ECAI 2014
T. Schaub et al. (Eds.)
© 2014 The Authors and IOS Press.

doi:10.3233/978-1-61499-419-0-93

Automating Gödel's Ontological Proof of God's Existence with Higher-order Automated Theorem Provers

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

Abstract. Kurt Gödel's ontological argument for God's existence has been formalized and automated on a computer with higher-order automated theorem provers. From Gödel's premises, the computer proved: necessarily, there exists God. On the other hand, the theorem provers have also confirmed prominent criticism on Gödel's ontological argument, and they found some new results about it.

The background theory of the work presented here offers a novel perspective towards a *computational theoretical philosophy*.

1 INTRODUCTION

Kurt Gödel proposed an argumentation formalism to prove the existence of God [23, 30]. Attempts to prove the existence (or non-existence) of God by means of abstract, ontological arguments are an old tradition in western philosophy. Before Gödel, several prominent philosophers, including St. Anselm of Canterbury, Descartes and Leibniz, have presented similar arguments. Moreover, there is an impressive body of recent and ongoing work (cf. [31, 19, 18] and the references therein). Ontological arguments, for or against the existence of God, illustrate well an essential aspect of metaphysics: some (necessary) facts for our existing world are deduced by purely a priori, analytical means from some abstract definitions and axioms.

What motivated Gödel as a logician was the question, whether it is possible to deduce the existence of God from a small number of foundational (but debatable) axioms and definitions, with a mathematically precise, formal argumentation chain in a well defined logic.

In theoretical philosophy, formal logical confrontations with such ontological arguments had been so far (mainly) limited to paper and pen. Up to now, the use of computers was prevented, because the logics of the available theorem proving systems were not expressive enough to formalize the abstract concepts adequately. Gödel's proof uses, for example, a complex higher-order modal logic (HOML) to handle concepts such as *possibility* and *necessity* and to support quantification over individuals and properties.

Current works [10, 9] of the first author and Paulson illustrate that many expressive logics, including quantified (multi-)modal logics, can be embedded into the classical higher-order logic (HOL), which can thus be seen as a universal logic [6]. For this universal logic, efficient automated theorem provers have been developed in recent years, and these systems were now employed in our work.

Gödel defines God (see Fig. 1) 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. [31, 3, 2, 19, 1, 18]).

A1 Either a property or its negation is positive, but not both: $\forall\phi[P(\neg\phi) \equiv \neg P(\phi)]$

A2 A property necessarily implied by a positive property is positive: $\forall\phi\forall\psi[(P(\phi) \wedge \Box\forall x[\phi(x) \supset \psi(x)]) \supset P(\psi)]$

T1 Positive properties are possibly exemplified: $\forall\phi[P(\phi) \supset \Diamond\exists x\phi(x)]$

D1 A *God-like* being possesses all positive properties: $G(x) \equiv \forall\phi[P(\phi) \supset \phi(x)]$

A3 The property of being God-like is positive: $P(G)$

C Possibly, God exists: $\Diamond\exists x G(x)$

A4 Positive properties are necessarily positive: $\forall\phi[P(\phi) \supset \Box\ P(\phi)]$

D2 An *essence* of an individual is a property possessed by it and necessarily implying any of its properties: $\phi\ ess.\ x \equiv \phi(x) \wedge \forall\psi(\psi(x) \supset \Box\forall y(\phi(y) \supset \psi(y)))$

T2 Being God-like is an essence of any God-like being: $\forall x[G(x) \supset G\ ess.\ x]$

D3 *Necessary existence* of an individ. is the necessary exemplification of all its essences: $NE(x) \equiv \forall\phi[\phi\ ess.\ x \supset \Box\exists y\phi(y)]$

A5 Necessary existence is a positive property: $P(NE)$

T3 Necessarily, God exists: $\Box\exists x G(x)$

Figure 1. Scott's version of Gödel's ontological argument [30].

The overall idea of Gödel's proof is in the tradition of Anselm's argument, who defined God as some entity of which nothing greater can be conceived. Anselm argued that existence in the actual world would make such an assumed being even greater; hence, by definition God must exist. Gödel's ontological argument is clearly related to this reasoning pattern. However, it also tries to fix some fundamental weaknesses in Anselm's work. For example, Gödel explicitly proves that God's existence is possible, which has been a basic assumption of Anselm. Because of this, Anselm's argument has been criticized as incomplete by Leibniz. Leibniz instead claimed that the assumption should be derivable from the definition of God as a perfect being and from the notion of perfection. Gödel's proof addresses this critique, and it also addresses the critique of others, including Kant's objection that existence should not be treated as a predicate. On the other hand, Gödel's work still leaves room for criticism, in particular, his axioms are so strong that they imply *modal collapse*, that is, a situation where contingent truths and necessary truths coincide. More information on the philosophical debate on Gödel's proof is provided in [31].

[1] Freie Universität Berlin, Germany, email: c.benzmueller@fu-berlin.de; this author has been supported by the German National Research Foundation (DFG) under grants BE 2501/9-1 and BE 2501/11-1.

[2] Technical University Vienna, Austria, email: bruno@logic.at

We have analyzed Dana Scott's version of Gödel's proof [30] (cf. Fig. 1) for the first-time with an unprecedented degree of detail and formality with the help of higher-order automated theorem provers (HOL-ATPs).[3] The following has been done (and in this order): (i) a detailed natural deduction proof; (ii) a formalization in TPTP THF syntax [33]; (iii) an automatic verification of the consistency of the axioms and definitions with Nitpick [16]; (iv) an automatic demonstration of the theorems with the provers LEO-II [11] and Satallax [17]; (v) a step-by-step formalization using the Coq proof assistant [15]; (vi) a formalization using the Isabelle proof assistant [26], where the theorems (and some additional lemmata) have been automated with the Isabelle tools Sledgehammer and Metis. Subsequently, we have studied additional consequences of Gödel's axioms, including modal collapse and monotheism, and we have investigated variations of the proof, for example, by switching from constant domain semantics to varying domain semantics.

In this paper we focus on the core aspect of our work related to AI: proof automation with HOL-ATPs (cf. aspects (ii)–(iv) above). The particular contributions of this paper are as follows: In Sec. 2 we present an elegant embedding of HOML [21, 25] in HOL [5, 7]. This background theory extends and adapts the work as presented in [9, 10]. In Sec. 3, we present details on the encoding of this embedding and of Gödel's argument in the concrete THF syntax [33] for HOL, and we report on the experiments we have conducted with HOL-ATPs. The main findings of these experiments are summarized in Sec. 4. Related and future work is addressed in Sec. 5, and the paper is concluded in Sec. 6. None of the above aspects have been addressed (at least not in depth) in any of our other existing (short and partly non-reviewed) publications on the subject [12, 13, 14, 34].

2 THEORY FRAMEWORK

An embedding of quantified modal logic (with first-order and propositional quantifiers) in HOL has been presented in [10]. The theory below extends this work: quantifiers for all types are now supported, and nested uninterpreted predicate and function symbols of arbitrary types are allowed as opposed to allowing top-level uninterpreted predicate symbols over individual variables only.

2.1 Higher-order modal logic

A notion of HOML is introduced that extends HOL with a modal operator $\Box$. An appropriate notion of semantics for HOML is obtained by adapting Henkin semantics for HOL (cf. [24] and [21]). The presentation in this section is adapted from [25] and [5].

Def. 1 *The set T of* simple types *is freely generated from the set of basic types $\{o, \mu\}$ (o stands for Booleans and μ for individuals) using the function type constructor $\rightarrow$. We may avoid parentheses, and $\alpha \rightarrow \alpha \rightarrow \alpha$ then stands for $(\alpha \rightarrow (\alpha \rightarrow \alpha))$, that is, function types associate to the right.*

Def. 2 *The* grammar *for HOML is:*

$$\begin{aligned} s, t \quad ::= \quad & p_\alpha \mid X_\alpha \mid (\lambda X_\alpha \text{.}\, s_\beta)_{\alpha \rightarrow \beta} \mid (s_{\alpha \rightarrow \beta}\, t_\alpha)_\beta \mid (\neg_{o \rightarrow o}\, s_o)_o \mid \\ & ((\vee_{o \rightarrow o \rightarrow o} s_o)\, t_o)_o \mid (\forall_{(\alpha \rightarrow o) \rightarrow o}(\lambda X_\alpha \text{.}\, s_o))_o \mid (\Box_{o \rightarrow o}\, s_o)_o \end{aligned}$$

where $\alpha, \beta \in T$. p_α denotes typed constants and X_α typed variables (distinct from p_α). Complex typed terms are constructed via abstraction and application. The type of each term is given as a subscript. Terms s_o of type o are called formulas. The logical connectives of choice are $\neg_{o \rightarrow o}$, $\vee_{o \rightarrow o \rightarrow o}$, $\forall_{(\alpha \rightarrow o) \rightarrow o}$ (for $\alpha \in T$), and $\Box_{o \rightarrow o}$. Type subscripts may be dropped if irrelevant or obvious. Similarly, parentheses may be avoided. Binder notation $\forall X_\alpha \text{.}\, s_o$ is used as shorthand for $\forall_{(\alpha \rightarrow o) \rightarrow o}(\lambda X_\alpha \text{.}\, s_o)$, and infix notation $s \vee t$ is employed instead of $((\vee s)\, t)$. From the above connectives, other logical connectives, such as $\top$, $\bot$, $\wedge$, $\supset$, $\equiv$, $\exists$, and $\Diamond$, can be defined in the usual way.

Def. 3 Substitution *of a term A_α for a variable X_α in a term B_β is denoted by $[A/X]B$. Since we consider α-conversion implicitly, we assume the bound variables of B avoid variable capture.*

Def. 4 *Two common relations on terms are given by β-reduction and η-reduction. A β-redex has the form $(\lambda X \text{.}\, s)t$ and β-reduces to $[t/X]s$. An η-redex has the form $(\lambda X \text{.}\, sX)$ where variable X is not free in s; it η-reduces to s. We write $s =_\beta t$ to mean s can be converted to t by a series of β-reductions and expansions. Similarly, $s =_{\beta\eta} t$ means s can be converted to t using both β and η. For each $s_\alpha \in HOML$ there is a unique β-*normal form *and a unique $\beta\eta$-*normal form.

Def. 5 *A* frame *D is a collection $\{D_\alpha\}_{\alpha \in T}$ of nonempty sets D_α, such that $D_o = \{T, F\}$ (for truth and falsehood). The $D_{\alpha \rightarrow \beta}$ are collections of functions mapping D_α into D_β.*

Def. 6 *A* variable assignment *g maps variables X_α to elements in D_α. $g[d/W]$ denotes the assignment that is identical to g, except for variable W, which is now mapped to d.*

Def. 7 *A* model *for HOML is a quadruple $M = \langle W, R, D, \{I_w\}_{w \in W} \rangle$, where W is a set of worlds (or states), R is an accessibility relation between the worlds in W, D is a frame, and for each $w \in W$, $\{I_w\}_{w \in W}$ is a family of typed interpretation functions mapping constant symbols p_α to appropriate elements of D_α, called the* denotation of *p_α* in *world w (the logical connectives $\neg$, $\vee$, $\forall$, and $\Box$ are always given the standard denotations, see below). Moreover, it is assumed that the domains $D_{\alpha \rightarrow \alpha \rightarrow o}$ contain the respective identity relations on objects of type α (to overcome the extensionality issue discussed in [4]).*

Def. 8 *The* value *$\|s_\alpha\|^{M,g,w}$ of a HOML term s_α on a model $M = \langle W, R, D, \{I_w\}_{w \in W} \rangle$ in a world $w \in W$ under variable assignment g is an element $d \in D_\alpha$ defined in the following way:*

1. *$\|p_\alpha\|^{M,g,w} = I_w(p_\alpha)$ and $\|X_\alpha\|^{M,g,w} = g(X_\alpha)$*
2. *$\|(s_{\alpha \rightarrow \beta}\, t_\alpha)_\beta\|^{M,g,w} = \|s_{\alpha \rightarrow \beta}\|^{M,g,w}(\|t_\alpha\|^{M,g,w})$*
3. *$\|(\lambda X_\alpha \text{.}\, s_\beta)_{\alpha \rightarrow \beta}\|^{M,g,w}$ = the function f from D_α to D_β such that $f(d) = \|s_\beta\|^{M,g[d/X_\alpha],w}$ for all $d \in D_\alpha$*
4. *$\|(\neg_{o \rightarrow o}\, s_o)_o\|^{M,g,w} = T$ iff $\|s_o\|^{M,g,w} = F$*
5. *$\|((\vee_{o \rightarrow o \rightarrow o}\, s_o)\, t_o)_o\|^{M,g,w} = T$ iff $\|s_o\|^{M,g,w} = T$ or $\|t_o\|^{M,g,w} = T$*
6. *$\|(\forall_{(\alpha \rightarrow o) \rightarrow o}(\lambda X_\alpha \text{.}\, s_o))_o\|^{M,g,w} = T$ iff for all $d \in D_\alpha$ we have $\|s_o\|^{M,g[d/X_\alpha],w} = T$*
7. *$\|(\Box_{o \rightarrow o}\, s_o)_o\|^{M,g,w} = T$ iff for all $v \in W$ with wRv we have $\|s_o\|^{M,g,v} = T$*

Def. 9 *A model $M = \langle W, R, D, \{I_w\}_{w \in W} \rangle$ is called a* standard model *iff for all $\alpha, \beta \in T$ we have $D_{\alpha \rightarrow \beta} = \{f \mid f : D_\alpha \longrightarrow D_\beta\}$. In a* Henkin model *function spaces are not necessarily full. Instead it is only required that $D_{\alpha \rightarrow \beta} \subseteq \{f \mid f : D_\alpha \longrightarrow D_\beta\}$ (for all $\alpha, \beta \in T$) and that the valuation function $\|\cdot\|^{M,g,w}$ from above is total (i.e., every term denotes). Any standard model is obviously also a Henkin model. We consider Henkin models in the remainder.*

[3] All sources of our formalization are publicly available at `https://github.com/FormalTheology/GoedelGod`. Our work has attracted major public interest, and leading media institutions worldwide have reported on it; some exemplary links to respective media reports and interviews are available at the above URL (see 'Press' subfolder).

Def. 10 *A formula s_o is* true *in model M for world w under assignment g iff $\|s_o\|^{M,g,w} = T$; this is also denoted as $M, g, w \models s_o$. A formula s_o is called* valid *in M iff $M, g, w \models s_o$ for all $w \in W$ and all assignments g. Finally, a formula s_o is called* valid, *which we denote by $\models s_o$, iff s_o is valid for all M. Moreover, we write $\Gamma \models \Delta$ (for sets of formulas Γ and Δ) iff there is a model $M = \langle W, R, D, \{I_w\}_{w \in W}\rangle$, an assignment g, and a world $w \in W$, such that $M, g, w \models s_o$ for all $s_o \in \Gamma$ and $M, g, w \models t_o$ for at least one $t_o \in \Delta$.*

The above definitions introduce higher-order modal logic K. In order to obtain logics KB and S5 respective conditions on accessibility relation R are postulated: R is a symmetric relation in logic KB, and it is an equivalence relation in logic S5. If these restriction apply, we use the notations $\models^{KB}$ and $\models^{S5}$. Gödel's argument has been developed and studied in the context of logic S5 (and logic S5 has subsequently been criticized). However, the HOL-ATPs discovered (cf. Sec. 4) that logic KB is sufficient.

An important issue for quantified modal logics is whether constant domain or varying domain semantics is considered. The theory above introduces constant domains. Terms (other than those of Boolean type) are modeled as rigid, that is, their denotation is fixed for all worlds. An adaptation to varying or cumulative domains is straightforward (cf. [20]). Moreover, non-rigid terms could be modeled; that is, terms whose denotation may switch from world to world. The respective assumptions of Gödel are not obvious to us.

2.2 Classical higher-order logic

HOL is easily obtained from HOML by removing the modal operator $\Box$ from the grammar, and by dropping the set of possible worlds W and the accessibility relation R from the definition of a model. Nevertheless, we explicitly state the most relevant definitions for the particular notion of HOL as employed in this paper. One reason is that we do want to carefully distinguish the HOL and HOML languages in the remainder (we use boldface fonts for HOL and standard fonts for HOML). There is also a subtle, but harmless, difference in the HOL language as employed here in comparison to the standard presentation: here three base types are employed, whereas usually only two base types are considered. The third base type plays a crucial role in our embedding of HOML in HOL.

Def. 11 *The set $\boldsymbol{T}$ of simple types freely generated from a set of basic types $\{\boldsymbol{o}, \boldsymbol{\mu}, \boldsymbol{\iota}\}$ using the function type constructor $\rightarrow$. $\boldsymbol{o}$ is the type of Booleans, $\boldsymbol{\mu}$ is the type of individuals, and type $\boldsymbol{\iota}$ is employed as the type of possible worlds below. As before we may avoid parentheses.*

Def. 12 *The grammar for higher-order logic HOL is:*

$$\begin{aligned} \boldsymbol{s}, \boldsymbol{t} \quad ::= \quad & \boldsymbol{p}_\alpha \mid \boldsymbol{X}_\alpha \mid (\boldsymbol{\lambda X}_\alpha . \boldsymbol{s}_\beta)_{\alpha\rightarrow\beta} \mid (\boldsymbol{s}_{\alpha\rightarrow\beta}\, \boldsymbol{t}_\alpha)_\beta \mid \boldsymbol{\neg}_{o\rightarrow o}\, \boldsymbol{s}_o \mid \\ & ((\boldsymbol{\vee}_{o\rightarrow o\rightarrow o}\, \boldsymbol{s}_o)\, \boldsymbol{t}_o) \mid \boldsymbol{\forall}_{(\alpha\rightarrow o)\rightarrow o}(\boldsymbol{\lambda X}_\alpha . \boldsymbol{s}_o) \end{aligned}$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \boldsymbol{T}$. The text from Def. 2 analogously applies, except that we do not consider the modal connectives $\Box$ and $\Diamond$.

The definitions for substitution (Def. 3), β- and η-reduction (Def. 4), frame (Def. 5), and assignment (Def. 6) remain unchanged.

Def. 13 *A* model *for HOL is a tuple $\boldsymbol{M} = \langle \boldsymbol{D}, \boldsymbol{I} \rangle$, where $\boldsymbol{D}$ is a frame, and $\boldsymbol{I}$ is a family of typed interpretation functions mapping constant symbols $\boldsymbol{p}_\alpha$ to appropriate elements of $\boldsymbol{D}_\alpha$, called the* denotation of *$\boldsymbol{p}_\alpha$ (the logical connectives $\boldsymbol{\neg}$, $\boldsymbol{\vee}$, and $\boldsymbol{\forall}$ are always given the standard denotations, see below). Moreover, we assume that the domains $\boldsymbol{D}_{\alpha\rightarrow\alpha\rightarrow o}$ contain the respective identity relations.*

Def. 14 *The* value $\|\boldsymbol{s}_\alpha\|^{\boldsymbol{M},\boldsymbol{g}}$ *of a HOL term $\boldsymbol{s}_\alpha$ on a model $\boldsymbol{M} = \langle \boldsymbol{D}, \boldsymbol{I} \rangle$ under assignment $\boldsymbol{g}$ is an element $\boldsymbol{d} \in \boldsymbol{D}_\alpha$ defined in the following way:*

1. $\|\boldsymbol{p}_\alpha\|^{\boldsymbol{M},\boldsymbol{g}} = \boldsymbol{I}(\boldsymbol{p}_\alpha)$ *and* $\|\boldsymbol{X}_\alpha\|^{\boldsymbol{M},\boldsymbol{g}} = \boldsymbol{g}(\boldsymbol{X}_\alpha)$
2. $\|(\boldsymbol{s}_{\alpha\rightarrow\beta}\, \boldsymbol{t}_\alpha)_\beta\|^{\boldsymbol{M},\boldsymbol{g}} = \|\boldsymbol{s}_{\alpha\rightarrow\beta}\|^{\boldsymbol{M},\boldsymbol{g}}(\|\boldsymbol{t}_\alpha\|^{\boldsymbol{M},\boldsymbol{g}})$
3. $\|(\boldsymbol{\lambda X}_\alpha . \boldsymbol{s}_\beta)_{\alpha\rightarrow\beta}\|^{\boldsymbol{M},\boldsymbol{g}}$ *= the function $\boldsymbol{f}$ from $\boldsymbol{D}_\alpha$ to $\boldsymbol{D}_\beta$ such that $\boldsymbol{f}(\boldsymbol{d}) = \|\boldsymbol{s}_\beta\|^{\boldsymbol{M},\boldsymbol{g}[\boldsymbol{d}/\boldsymbol{X}_\alpha]}$ for all $\boldsymbol{d} \in \boldsymbol{D}_\alpha$*
4. $\|(\boldsymbol{\neg}_{o\rightarrow o}\, \boldsymbol{s}_o)_o\|^{\boldsymbol{M},\boldsymbol{g}} = \boldsymbol{T}$ *iff* $\|\boldsymbol{s}_o\|^{\boldsymbol{M},\boldsymbol{g}} = \boldsymbol{F}$
5. $\|((\boldsymbol{\vee}_{o\rightarrow o\rightarrow o}\, \boldsymbol{s}_o)\, \boldsymbol{t}_o)_o\|^{\boldsymbol{M},\boldsymbol{g}} = \boldsymbol{T}$ *iff* $\|\boldsymbol{s}_o\|^{\boldsymbol{M},\boldsymbol{g}} = \boldsymbol{T}$ *or* $\|\boldsymbol{t}_o\|^{\boldsymbol{M},\boldsymbol{g}} = \boldsymbol{T}$
6. $\|(\boldsymbol{\forall}_{(\alpha\rightarrow o)\rightarrow o}(\boldsymbol{\lambda X}_\alpha . \boldsymbol{s}_o))_o\|^{\boldsymbol{M},\boldsymbol{g}} = \boldsymbol{T}$ *iff for all $\boldsymbol{d} \in \boldsymbol{D}_\alpha$ we have* $\|\boldsymbol{s}_o\|^{\boldsymbol{M},\boldsymbol{g}[\boldsymbol{d}/\boldsymbol{X}_\alpha]} = \boldsymbol{T}$

The definition for standard and Henkin models (Def. 9), and for truth in a model, validity, etc. (Def. 10) are adapted in the obvious way, and we use the notation $\boldsymbol{M}, \boldsymbol{g} \models \boldsymbol{s}_o$, $\models \boldsymbol{s}_o$, and $\boldsymbol{\Gamma} \models \boldsymbol{\Delta}$. As for HOML, we assume Henkin semantics in the remainder.

2.3 HOML as a fragment of HOL

The encoding of HOML in HOL is simple: we identify HOML formulas of type o with certain HOL formulas of type $\boldsymbol{\iota} \rightarrow \boldsymbol{o}$. The HOL type $\boldsymbol{\iota} \rightarrow \boldsymbol{o}$ is abbreviated as $\boldsymbol{\sigma}$ in the remainder. More generally, we define for each HOML type $\alpha \in T$ the associated raised HOL type $\lceil \alpha \rceil$ as follows: $\lceil \mu \rceil = \boldsymbol{\mu}$, $\lceil o \rceil = \boldsymbol{\sigma} = \boldsymbol{\iota} \rightarrow \boldsymbol{o}$, and $\lceil \alpha \rightarrow \beta \rceil = \lceil \alpha \rceil \rightarrow \lceil \beta \rceil$. Hence, all HOML terms are rigid, except for those of type o.

Def. 15 *HOML terms s_α are associated with type-raised HOL terms $\lceil s_\alpha \rceil$ in the following way:*

$$\begin{aligned} \lceil p_\alpha \rceil &= \boldsymbol{p}_{\lceil \alpha \rceil} \\ \lceil X_\alpha \rceil &= \boldsymbol{X}_{\lceil \alpha \rceil} \\ \lceil (s_{\alpha\rightarrow\beta}\, t_\alpha) \rceil &= (\lceil s_{\alpha\rightarrow\beta} \rceil \lceil t_\alpha \rceil) \\ \lceil (\lambda X_\alpha . s_\beta) \rceil &= (\boldsymbol{\lambda} \lceil X_\alpha \rceil . \lceil s_\beta \rceil) \\ \lceil (\neg_{o\rightarrow o}\, s_o) \rceil &= (\dot{\boldsymbol{\neg}}_{\sigma\rightarrow\sigma} \lceil s_\alpha \rceil) \\ \lceil ((\vee_{o\rightarrow o\rightarrow o}\, s_o)\, t_o) \rceil &= ((\dot{\boldsymbol{\vee}}_{\sigma\rightarrow\sigma\rightarrow\sigma} \lceil s_\alpha \rceil) \lceil t_\alpha \rceil) \\ \lceil ((\forall_{(\alpha\rightarrow o)\rightarrow o}\, (\lambda X_\alpha . s_\beta) \rceil &= (\dot{\boldsymbol{\forall}}_{(\alpha\rightarrow\sigma)\rightarrow\sigma}\, (\boldsymbol{\lambda} \lceil X_\alpha \rceil . \lceil s_\beta \rceil) \\ \lceil (\Box_{o\rightarrow o}\, s_o) \rceil &= (\dot{\boldsymbol{\Box}}_{\sigma\rightarrow\sigma} \lceil s_o \rceil) \end{aligned}$$

$\dot{\boldsymbol{\neg}}, \dot{\boldsymbol{\vee}}, \dot{\boldsymbol{\forall}}$, *and* $\dot{\boldsymbol{\Box}}$ *are the* type-raised modal HOL connectives *associated with the corresponding modal HOML connectives. They are defined as follows (where $\boldsymbol{r}_{\iota\rightarrow\iota\rightarrow o}$ is a new constant symbol in HOL associated with the accessibility relation R of HOML):*

$$\begin{aligned} \dot{\boldsymbol{\neg}}_{\sigma\rightarrow\sigma} &= \boldsymbol{\lambda s}_\sigma . \boldsymbol{\lambda W}_\iota . \boldsymbol{\neg}\, (\boldsymbol{s}\, \boldsymbol{W}) \\ \dot{\boldsymbol{\vee}}_{\sigma\rightarrow\sigma\rightarrow\sigma} &= \boldsymbol{\lambda s}_\sigma . \boldsymbol{\lambda t}_\sigma . \boldsymbol{\lambda W}_\iota . \boldsymbol{s}\, \boldsymbol{W} \vee \boldsymbol{t}\, \boldsymbol{W} \\ \dot{\boldsymbol{\forall}}_{(\alpha\rightarrow\sigma)\rightarrow\sigma} &= \boldsymbol{\lambda s}_{\alpha\rightarrow\sigma} . \boldsymbol{\lambda W}_\iota . \boldsymbol{\forall X}_\alpha . \boldsymbol{s}\, \boldsymbol{X}\, \boldsymbol{W} \\ \dot{\boldsymbol{\Box}}_{\sigma\rightarrow\sigma} &= \boldsymbol{\lambda s}_\sigma . \boldsymbol{\lambda W}_\iota . \boldsymbol{\forall V}_\iota . \boldsymbol{\neg}(\boldsymbol{r}_{\iota\rightarrow\iota\rightarrow o}\, \boldsymbol{W}\, \boldsymbol{V}) \vee \boldsymbol{s}\, \boldsymbol{V} \end{aligned}$$

As before, we write $\dot{\boldsymbol{\forall}} \boldsymbol{X}_\alpha . \boldsymbol{s}_\sigma$ as shorthand for $\dot{\boldsymbol{\forall}}_{(\alpha\rightarrow\sigma)\rightarrow\sigma}(\boldsymbol{\lambda X}_\alpha . \boldsymbol{s}_\sigma)$. Further operators, such as $\dot{\top}$, $\dot{\bot}$, $\dot{\wedge}$, $\dot{\supset}$, $\dot{\equiv}$, $\dot{\Diamond}$, and $\dot{\exists}$ ($\dot{\exists} \boldsymbol{X}_\alpha . \boldsymbol{s}_\sigma$ is used as shorthand for $\dot{\exists}_{(\alpha\rightarrow\sigma)\rightarrow\sigma}(\boldsymbol{\lambda X}_\alpha . \boldsymbol{s}_\sigma)$) can now be easily defined.[4] The above equations can be treated as abbreviations in HOL theorem provers. Alternatively, they can be stated as axioms where = is either Leibniz equality or primitive equality (if additionally provided in the HOL grammar, as is the case for most modern HOL provers).

[4] We could introduce further modal operators, such as the difference modality $\boldsymbol{D}$, the global modality $\boldsymbol{E}$, nominals with **!**, and the @ operator (cf. [10]).

As a consequence of the above embedding we can express HOML proof problems elegantly in the type-raised syntax of HOL. Using rewriting or definition expanding, we can reduce these representations to corresponding statements containing only the basic HOL connectives $\boldsymbol{\neg}_{o\to o}$, $\boldsymbol{\vee}_{o\to o\to o}$, and $\boldsymbol{\forall}_{(\alpha\to o)\to o}$.

Ex. 1 *The HOML formula* $\Box\,\exists P_{\mu\to o}.\, P\, a_\mu$ *is associated with the type raised HOL formula* $\dot{\Box}\,\dot{\exists} \boldsymbol{P}_{\mu\to\sigma}.\, \boldsymbol{P}\, \boldsymbol{a}_\mu$, *which rewrites into the following* $\beta\eta$*-normal HOL term of type* $\boldsymbol{\sigma}$

$$\lambda \boldsymbol{W}_\iota.\, \forall \boldsymbol{V}_\iota.\, \neg(\boldsymbol{r}\, \boldsymbol{W}\, \boldsymbol{V}) \vee \neg\forall \boldsymbol{P}_{\mu\to\sigma}.\, \neg(\boldsymbol{P}\, \boldsymbol{a}_\mu\, \boldsymbol{V})$$

Next, we define validity of type-raised modal HOL propositions $\boldsymbol{s}_\sigma$ in the obvious way: $\boldsymbol{s}_\sigma$ is valid iff for all possible worlds $\boldsymbol{w}_\iota$ we have $\boldsymbol{w}_\iota \in \boldsymbol{s}_\sigma$, that is, iff $(\boldsymbol{s}_\sigma\, \boldsymbol{w}_\iota)$ holds.

Def. 16 Validity *is modeled as an abbreviation for the following* λ*-term:* ***valid*** $= \lambda \boldsymbol{s}_{\iota\to o}.\, \forall \boldsymbol{W}_\iota.\, \boldsymbol{s}\, \boldsymbol{W}$ *(alternatively, we could define validity simply as* $\boldsymbol{\forall}_{(\iota\to o)\to o}$*). Instead of* **valid** $\boldsymbol{s}_\sigma$ *we also use the notation* $[\boldsymbol{s}_\sigma]$.

Ex. 2 *We analyze whether the type-raised modal HOL formula* $\dot{\Box}\,\dot{\exists} \boldsymbol{P}_{\mu\to\sigma}.\, (\boldsymbol{P}\, \boldsymbol{a}_\mu)$ *is valid or not. For this, we formalize the HOL proof problem* $[\dot{\Box}\,\dot{\exists} \boldsymbol{P}_{\mu--\to\sigma}.\, (\boldsymbol{P}\, \boldsymbol{a}_\mu)]$, *which expands into* $\forall \boldsymbol{W}_\iota.\, \forall \boldsymbol{V}_\iota.\, \neg(\boldsymbol{r}\, \boldsymbol{W}\, \boldsymbol{V}) \vee \neg\forall \boldsymbol{P}_{\mu\to\sigma}.\, \neg(\boldsymbol{P}\, \boldsymbol{a}_\mu\, \boldsymbol{V})$*. It is easy to check that this term is valid in Henkin semantics: put* $\boldsymbol{P} = \lambda \boldsymbol{X}_\mu.\, \lambda \boldsymbol{Y}_\iota.\, \top$.

Theorem 1 (Soundness and Completeness) *For all HOML formulas* s_o *we have:*

$$\models s_o \quad \textit{iff} \quad \boldsymbol{\models} [\lceil s_o \rceil]$$

Proof sketch: *The proof adapts the ideas presented in [10]. By contraposition it is sufficient to show* $\not\models\ s_o$ *iff* $\boldsymbol{\not\models}\ [\lceil s_o \rceil]$*, that is,* $\|s_o\|^{M,g,w}$ *(for some HOML model M, assignment g, and w) iff* $\|\forall \boldsymbol{W}_\iota.\, \lceil \boldsymbol{s_o} \rceil\, \boldsymbol{W}\|^{\boldsymbol{M},\boldsymbol{g}}$ *(for some HOL model* $\boldsymbol{M}$ *and assignment* $\boldsymbol{g}$*) iff* $\|\lceil \boldsymbol{s_o} \rceil\, \boldsymbol{W}\|^{\boldsymbol{M},\boldsymbol{g}[w/W]}$ *(for some* $\boldsymbol{M}$*,* $\boldsymbol{g}$*, and w). We easily get the proof by choosing the obvious correspondences between D and* **D***, W and* $\boldsymbol{D}_\iota$*, I and* **I***, g and* **g***, R and* $r_{\iota\to\iota\to o}$*, and w and* **w***.* □

From Theorem 1 we get the following corollaries:

$$\models^{KB} s_o \quad \text{iff} \quad (\textbf{symmetric}\ \boldsymbol{r}_{\iota\to\iota\to o}) \boldsymbol{\models} [\lceil s_o \rceil]$$

$$\models^{S5} s_o \quad \text{iff} \quad (\textbf{equiv-rel}\ \boldsymbol{r}_{\iota\to\iota\to o}) \boldsymbol{\models} [\lceil s_o \rceil]$$

where **symmetric** and **equiv-rel** are defined in an obvious way.

Constant domain quantification is addressed above. Techniques for handling varying domain and cumulative domain quantification in the embedding of first-order modal logics in HOL have been outlined in [8]. These techniques, which have also been adapted for the theory above, cannot be presented here for space limitations.

Note that also non-rigid terms can easily be modeled by type-raising. For example, a non-rigid HOML constant symbol kingOfFrance$_\mu$ would be mapped to a type-raised (and thus world-depended) HOL constant symbol **kingOfFrance**$_{\iota\to\mu}$.

3 EXPERIMENTS

The above embedding has been encoded in the concrete THF0 syntax [33] for HOL; cf. the files `Quantified_K/_KB/_S5.ax`[5] available at `https://github.com/FormalTheology/GoedelGod/tree/master/Formalizations/THF` (all files mentioned below are provided under this URL). The definition for quantifier $\dot{\boldsymbol{\forall}}_{((\mu\to\sigma)\to\sigma)\to\sigma}$, for example, is given as[6]

```
thf(mforall_indset,definition,
    ( mforall_indset
    = ( ^ [S: ( mu > $i > $o ) > $i > $o,W: $i] :
          ! [X: mu > $i > $o] :
            ( S @ X @ W ) ) )).
```

Subsequently the axioms, definitions, and theorems from Fig. 1 and some further, related problems have been encoded in THF0. Then the THF0 compliant HOL-ATPs LEO-II [11], Satallax [17], and Nitpick [16] have been employed to automate the proof problems. LEO-II, which internally cooperates with the first-order prover E [29], was used exclusively in the initial round of experiments, that is, it was the first prover to automate Gödel's ontological argument.

Theorem T1 from Fig. 1, for example, is formalized as

```
thf(thmT1,conjecture,
   ( v
    @ ( mforall_indset
      @ ^ [Phi: mu > $i > $o] :
          ( mimplies @ ( p @ Phi )
          @ ( mdia
            @ ( mexists_ind
              @ ^ [X: mu] :
                  ( Phi @ X ) ) ) ) ) )).
```

This encodes the HOL formula

$$[\dot{\forall} \boldsymbol{\phi}_{\mu\to\sigma}.\, \boldsymbol{p}_{(\mu\to\sigma)\to\sigma} \boldsymbol{\phi}\ \dot{\supset}\ \dot{\Diamond}\dot{\exists} \boldsymbol{X}_\mu.\, \boldsymbol{\phi} \boldsymbol{X}]$$

`v` in the THF0 encoding stands for **valid** and p corresponds to the uppercase P, for 'positive', from Fig. 1. The respective encodings and the results of a series of recent experiments with LEO-II (version 1.6.2), Satallax (version 2.7), and Nitpick (version 2013) are provided in Fig. 2. The first row marked with T1, for example, shows that theorem T1 follows from axioms A2 and A1 (where only the ⊃-direction is needed); LEO-II and Satallax confirm this in 0.1 second. The experiments have been carried out w.r.t. the logics K and/or KB, and w.r.t. constant (const) and varying (vary) domain semantics for the domains of individuals. The exact dependencies (available axioms and definitions) are displayed for each single problem. The results of the prover calls are given in seconds. '—' means timeout. 'THM', 'CSA', 'SAT', and 'UNS' are the reported result statuses; they stand for 'Theorem', 'CounterSatisfiable', 'Satisfiable', and 'Unsatisfiable', respectively. The experiments can be easily reproduced: all relevant files have been provided at the above URL. For example, the two THF0 problem files associated with the first table row for T1 are `T1_K_const_min.p` and `T1_K_vary_min.p`, and those associated with the second row for T1 are `T1_K_const_max.p` and `T1_K_vary_max.p`, respectively. Moreover, a simple shell script `call_tptp.sh` is provided, which can be used to make remote calls to LEO-II, Satallax, and Nitpick installed at Sutcliffe's SystemOnTPTP infrastructure [32] at the University of Miami. The experiments used standard 2.80GHz computers with 1GB memory remotely located in Miami.

[5] The formalization in these files slightly varies from the above theory w.r.t. technical details. For example, a generic □-operator is introduced that can be instantiated for different accessibility relations as e.g. required for multi-modal logic applications (cf. [10]). Moreover, since THF0 does not support polymorphism, copies of the $\dot{\boldsymbol{\forall}}_{(\alpha\to\sigma)\to\sigma}$ and $\dot{\boldsymbol{\exists}}_{(\alpha\to\sigma)\to\sigma}$ connectives are provided only for the selected types $(\mu \to \sigma) \to \sigma$ and $((\mu \to \sigma) \to \sigma) \to \sigma$ as precisely required in Gödels's proof. The Isabelle version [13] and the Coq version of the encoding instead provide respective polymorphic definitions.

[6] `$i`, `$o`, and `mu` represent the HOL base types $\boldsymbol{i}$, $\boldsymbol{o}$, and $\boldsymbol{\mu}$. `$i>$o` encodes a function (predicate) type. Function application is represented by `@`, and for universal quantification, existential quantification and λ-abstraction the symbols `!`, `?` and `^` are employed. ¬, ∨, ∧, and ⊃ are written as `~`, `|`, `&`, and `=>`, respectively. The type-raised modal connectives are called `mforall_*`, `mexists_*`, `mnot`, `mor`, `mand`, `mimplies`, etc.

	HOL encoding	dependencies	logic	status	LEO-II	Satallax	Nitpick
					const/vary	const/vary	const/vary
A1	$[\dot\forall\phi_{\mu\to\sigma}\centerdot p_{(\mu\to\sigma)\to\sigma}(\lambda X_\mu\centerdot\dot\neg(\phi X))\dot\equiv\dot\neg(p\phi)]$						
A2	$[\dot\forall\phi_{\mu\to\sigma}\centerdot\dot\forall\psi_{\mu\to\sigma}\centerdot(p_{(\mu\to\sigma)\to\sigma}\phi\dot\wedge\dot\Box\dot\forall X_\mu\centerdot(\phi X\dot\supset\psi X))\dot\supset p\psi]$						
T1	$[\dot\forall\phi_{\mu\to\sigma}\centerdot p_{(\mu\to\sigma)\to\sigma}\phi\dot\supset\dot\Diamond\dot\exists X_\mu\centerdot\phi X]$	A1(⊃), A2	K	THM	0.1/0.1	0.0/0.0	—/—
		A1, A2	K	THM	0.1/0.1	0.0/5.2	—/—
D1	$g_{\mu\to\sigma} = \lambda X_\mu\centerdot\dot\forall\phi_{\mu\to\sigma}\centerdot p_{(\mu\to\sigma)\to\sigma}\phi\dot\supset\phi X$						
A3	$[p_{(\mu\to\sigma)\to\sigma}g_{\mu\to\sigma}]$						
C	$[\dot\Diamond\dot\exists X_\mu\centerdot g_{\mu\to\sigma}X]$	T1, D1, A3	K	THM	0.0/0.0	0.0/0.0	—/—
		A1, A2, D1, A3	K	THM	0.0/0.0	5.2/31.3	—/—
A4	$[\dot\forall\phi_{\mu\to\sigma}\centerdot p_{(\mu\to\sigma)\to\sigma}\phi\dot\supset\dot\Box p\phi]$						
D2	$\mathbf{ess}_{(\mu\to\sigma)\to\mu\to\sigma} = \lambda\phi_{\mu\to\sigma}\centerdot\lambda X_\mu\centerdot\phi X\dot\wedge\dot\forall\psi_{\mu\to\sigma}\centerdot(\psi X\dot\supset\dot\Box\dot\forall Y_\mu\centerdot(\phi Y\dot\supset\psi Y))$						
T2	$[\dot\forall X_\mu\centerdot g_{\mu\to\sigma}X\dot\supset(\mathbf{ess}_{(\mu\to\sigma)\to\mu\to\sigma}gX)]$	A1, D1, A4, D2	K	THM	19.1/18.3	0.0/0.0	—/—
		A1, A2, D1, A3, A4, D2	K	THM	12.9/14.0	0.0/0.0	—/—
D3	$\mathbf{NE}_{\mu\to\sigma} = \lambda X_\mu\centerdot\dot\forall\phi_{\mu\to\sigma}\centerdot(\mathbf{ess}\,\phi X\dot\supset\dot\Box\dot\exists Y_\mu\centerdot\phi Y)$						
A5	$[p_{(\mu\to\sigma)\to\sigma}\mathbf{NE}_{\mu\to\sigma}]$						
T3	$[\dot\Box\dot\exists X_\mu\centerdot g_{\mu\to\sigma}X]$	D1, C, T2, D3, A5	K	CSA	—/—	—/—	3.8/6.2
		A1, A2, D1, A3, A4, D2, D3, A5	K	CSA	—/—	—/—	8.2/7.5
		D1, C, T2, D3, A5	KB	THM	0.0/0.1	0.1/5.3	—/—
		A1, A2, D1, A3, A4, D2, D3, A5	KB	THM	—/—	—/—	—/—
MC	$[s_\sigma\dot\supset\dot\Box s_\sigma]$	D2, T2, T3	KB	THM	17.9/—	3.3/3.2	—/—
		A1, A2, D1, A3, A4, D2, D3, A5	KB	THM	—/—	—/—	—/—
FG	$[\dot\forall\phi_{\mu\to\sigma}\centerdot\dot\forall X_\mu\centerdot(g_{\mu\to\sigma}X\dot\supset(\dot\neg(p_{(\mu\to\sigma)\to\sigma}\phi)\dot\supset\dot\neg(\phi X)))]$	A1, D1	KB	THM	16.5/—	0.0/0.0	—/—
		A1, A2, D1, A3, A4, D2, D3, A5	KB	THM	12.8/15.1	0.0/5.4	—/—
MT	$[\dot\forall X_\mu\centerdot\dot\forall Y_\mu\centerdot(g_{\mu\to\sigma}X\dot\supset(g_{\mu\to\sigma}Y\dot\supset X\doteq Y))]$	D1, FG	KB	THM	—/—	0.0/3.3	—/—
		A1, A2, D1, A3, A4, D2, D3, A5	KB	THM	—/—	—/—	—/—
CO	$\emptyset$ (no goal, check for consistency)	A1, A2, D1, A3, A4, D2, D3, A5	KB	SAT	—/—	—/—	7.3/7.4
D2'	$\mathbf{ess}_{(\mu\to\sigma)\to\mu\to\sigma} = \lambda\phi_{\mu\to\sigma}\centerdot\lambda X_\mu\centerdot\dot\forall\psi_{\mu\to\sigma}\centerdot(\psi X\dot\supset\dot\Box\dot\forall Y_\mu\centerdot(\phi Y\dot\supset\psi Y))$						
CO'	$\emptyset$ (no goal, check for consistency)	A1(⊃), A2, D2', D3, A5	KB	UNS	7.5/7.8	—/—	—/—
		A1, A2, D1, A3, A4, D2', D3, A5	KB	UNS	—/—	—/—	—/—

Figure 2. HOL encodings and experiment results for Scott's version of Gödel's ontological argument from Fig. 1.

4 MAIN FINDINGS

Several interesting and partly novel findings have been contributed by the HOL-ATPs, including:

1. The axioms and definitions from Fig. 1 are consistent (cf. CO in Fig. 2).
2. Logic K is sufficient for proving T1, C and T2.
3. For proving the final theorem T3, logic KB is sufficient (and for K a countermodel is reported). This is highly relevant since several philosophers have criticized Gödel's argument for the use of logic S5. This criticism is thus provably pointless.
4. Only for T3 the HOL-ATPs still fail to produce a proof directly from the axioms; thus, T3 remains an interesting benchmark problem; T1, C, and T2 are rather trivial for HOL-ATPs.
5. Gödel's original version of the proof [23], which omits conjunct $\phi(x)$ in the definition of *essence* (cf. D2'), seems inconsistent (cf. the failed consistency check for CO' in Fig. 2). As far as we are aware of, this is a new result.
6. Gödel's axioms imply what is called the modal collapse (cf. MC in Fig. 2) $\phi \supset \Box\phi$, that is, contingent truth implies necessary truth (which can even be interpreted as an argument against free will; cf. [31]). MC is probably the most fundamental criticism put forward against Gödel's argument.
7. For proving T1, only the ⊃-direction of A1 is needed. However, the ⊂-direction of A1 is required for proving T2. Some philosophers (e.g. [3]) try to avoid MC by eluding/replacing the ⊃-direction of A1.
8. Gödel's axioms imply a 'flawless God', that is, an entity that can only have 'positive' properties (cf. FG in Fig. 2). However, a comment by Gödel in [23] explains that 'positive' is to be interpreted in a moral aesthetic sense only.
9. Another implication of Gödel's axioms is monotheism (see MT in Fig. 2). MT can easily be proved by Satallax from FG and D1. It remains non-trivial to prove it directly from Gödel's axioms.
10. All of the above findings hold for both constant domain semantics and varying domain semantics (for the domain of individuals).

The above findings, in particular (10), well illustrate that the theory framework from Sec. 2 has a great potential towards a flexible support system for *computational theoretical philosophy*. In fact, Gödel's ontological argument has been verified and even automated not only for one particular setting of logic parameters, but these logic parameters have been varied and the validity of the argument has been reconfirmed (or falsified, cf. D2' and CO') for the modified setting. Moreover, our framework is not restricted to a particular theorem proving system, but has been fruitfully employed with some of the most prominent automated and interactive theorem provers available to date.

5 RELATED AND FUTURE WORK

We are pioneering the computer-supported automation of modern versions of the ontological argument. There are two related papers [27, 28]. Both focus on the comparably simpler argument by Anselm. [27] encodes (a variant) of Anselm's argument in first-order logic and employs the theorem prover PROVER9 in experiments; this work has been criticized in [22]. The work in [28], which has evolved in parallel to ours, interactively verifies Anselm's argument in the higher-order proof assistant PVS. Note in particular, that both formalizations do not achieve the close correspondence between the original formulations and the formal encodings that can be found in our approach.

A particular strength of our universal logic framework is that it can be easily adapted for logic variations and even supports flexible combinations of logics (cf. [6]). In ongoing and future work we will therefore investigate further logic parameters for Gödel's argument, including varying domains at higher types and non-rigid terms. We plan to make the entire landscape of results available to the interested communities. This is relevant, since philosophers are sometimes imprecise about the very details of the logics they employ.

6 CONCLUSION

While computers can now calculate, play games, translate, plan, learn and classify data much better than we humans do, tasks involving philosophical and theological inquiries have remained mostly untouched by our technological progress up to now. Due to the abstract and sophisticated types of reasoning they require, they can be considered a challenging frontier for automated reasoning.

We accepted this challenge and decided to tackle, with automated reasoning techniques, a philosophical problem that is almost 1000 years old: the ontological argument for God's existence, firstly proposed by St. Anselm of Canterbury and greatly improved by Descartes, Leibniz, Gödel and many others throughout the centuries. So far, there was no AI system capable of dealing with such complex problems. We created a prototypical infrastructure extending widely used systems such as LEO-II, Satallax, and Nitpick (and Isabelle and Coq) to allow them to cope with modalities; and using the extended systems we were able to automatically reconstruct and verify Gödel's argument, as well as discover new facts and confirm controversial claims about it. This is a landmark result, with media repercussion in a global scale, and yet it is only a glimpse of what can be achieved by combining computer science, philosophy and theology.

Our work, in this sense, offers new perspectives for a computational 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!

ACKNOWLEDGEMENTS

We thank Alexander Steen, Max Wisniewski, and the anonymous reviewers for their comments and suggestions.

REFERENCES

[1] R.M. Adams, 'Introductory note to *1970', in *Kurt Gödel: Collected Works Vol. 3: Unpubl. Essays and Letters*, Oxford Univ. Press, (1995).

[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*, 167–172, Springer, (1996).

[3] C.A. Anderson, 'Some emendations of Gödel's ontological proof', *Faith and Philosophy*, **7**(3), (1990).

[4] P.B. Andrews, 'General models and extensionality', *Journal of Symbolic Logic*, **37**(2), 395–397, (1972).

[5] P.B. Andrews, 'Church's type theory', in *The Stanford Encyclopedia of Philosophy*, ed., E.N. Zalta, spring 2014 edn., (2014).

[6] C. Benzmüller, 'HOL based universal reasoning', in *Handbook of the 4th World Congress and School on Universal Logic*, ed., J.Y. Beziau et al., pp. 232–233, Rio de Janeiro, Brazil, (2013).

[7] C. Benzmüller and D. Miller, 'Automation of higher-order logic', in *Handbook of the History of Logic, Volume 9 — Logic and Computation*, Elsevier, (2014). Forthcoming; preliminary version available at http://christoph-benzmueller.de/papers/B5.pdf.

[8] C. Benzmüller, J. Otten, and Th. Raths, 'Implementing and evaluating provers for first-order modal logics', in *Proc. of the 20th European Conference on Artificial Intelligence (ECAI)*, pp. 163–168, (2012).

[9] 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*, ed., C. Benzmüller et al., 386–406, College Publications, (2008).

[10] C. Benzmüller and L.C. Paulson, 'Quantified multimodal logics in simple type theory', *Logica Universalis*, **7**(1), 7–20, (2013).

[11] 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*, number 5195 in LNAI, pp. 162–170. Springer, (2008).

[12] C. Benzmüller and B. Woltzenlogel-Paleo, 'Formalization, mechanization and automation of Gödel's proof of God's existence', *arXiv:1308.4526*, (2013).

[13] C. Benzmüller and B. Woltzenlogel-Paleo, 'Gödel's God in Isabelle/HOL', *Archive of Formal Proofs*, (2013).

[14] C. Benzmüller and B. Woltzenlogel-Paleo, 'Gödel's God on the computer', in *Proceedings of the 10th International Workshop on the Implementation of Logics*, EPiC Series. EasyChair, (2013). Invited abstract.

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

[16] 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, pp. 131–146. Springer, (2010).

[17] C.E. Brown, 'Satallax: An automated higher-order prover', in *Proc. of IJCAR 2012*, number 7364 in LNAI, pp. 111 – 117. Springer, (2012).

[18] R. Corazzon. Contemporary bibliography on ontological arguments: `http://www.ontology.co/biblio/ontological-proof-contemporary-biblio.htm`.

[19] M. Fitting, *Types, Tableaux and Gödel's God*, Kluwer, 2002.

[20] M. Fitting and R.L. Mendelsohn, *First-Order Modal Logic*, volume 277 of *Synthese Library*, Kluwer, 1998.

[21] D. Gallin, *Intensional and Higher-Order Modal Logic*, North-Holland, 1975.

[22] P. Garbacz, 'PROVER9's simplifications explained away', *Australasian Journal of Philosophy*, **90**(3), 585–592, (2012).

[23] K. Gödel, *Appx.A: Notes in Kurt Gödel's Hand*, 144–145. In [31], 2004.

[24] L. Henkin, 'Completeness in the theory of types', *Journal of Symbolic Logic*, **15**(2), 81–91, (1950).

[25] R. Muskens, 'Higher Order Modal Logic', in *Handbook of Modal Logic*, ed., P Blackburn et al., 621–653, Elsevier, Dordrecht, (2006).

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

[27] P.E. Oppenheimera and E.N. Zalta, 'A computationally-discovered simplification of the ontological argument', *Australasian Journal of Philosophy*, **89**(2), 333–349, (2011).

[28] J. Rushby, 'The ontological argument in PVS', in *Proc. of CAV Workshop "Fun With Formal Methods"*, St. Petersburg, Russia,, (2013).

[29] S. Schulz, 'E – a brainiac theorem prover', *AI Communications*, **15**(2), 111–126, (2002).

[30] D. Scott, *Appx.B: Notes in Dana Scott's Hand*, 145–146. In [31], 2004.

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

[32] G. Sutcliffe, 'The TPTP problem library and associated infrastructure', *Journal of Automated Reasoning*, **43**(4), 337–362, (2009).

[33] 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).

[34] B. Woltzenlogel-Paleo and C. Benzmüller, 'Automated verification and reconstruction of Gödel's proof of God's existence', *OCG J.*, (2013).