence) of God by means of abstract ontological arguments are an old tradition in philosophy and theology. Gödel's proof [12, 13] is a modern culmination of this tradition, following particularly the footsteps of Leibniz. Gödel defines God as a being who possesses all *positive* properties. He does not extensively discuss what positive properties are, but instead he states a few reasonable (but debatable) axioms that they should satisfy. Various slightly different versions of axioms and definitions have been considered by Gödel and by several philosophers who commented on his proof; cf. [17, 2, 11, 1, 10]. We have analyzed Scott's version of Gödel's proof [16] for the first-time with an unprecedented degree of detail and formality with the help of theorem provers; cf. `https://github.com/FormalTheology/GoedelGod`.

The following has been done (and in this order): (i) a detailed natural deduction proof; (ii) a formalization in TPTP THF syntax [18]; (iii) an automatic verification of the consistency of the axioms and definitions with Nitpick [8]; (iv) an automatic demonstration of the theorems with the provers LEO-II [5] and Satallax [9]; (v) a step-by-step formalization using the Coq proof assistant [6]; (vi) a formalization using the Isabelle proof assistant [15], where the theorems have been automated with the Isabelle tools Sledgehammer [7] and Metis [14].

Furthermore, the multi-step THF development of the proof has recently been encoded in Sutcliffe's new TPI scripting language [19]. This scripting language offers important features that were missing in the TPTP world so far. Our collaboration with Sutcliffe has also led to some improvements of the TPI language.

Gödel's proof is challenging to formalize and verify because it requires an expressive logical language with modal operators (*possibly* and *necessarily*) and with quantifiers for individuals and properties. Our computer-assisted formalizations rely on an embedding of the modal logic into classical higher-order logic with Henkin semantics. The formalization is thus essentially done in classical higher-order logic where quantified modal logic is emulated.

In our ongoing computer-assisted study of Gödel's proof, the automated reasoning tools have made some interesting observations, some of which were unknown so far. This attests the maturity of contemporary interactive and automated deduction tools for classical higher-order logic and demonstrates the elegance and practical relevance of the embeddings-based approach. Most importantly, our work opens new perspectives for a computer-assisted theoretical philosophy. The critical discussion of the underlying concepts, definitions and axioms remains a human responsibility, but the computer can assist in building and checking rigorously correct logical arguments. In case of logico-philosophical disputes, the computer can check the disputing arguments and partially fulfill Leibniz' dictum: Calculemus — Let us calculate!

References

[1] R.M. Adams. Introductory note to *1970. In *Kurt Gödel: Collected Works Vol. 3: Unpublished Essays and Letters*. Oxford University Press, 1995.

*This work has been supported by the German Research Foundation under grant BE2501/9-1.

[2] A.C. Anderson and M. Gettings. Gödel ontological proof revisited. In *Gödel'96: Logical Foundations of Mathematics, Computer Science, and Physics: Lecture Notes in Logic 6*. Springer, 1996.

[3] C. Benzmüller and L.C. Paulson. Exploring properties of normal multimodal logics in simple type theory with LEO-II. In *Festschrift in Honor of Peter B. Andrews on His 70th Birthday*, pages 386–406. College Publications.

[4] C. Benzmüller and L.C. Paulson. Quantified multimodal logics in simple type theory. *Logica Universalis (Special Issue on Multimodal Logics)*, 7(1):7–20, 2013.

[5] C. Benzmüller, F. Theiss, L. Paulson, and A. Fietzke. LEO-II - a cooperative automatic theorem prover for higher-order logic. In *Proc. of IJCAR 2008*, volume 5195 of *LNAI*, pages 162–170. Springer, 2008.

[6] Y. Bertot and P. Casteran. *Interactive Theorem Proving and Program Development*. Springer, 2004.

[7] J.C. Blanchette, S. Böhme, and L.C. Paulson. Extending Sledgehammer with SMT solvers. *Journal of Automated Reasoning*, 51(1):109–128, 2013.

[8] J.C. Blanchette and T. Nipkow. Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In *Proc. of ITP 2010*, number 6172 in LNCS, pages 131–146. Springer, 2010.

[9] C.E. Brown. Satallax: An automated higher-order prover. In *Proc. of IJCAR 2012*, number 7364 in LNAI, pages 111 – 117. Springer, 2012.

[10] R. Corazzon. Contemporary bibligraphy on the ontological proof (`http://www.ontology.co/biblio/ontological-proof-contemporary-biblio.htm`).

[11] M. Fitting. *Types, Tableaux and Gödel's God*. Kluver Academic Press, 2002.

[12] K. Gödel. Ontological proof. In *Kurt Gödel: Collected Works Vol. 3: Unpublished Essays and Letters*. Oxford University Press, 1970.

[13] K. Gödel. *Appendix A. Notes in Kurt Gödel's Hand*, pages 144–145. In [17], 2004.

[14] J. Hurd. First-order proof tactics in higher-order logic theorem provers. In *Design and Application of Strategies/Tactics in Higher Order Logics, NASA Tech. Rep. NASA/CP-2003-212448*, 2003.

[15] T. Nipkow, L.C. Paulson, and M. Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Number 2283 in LNCS. Springer, 2002.

[16] D. Scott. *Appendix B. Notes in Dana Scott's Hand*, pages 145–146. In [17], 2004.

[17] J.H. Sobel. *Logic and Theism: Arguments for and Against Beliefs in God*. Cambridge U. Press, 2004.

[18] G. Sutcliffe and C. Benzmüller. Automated reasoning in higher-order logic using the TPTP THF infrastructure. *Journal of Formalized Reasoning*, 3(1):1–27, 2010.

[19] G. Sutcliffe. The TPTP TPI Scripting Language (check!). *Proc. of IWIL-10*, 2013.