Benzmüller, Christoph; Scott, Dana

Notes on Gödel’s and Scott’s variants of the ontological argument

Date of secondary publication: 09.01.2026

Version of Record (Published Version), Article

Persistent identifier: urn:nbn:de:bvb:473-irb-112503x

Primary publication

Benzmüller, Christoph; Scott, Dana (2025): Notes on Gödel’s and Scott’s variants of the ontological argument, in: Monatshefte für Mathematik, Wien [u.a.]: Springer, Vol. 208, Nr. 4, pp. 569–611, doi: 10.1007/s00605-025-02078-x.

Monatshefte für Mathematik (2025) 208:569–611
https://doi.org/10.1007/s00605-025-02078-x

Notes on Gödel's and Scott's variants of the ontological argument

Christoph Benzmüller[1,2] · **Dana Scott**[3]

Received: 11 February 2025 / Accepted: 25 March 2025 / Published online: 21 April 2025

Abstract
Notes on Kurt Gödel's modal ontological argument and Dana Scott's variant of it are presented. These remarks, supported by experimental studies with a proof assistant system for classical higher-order logic, implicitly answer some questions the authors have received over the last decade(s). In addition, some new insights resulting from the conducted experiments are reported.

Keywords Gödel's ontological argument · Interactive and automated theorem proving · Higher-order modal logic

Mathematics Subject Classification 68V15 · 68V20 · 03B16 · 03B38 · 03B45 · 03B80 · 03A05 · 03-03 · 97E20 · 97E30 · 97E50

1 Introduction

In spring 1970, Kurt Gödel met with the senior author (Scott) at the Princeton Philosophy Department to ask him confidentially to preserve some papers and notes which he did not want to be overlooked in case of his illness and death. One of the notes was a very brief sketch of an ontological argument for the existence of God, drawing upon

---

Communicated by Adrian Constantin.

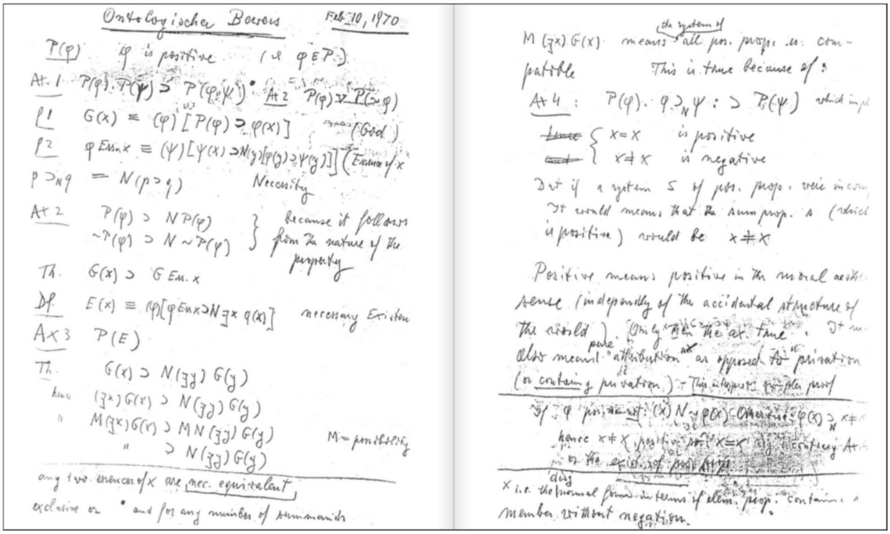
✉ Christoph Benzmüller
christoph.benzmueller@uni-bamberg.de

Dana Scott
scott@andrew.cmu.edu

[1] AI Systems Engineering, Otto-Friedrich Universität Bamberg, An der Weberei 5, 96047 Bamberg, Bavaria, Germany

[2] Faculty of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 7, 14195 Berlin, Germany

[3] Topos Institute, 2140 Shattuck Ave, Berkeley, CA 94704, USA



Ontologischer Beweis   Feb 10, 1970

$P(\varphi)$ φ is positive (or $\varphi \in P$)

At. 1 $P(\varphi).P(\psi) \supset P(\varphi.\psi)$ • At 2 $P(\varphi) \vee P(\sim\varphi)$

Df 1 $G(x) \equiv (\varphi)[P(\varphi) \supset \varphi(x)]$ (God)

Df 2 $\varphi \text{ Ess. } x \equiv (\psi)[\psi(x) \supset N(y)[\varphi(y) \supset \psi(y)]]$ (Essence of x

$p \supset_N q = N(p \supset q)$ Necessity

At 2 $P(\varphi) \supset NP(\varphi)$, $\sim P(\varphi) \supset N\sim P(\varphi)$ } because it follows from the nature of the property

Th. $G(x) \supset G \text{ Ess. } x$

Df. $E(x) \equiv (\varphi)[\varphi \text{ Ess. } x \supset N\exists x\, \varphi(x)]$ necessary Existen

Ax 3 $P(E)$

Th. $G(x) \supset N(\exists y)G(y)$

hence $(\exists x)G(x) \supset N(\exists y)G(y)$

" $M(\exists x)G(x) \supset MN(\exists y)G(y)$ M = possibility

" $\supset N(\exists y)G(y)$

any two essences of x are nec. equivalent

exclusive or • and for any number of summands

$M(\exists x)G(x)$ means the system of all pos. props. is compatible   This is true because of:

At 4: $P(\varphi).\varphi \supset_N \psi : \supset P(\psi)$ which impl

~~[illegible]~~ { $x=x$ is positive

~~[illegible]~~ { $x \neq x$ is negative

But if a system S of pos. props. were incom It would mean that the sum prop. s (which is positive) would be $x \neq x$

Positive means positive in the moral aesth. sense (independently of the accidental structure of the world). Only then the ax. true. It m also meant pure "attribution"* as opposed to "privation" (or containing privation). This interpret. simpler proof

If φ pos. not: $(x)N\sim\varphi(x)$. Otherwise: $\varphi(x) \supset_N x \neq x$

hence $x \neq x$ positive not $x=x$ [illegible] contrary At or the exist. of pos. prop.

* i.e. the disj. normal form in terms of elem. prop. contains a member without negation.

Fig. 1 Manuscript of Gödel's ontological argument, dated 1970, as discovered in his Nachlass. (Disclaimer: Unpublished works of Kurt Gödel are Copyright Institute for Advanced Study and are used with permission. All rights reserved by Institute for Advanced Study.)

the works of Anselm, Leibniz and others, which Gödel had studied for many years.[1] His version introduced some relevant novel elements, in particular, it was developed with a high degree of mathematical formality in a modal higher-order logic significantly advancing, he felt, on earlier more informal, natural explorations in the works of the philosophers he had consulted.

Soon after, Scott - without permission - mentioned the proof in a modal logic seminar as an amusing application of logic. His action was very unprofessional as Gödel had asked for confidentially! Unfortunately word spread very quickly indeed about the proof (which actually Scott had slightly modified), and over the years many, many commentaries have been published. Scott still feels embarrassed in remembering what he had done. He never subsequently discussed the matter with Gödel himself as he left Princeton soon after to take up the new chair in Mathematical Logic at Oxford.

Later, in Gödel's Nachlass, a manuscript[2] dated 1970 was found [5], containing Gödel's (supposedly) final version of the argument; see also Fig. 1. The notes that Gödel showed Scott in their meeting were very brief compared to that version. The manuscript is also different from the slightly modified version [6] that Scott discussed in the Princeton seminar. His Nachlass contains other earlier variants and notes which show that Gödel had been working on the ontological argument for many years, apparently as early as the 1940s or possibly even the 1930s; we refer to Kanckos and Lethen [4] for further insights into these historical aspects.

---

[1] For relevant background information we refer e.g. to Adams [1], Wang [2], Sobel [3] and Kanckos and Lethen [4].

[2] The exact location of this manuscript in Gödel's Nachlass is: Kurt Gödel Papers, Box 12, Folder 41, item accession 060565.

In 2014, Benzmüller and Bruno Woltzenlogel-Paleo formalized and verified [7–9] Scott's variant of Gödel's 1970 argument using modern automated theorem provers and proof assistants for higher-order logic [10, 11], demonstrating as a side effect the maturity that these formal tools have reached today also for experiments in philosophical logic and metaphysics. Subsequent work has contributed a number of similar studies of a number of variants of Gödel's ontological argument proposed by other authors, including Anderson, Hájek and Fitting [12–18].

The application of these systems was made possible by a novel bridge [19] between higher-order modal logic (HOML, the logic assumed in the arguments, see e.g. [17, 20–22]) and classical higher-order logic (HOL, the logic mechanized in the theorem proving systems, see e.g. [23–28]), using shallow embeddings of the former logic in the latter, which serves as the metalogic in which the tool-assisted experiments were performed. This bridge was previously explored by Benzmüller and Paulson [19], the core ideas of which later became the foundation of a logico-pluralistic knowledge representation and reasoning methodology called LogiKEy [29–31].

Although the experiments carried out by Benzmüller with colleagues, which revealed and computationally confirmed various findings, have already been extensively documented in the literature (see e.g. [7–9, 32–36] and further references in [30]), numerous requests for further clarification continue to be received via e-mail on a regular basis. The purpose of this paper is therefore twofold. On the one hand, we will address some questions that have been asked over the years, and on the other hand, we will add some new results, insights, and comments that result from our experiments and that we believe are of some value.

The formalization and experiments presented in this paper were carried out using the Isabelle/HOL proof assistant [37], and as a side contribution, this paper further demonstrates the use of modern proof assistants for formalizing and exploring foundations in logic and metaphysics [29, 30]. Other proof assistants could have been chosen; see e.g. [10, 38] for a discussion of the state of the art in this area.

We think that Gödel's ontological proof should one day be included in a future edition of Aigner and Ziegler's compendium "Proofs from THE BOOK" [39]. This is not only because of its elegance, richness, and conciseness, but also because it provides an interesting bridge linking theistic notions to prominent mathematical structures such as filters and ultrafilters [35] within a higher-order modal logic context. There is also the insight it highlights about proofs in general, namely their dependence on (possibly debatable or controversial) postulates and definitions, which may even include assumptions about the particular logic used. And there is Gödel's nuanced realist stance [40] and his general metaphysical-theological-mathematical views [41–43], which make this piece of Gödel's work, which apparently fascinated him throughout his research life, an interesting and thought-provoking addition to this compendium.

The structure of the article is as follows: Sect. 2 presents a short overview of interactive and automated higher-order reasoning tools that can be used in mathematics and other fields, including metaphysics. In addition, this section provides a brief introduction to some aspects of Isabelle/HOL that are relevant to this article. Section 3 clarifies questions about the detailed logical assumptions for HOML that we consider in the rest of the article; our interest is in providing as much detail as necessary to keep this article self-contained.[3] Readers who are less interested in proof assistant systems such as Isabelle/HOL or less interested in logical details and/or logic embeddings following the LogiKEy methodology, may simply skip the Sects. 2 and/or 3 at first reading and come back later. Section 4 formally studies Gödel's modal ontological argument as presented in his 1970 manuscript, using Isabelle/HOL as a tool. It stays as close as possible to Gödel's 1970 manuscript and examines several variants, one of which is novel. This section also briefly discusses the notion of evil in the given context, and shows that the Gödel's notion of positive properties constitutes a (modal) ultrafilter. Section 5 formally examines Scott's variant, using the same Isabelle/HOL encoding of HOML as before, and shows the exact relation to the variants of Gödel's 1970 argument from Sect. 4. Section 6 offers some final remarks.

2 Interactive and automated proof assistant systems

While first-order logic and axiomatizations of set theory have played and continue to play a prominent role in mathematical logic and its foundations, it is classical or constructive/intuitionistic variants of higher-order logic (based on typed λ-calculi) that have been chosen as foundation in modern proof assistance systems. These systems are being developed for practical use not only in mathematics, but also in a number of other fields, including, for example, computer science, symbolic AI and, as we demonstrate in this article, metaphysics. The development of formal libraries of verified theories in these areas has been underway for some time in higher-order proof assistants such as Isabelle/HOL [37, 45], Coq [46], Lean [47, 48], HOL4 [49], HOL light [50], PVS [51], NuPrL [52], Agda [53], and others. Some of these systems have even been used to resolve challenge questions in maths, such as the verification of Hales' proof of the Keppler conjecture [54]. Lean in particular has recently received some public attention as one option from the above system family that could play an important role also in the development of future generations of hybrid AI technology to be applied in mathematics and much beyond.[4]

The logico-pluralistic experiments in metaphysics presented in this article can in principle be carried out, modulo appropriate adaptations, in several of these proof

---

[3] The article shows the formalization sources and they are attached to this article as a supplement. In addition they are available for download in the Archive of Formal Proofs (AFP) at https://www.isa-afp.org/entries/Notes_On_Goedels_Ontological_Argument.html [44] (the AFP version of the sources is slightly modified to comply with AFP requirements).

[4] This identified increased importance, e.g. in the context of hybrid AI technology, applies analogously to all systems mentioned, and is relevant towards both the feasibility of strong AI, resp. Artificial General Intelligence, and, most importantly, on the way to greater (verifiable) reliability, safety, and trust in a next generation of AI systems (see e.g. the discussions in [55–58]). Proposals to e.g. use symbolic AI methods to control subsymbolic AI models are indeed not new (see e.g. [31, 59–63]), but they are now receiving more attention, and this also includes the direction of autoformalizations of natural language proofs and arguments [64]. Leibniz, and with him Gödel, would presumably be delighted to be part of the developments ahead in terms of an interdisciplinary use of expressive logics and associated tools in AI, mathematics and philosophy, for remember: *"If we had it [a characteristica universalis], we should be able to reason in metaphysics and morals in much the same way as in geometry and analysis."* Leibniz ([65, p.4] with reference to [66]).

assistants.[5] Our arguments for choosing Isabelle/HOL include: (i) in addition to its powerful internal proof tactics, it provides excellent proof automation tools such as Sledgehammer [67] (which interfaces with the world's leading automated theorem provers, see below), (ii) it provides the higher-order model finder Nitpick [68], which is capable of computing (finite) examples or counterexamples to conjectures, (iii) it supports vernacular, island-style high-level proofs using the Isar proof language [69], and (iv) it comes with a very sophisticated, configurable user interface supporting e.g. LaTeX-quality formula representations.[6]

Over the past two decades, there has also been a notable shift of interest within the automated theorem proving community from first-order logic to higher-order logic.[7] This has led to the development of increasingly sophisticated automated theorem provers for higher-order logic, such as Zipperposition [76], Vampire [77], E [78], Leo-II [79], Leo-III [80], CVC5 [81], Satallax [82] to name but a few. Some of these systems are designed as extensions of SMT solvers or superposition-based first-order provers for reasoning in higher-order logic. Moreover, almost all of them integrate or interface in some form with SAT solvers, and SAT solvers on their own have been used in recent years to tackle open mathematical problems [83, 84]. It is such automated theorem provers that can be invoked within the Isabelle/HOL proof assistant using the Sledgehammer tool. Prominent early higher-order theorem proving systems that have pioneered interactive and automated proof already in the 1980s are TPS [85] and Automath [86]. A well known and easy theorem that has already been studied in such early systems is the surjective Cantor theorem [87, 88], see Fig. 2, which we reuse here despite its folklore status to briefly introduce and discuss some features of Isabelle/HOL relevant to the remainder of the article.

Isabelle/HOL comes with a notion of simple types extended by some restricted notion of polymorphism and axiomatic type classes (type classes are not used in this article). The bivalent type of Booleans (termed *bool*) is predefined and further base types can be declared (examples can e.g. be found in lines 7–8 of Fig. 3 where two base types i and e are introduced). For two given types α and β a function type $\alpha{\Rightarrow}\beta$ is induced, whose associated domain $D_{\alpha\Rightarrow\beta}$ contains total functions from the underlying domains D_α to D_β. To support polymorphism, type variables are allowed which represent arbitrary types from the entire type hierarchy (see e.g. the quoted symbols $'a$ in line 4 of Fig. 2). Function types mapping to type *bool*, such as the type $'a{\Rightarrow}bool$ (in line 4), can also be seen as set/predicate types, since terms of that type (such as $\lambda X{::}'a.X{=}X$, where :: is the typing symbol of Isabelle/HOL) denote characteristic functions for associated sets (such as the set $\{X{::}'a|X{=}X\}$, resp. $\{X{::}'a|True\}$, of

---

[5] Related experiments using the proof assistant Coq are reported, for example, in [8].

[6] Unfortunately, due to the complexity of the system, the learning curve for novice users of Isabelle/HOL (or other higher-order proof assistant systems) is still rather steep, and this is one of the main challenges to be overcome in the future to enable better adoption outside the core user community.

[7] The current focus thereby is still on the automation of (fragments of) classical higher-order logic HOL (Church's type theory). However, recent works, such as [70, 71], aim at extending these developments also in direction of supporting more expressive type systems, including dependent types. One of the key drivers of this movement has been the development of an international joint infrastructure for higher-order automated theorem proving [72, 73], in combination with the organization of annual world championship competitions at the CADE and IJCAR conferences [74]. The introduction of TPTP standards also fostered the development, resp. comparison, of proof hammering systems such as Sledgehammer and Hol(y)Hammer [75].

```
theory SurjectiveCantor  imports Main
begin
—‹Surjective Cantor theorem: traditional interactive proof›
theorem SurjectiveCantor:  "¬(∃G.∀F::'a⇒bool.∃X::'a. G X = F)"
 proof
  assume 1: "∃G.∀F::'a⇒bool.∃X::'a. G X = F"      —‹assume surjective G exist ... show contradiction›
  obtain g::"'a⇒('a⇒bool)" where                    —‹fix such a mapping G, call it g›
              2: "∀F.∃X. g X = F" using 1 by auto    —‹this g is surjective by assumption›
  let ?F = "λX.¬ g X X"                              —‹consider ?F = {X | ¬ X ∈ (g X)} — diagonalization›
  have     3: "∃Y. g Y = ?F" using 2 by metis        —‹obviously, there is Y s.t. g Y = ?F — since g surjective›
  obtain a::'a where                                  —‹fix such a Y, call it y›
              4: "g a = ?F" using 3 by auto           —‹obviously, g a = ?F ›
  have     5: "g a a = ?F a"  using 4 by metis       —‹obviously, g a a = ?F a — by functional extensionality›
  have     6: "g a a = (¬ g a a)" using 5 by auto    —‹hence, g a a = ¬ g a a  — by def. of ?F›
  show False using 6 by auto                          —‹thus, contradiction›
 qed
—‹Avoiding proof by contradiction (Fuenmayor & Benzmüller); doi: 10.13140/RG.2.2.31069.95201/1›
theorem SurjectiveCantor':    "¬(∃G.∀F::'a⇒bool.∃X::'a. G X = F)"
 proof -
  {fix g :: "'a⇒('a⇒bool)"
   have 1: "∀X.∃Y.(¬g X Y) = (¬g Y Y)" by auto                  —‹trivial statement: choose Y=X›
   have 2: "∀X.∃Y.(¬g X Y) = ((λZ.¬g Z Z) Y)" using 1 by auto   —‹by λ-conversion & replacement›
   have 3: "∃F.∀X.∃Y.(¬g X Y) = (F Y)" using 2 by auto         —‹∃-introduction applied to (λZ.¬ g Z Z)›
   have 4: "∃F.∀X.¬(∀Y.(g X Y) = (F Y))" using 3 by auto       —‹pull negation outwards›
   have     "∃F.∀X.¬(g X = F)" using 4 by metis                 —‹by functional extensionality›
   }
  hence 5: "∀G.∃F::'a⇒bool.∀X::'a.¬(G X = F)" by auto           —‹∀-introduction: g was chosen arbitrary›
  have   6: "¬(∃G.∀F::'a⇒bool.∃X::'a. G X = F)" using 5 by auto —‹pull negation outwards›
  thus ?thesis .                                                  —‹done, avoiding proof by contradiction›
qed
—‹Surjective Cantor theorem: automated proof by some internal/external theorem provers›
theorem SurjectiveCantor'':    "¬(∃G.∀F::'a⇒bool.∃X::'a. G X = F)"
  nitpick                                     —‹no counterexample found›
  sledgehammer                                —‹most internal provers give up›
  sledgehammer[remote_leo2 remote_leo3]      —‹proof found — external leo provers succeed›
  oops
—‹Surjective Cantor theorem (wrong formalization attempt): the types are crucial›
theorem SurjectiveCantor''':    "¬(∃G.∀F::'b.∃X::'a. G X = F)"
  nitpick                 —‹counterexample found for card 'a = 1 and card 'b = 1: G =  G =  (λx. (a1 := b1)›
  nitpick[satisfy]        —‹model  found for card 'a = 1 and card 'b = 2›
  nitpick[card 'a=2, card 'b=3]      —‹no counterexample found›
  oops
end
```

```
Nitpicking formula...
Nitpick found a counterexample for card 'a = 1 and card 'b = 1:
  Skolem constants:
    G = (λx. _)(a₁ := b₁)
    λF. X = (λx. _)(b₁ := a₁)
```

Fig. 2 Isar style proofs and automation attempts for the surjective Cantor theorem in Isabelle/HOL

all objects of type $'a$). Applications of terms with set/predicate types to terms t of type $'a$, such as $((\lambda X{::}'a.X{=}X)\ t{::}'a)$, can thus also be seen as tests for elementhood: $t{::}'a \in \{X{::}'a|True\}$ (see also the corresponding definitions in lines 4–10 of Fig. 5).

HOL with standard semantics expects the domains $D_{\alpha\Rightarrow\beta}$ to contain *all* functions from D_α to D_β, but this is not required for Henkin's generalized notion of semantics [89], in which the $D_{\alpha\Rightarrow\beta}$ may omit some functions from the full function spaces.[8] For the latter notion of semantics complete proof calculi exist, and are being used in theorem provers as mentioned.[9] For standard semantics, however, as we know from Gödel's first incompleteness theorem [92], the set of valid formulas is not recursively enumerable. Relevant, however, is the observation that every standard model is also a Henkin model, so that Henkin validity implies standard validity. In other words, if we

---

[8] A relevant condition, however, is that these generalized function domains still contain sufficiently many functions so that all terms in the language of HOL have denotations.

[9] For further details on Henkin semantics and associated proof calculi see e.g. [24, 26, 27, 90, 91].

use a sound and complete calculus for Henkin semantics to prove a formula valid in the Henkin sense (e.g. by using some of our automated higher-order theorem provers as listed above), then it is also valid in the standard sense.[10]

Isabelle/HOL[11] is a document-oriented proof assistant system for HOL, which processes input files as shown in Fig. 2 top down. Errors and failures in the verification process, if any, are highlighted in red (for an example, see the red-highlighted keyword sorry in Fig. 8). If there are no red-highlighted lines, then the verification of the formalizations and proofs was successful. In particular, this means that the trusted Isabelle/HOL tools mentioned as justifications in the Isar style proofs shown, such as "auto" and "metis" in Fig. 2 (or "blast", "simp", "force" and "smt" later on) successfully proved the stated propositions from the dependencies mentioned after the "using"-keyword.

The sample formalization in Fig. 2 starts with the introduction of a new theory name, here "SurjectiveCantor", and the import of the theory "Main", which provides HOL and some other theories like the natural numbers, which are, however, not relevant for this paper. Encapsulated in "$-<\ldots>$" and displayed in orange color, we find (e.g. in line 3) an example of a textual comment that is ignored by the Isabelle/HOL verification engine.

The surjective Cantor theorem,[12] which states that there is no surjective map G from a domain $D_{'a}$ to the domain $D_{'a\Rightarrow bool}$ (its powerset), is formalized in line 4 of Fig. 2 and proved interactively using the vernacular Isar proof language (in lines 5–16). The type for map $G{::}'a\Rightarrow('a\Rightarrow bool)$ can be omitted here, since it can be automatically derived by Isabelle/HOL from the types provided for the universally quantified variable $X{::}'a$ and the existentially quantified variable $F{::}'a\Rightarrow bool$.

To illustrate that these type constraints are highly relevant in the given context, we have slightly modified the statement in line 38 of Fig. 2 by replacing $'a\Rightarrow bool$ for the variable X with an arbitrary type $'b$. The modified statement now expresses that there is no surjective map G from a domain $D_{'a}$ to any other domain $D_{'b}$, which is of course false, as the model finder Nitpick tells us by providing a counterexample[13] select $D_{'a}{:=}\{a_1\}$ and $D_{'b}{:=}\{b_1\}$ and now let G be such that it maps a_1 to b_1. Nitpick can also be asked to compute counterexamples for fixed domain cardinalities (cf. line 41, where no counterexample is found when is chosen $card(D_{'a}){=}2$ and $card(D_{'b}){=}3$), and it can be asked to search for satisfying models for the given proposition (cf. line 40).

The proof of the type-correct formalization of the surjective Cantor theorem (in lines 4–16 of Fig. 2) proceeds as expected (see also the comments provided): assume

---

[10] Theoretically, however, some care must be taken when generating counterexamples to formulas in Henkin semantics, since they could in principle be non-standard. However, existing model finders such as Nitpick only generate standard model structures anyway, and these can also be inspected by the user.

[11] The experiments reported in this paper were conducted using the Isabelle2024 version on a MacBookPro 16,1 with a 6-Core Intel Core i7 processor (2.6 GHz) and 16 GB memory. The Isabelle/HOL system is available for download at https://isabelle.in.tum.de/ and a recent tutorial can be found online at https://isabelle.in.tum.de/dist/Isabelle2024/doc/tutorial.pdf; further documentation is provided at https://isabelle.in.tum.de/documentation.html.

[12] For a further discussion of the surjective Cantor theorem and its interpretation in Henkin semantics, see Andrews [24, p.254] and Brown [93, pp.131–150].

[13] See the information in the box as displayed that Isabelle/HOL automatically opens when the cursor is placed on the keyword Nitpick, as shown in line 39 of Fig. 2.

a surjective map g from $D_{'a}$ to $D_{'a \Rightarrow bool}$ exists and choose F as $\lambda X{::}'a.\neg(gX)X$, resp. $F := \{X | X \notin (gX)\}$, which is the key step of the diagonalization proof (and a core challenge for proof automation), then show a contradiction.

An alternative proof of the surjective Cantor theorem, using the same diagonal set for F but avoiding proof by contradiction, is shown in lines 18–30 of Fig. 2; cf. [94]. Moreover, attempts at automated analysis with Nitpick and Sledgehammer are shown in lines 32–36, where first Nitpick reports (in line 33) that it cannot find a (finite) counterexample, before some of the provers integrated with Sledgehammer report that they have found a proof.[14] This is especially true for the external theorem provers Leo-II and Leo-III (here remotely accessed via Sutcliffe's TPTP infrastructure [95] in the US), which prove this theorem robustly and quickly.[15] Unfortunately, the automated proof reconstruction and verification tasks routinely performed by Sledgehammer after proofs are reported to it by some (untrusted) automated theorem provers still fail here. If they were successful, Sledgehammer would e.g. suggest replacing the call to itself with calls to trusted internal tactics (such as auto and metis in lines 13–14 in Fig. 2) instead.[16] So here we have a situation where there is good evidence from several untrusted automated theorem provers that there is indeed a proof for the proposition, but where an interactive proof (such as the one given in lines 4–16) is still needed to finally verify this result.[17]

With this short demonstration and discussion of some features of Isabelle/HOL as relevant for the experiments reported in this article we can now get back to Gödel's ontological proof, the formalization and verification challenge tackled in this article.

Some readers might be tempted to ask why we cannot use HOL directly for encoding Gödel's ontological proof. The answer is that in his manuscript in Fig. 1. Gödel carefully distinguishes between the possible and the necessary existence of God, and for this he uses the alethic modalities M (for possible, resp. "Möglich" in German) and N (for necessary, resp. "Notwendig") [96, 97]. However, since these modalities are non-truth-functional terms that cannot be adequately represented (directly) in a classical logic such as HOL, higher-order modal logic is required.

---

[14] Such information is provided in a special output window area of the Isabelle/HOL proof assistant, which is not shown here.

[15] Since this theorem was already automated in the TPS prover four decades ago, this is what can be expected. Interestingly, however, some provers directly integrated with Isabelle/HOL and Sledgehammer do not report a proof in the call (using a standard parameter setting) in line 34, suggesting opportunities for further system improvements.

[16] There are also mechanisms provided in Isabelle/HOL that attempt to generate complete Isar-style proofs, with the goal of providing more insight to the user.

[17] Such inconclusive proof attempts can be closed with the keyword oops; see line 32 in Fig. 2. In this case, the stated proposition is not added to the library of established results. This is different when the keyword sorry is used instead; see e.g. the red-highlighted sorry in line 28 in Fig. 8). In this case the proposition is added to the library of established results and can referred to in subsequent proof steps. Since this introduces an unverified gap in a theory development the keyword sorry is highlighted in red. We also use the keyword oops in our work to close lines where a model or counterexample to the stated theorem or lemma was computed by the model finder Nitpick; corresponding examples are shown in lines 10–12 in Fig. 4, where Nitpick reports counterexamples.

3 Mechanization of HOML—But which logic exactly?

Although Gödel's 1970 manuscript from Fig. 1 provides some insight into the specific logical setting assumed, the information it provides, or that can be inferred from it, is not exhaustive. Since logical details were also not discussed in the meeting of Gödel and Scott, further interpretation and experimentation is warranted. Different, sometimes controversial, positions have been taken also on the precise logical assumptions by various authors who have commented on the argument or contributed to its development, and experimentation using the logico-pluralistic LogiKEy methodology has already proven its potential for resolving disputes by computation (see, e.g. [32]).

The formalization of higher-order modal logic (HOML), which we will adopt and use in the remainder of this paper, is shown in Fig. 3. The shown *shallow* embedding of HOML in HOL follows the LogiKEy methodology and is based on previous joint work of the first author with Paulson [19].[18] Readers less interested in these formalization details may simply skip this section for now and eventually get back later.

The relevant lines of our Isabelle/HOL formalization of HOML in Fig. 3 are briefly explained:

After importing the theory "Main" (providing HOL) and after declaring some parameter settings for some tools used in our experiments, we start with introducing (in lines 7–8) two base types i (for possible worlds) and e (for entities/individuals). Then the types σ and τ for modal propositions and modal predicates are declared as type synonyms for the predicate types $i{\Rightarrow}bool$ and $e{\Rightarrow}\sigma$ (lines 9–10), respectively. The predicate type σ reflects the core idea of our shallow embedding of HOML in HOL, namely that modal propositions are modeled as predicates over possible worlds, or, when reading these predicates as characteristic functions for sets, as truth sets of possible worlds. Exploiting this idea, modal logic connectives and modal quantifiers can then be declared as simple abbreviations for λ-terms in the language of HOL (lines 15–41) mapping σ-type formulas to σ-type formulas. For the reader not much familiar with Isabelle/HOL, only the equations shown towards the right in Fig. 3 are really relevant for the rest of this paper. Note that we distinguish the newly introduced logical connectives for HOML (such as $\boldsymbol{\vee}$) in our formalization files from their HOL counterparts (such as $\vee$) by the use of bold face fonts (in the text in this paper we don't do this though, since we only refer to modal logic connectives anyway). The terms on the left-hand sides of these equations introduce shorthand notations for the terms on the right-hand sides, which are either proper terms in logic HOL or contain other shorthand notations introduced in some previous line. The logical connective $\vee^e$ (in line 26) is exclusive or, and $\sim$ (line 26) and $\cdot$ (line 27) are negation and conjunction of properties. We introduce $\vee^e$, $\sim$, and $\cdot$ here because they are used in Gödel's 1970 manuscript. Moreover, note that the modal operators $\Box$ and $\Diamond$ are mapped to

---

[18] Shallow embeddings of an object logic (such as HOML) into a metalogic (HOL) are lightweight logic translations that encode the semantics of formulae of the object logic as terms or formulae of the metalogic. This is in contrast to so-called deep embeddings that encode the object logic formulae as uninterpreted data (usually as an inductively defined datatype), and define meta-theoretical notions such as interpretation and satisfiability as functions and predicates. Readers interested in more details and examples of shallow semantical embeddings in HOL may consult [29–31] and the various references therein. For an investigation of the connection between shallow semantical embeddings of non-classical logics (including different notions of quantification) in HOL and topological Boolean algebras we refer to Fuenmayor [98].

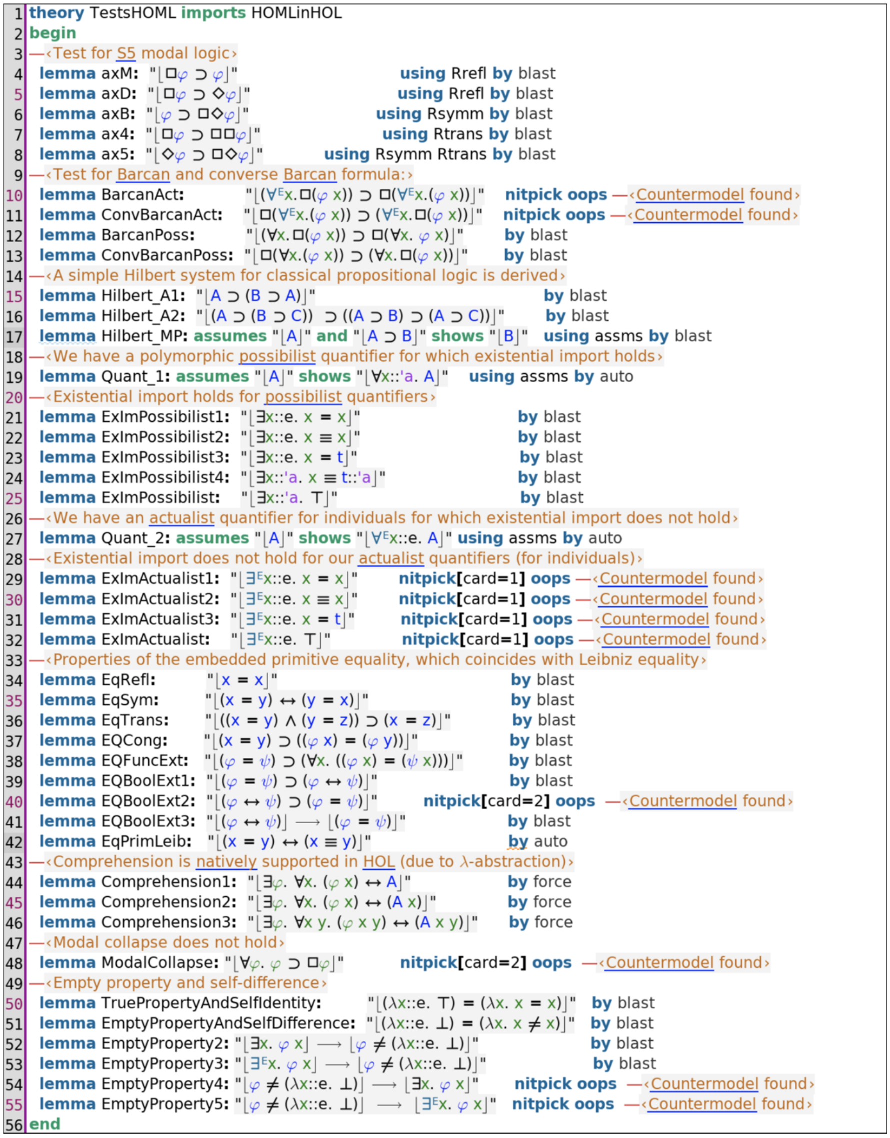
```
 1 theory HOMLinHOL imports Main
 2 begin
 3 —‹Global parameters setting for the model finder nitpick and the parser; unimport for the reader›
 4  nitpick_params[user_axioms,expect=genuine,show_all,format=2,max_genuine=3]
 5  declare[[syntax_ambiguity_warning=false]]
 6 —‹Type i is associated with possible worlds and type e with entities:›
 7  typedecl i     —‹Possible worlds›
 8  typedecl e     —‹Individuals/entities›
 9  type_synonym σ = "i⇒bool"     —‹World-lifted propositions›
10  type_synonym τ = "e⇒σ"        —‹Modal properties›
11  consts R::"i⇒i⇒bool" ("_r_")  —‹Accessibility relation between worlds›
12  axiomatization where
13    Refl: "∀x. xrx"   and   Rsymm: "∀x y. xry ⟶ yrx"   and   Rtrans: "∀x y z. xry ∧ yrz ⟶ xrz"
14 —‹Logical connectives (operating on truth-sets):›
15  abbreviation Mbot::σ ("⊥") where                             "⊥ ≡ λw. False"
16  abbreviation Mtop::σ ("⊤") where                             "⊤ ≡ λw. True"
17  abbreviation Mneg::"σ⇒σ" ("¬_" [52]53) where                "¬φ ≡ λw. ¬(φ w)"
18  abbreviation Mand::"σ⇒σ⇒σ" (infixl "∧" 50) where          "φ∧ψ ≡ λw. φ w ∧ ψ w"
19  abbreviation Mor::"σ⇒σ⇒σ" (infixl "∨" 49) where           "φ∨ψ ≡ λw. φ w ∨ ψ w "
20  abbreviation Mimp::"σ⇒σ⇒σ" (infixr "⊃" 48) where          "φ⊃ψ ≡ λw. φ w ⟶ ψ w"
21  abbreviation Mequiv::"σ⇒σ⇒σ" (infixl "↔" 47) where       "φ↔ψ ≡ λw. φ w ⟷ ψ w"
22  abbreviation Mbox::"σ⇒σ" ("□_" [54]55) where                "□φ ≡ λw.∀v. w r v ⟶ φ v"
23  abbreviation Mdia::"σ⇒σ" ("◇_" [54]55) where                "◇φ ≡ λw.∃v. w r v ∧ φ v"
24  abbreviation Mprimeq::"'a⇒'a⇒σ" ("_=_") where              "x=y ≡ λw. x=y"
25  abbreviation Mprimneq::"'a⇒'a⇒σ" ("_≠_") where             "x≠y ≡ λw. x≠y"
26  abbreviation Mnegpred::"τ⇒τ" ("~_") where                   "~Φ ≡ λx.λw. ¬Φ x w"
27  abbreviation Mconpred::"τ⇒τ⇒τ" (infixl "." 50) where      "Φ.Ψ ≡ λx.λw. Φ x w ∧ Ψ x w"
28  abbreviation Mexclor::"σ⇒σ⇒σ" (infixl "∨ᵉ" 49) where     "φ∨ᵉψ ≡ (φ ∨ ψ) ∧ ¬(φ ∧ ψ)"
29 —‹Possibilist quantifiers (polymorphic)›
30  abbreviation Mallposs::"('a⇒σ)⇒σ" ("∀") where               "∀Φ ≡ λw.∀x. Φ x w"
31  abbreviation Mallpossb (binder "∀" [8]9) where           "∀x. φ(x) ≡ ∀φ"
32  abbreviation Mexiposs::"('a⇒σ)⇒σ" ("∃") where               "∃Φ ≡ λw.∃x. Φ x w"
33  abbreviation Mexipossb (binder "∃" [8]9) where           "∃x. φ(x) ≡ ∃φ"
34 —‹Actualist quantifiers (for individuals/entities)›
35  consts existsAt::"e⇒σ" ("_@_")
36  abbreviation Mallact::"(e⇒σ)⇒σ" ("∀ᴱ") where               "∀ᴱΦ ≡ λw.∀x. x@w ⟶ Φ x w"
37  abbreviation Mallactb (binder "∀ᴱ" [8]9) where          "∀ᴱx. φ(x) ≡ ∀ᴱφ"
38  abbreviation Mexiact::"(e⇒σ)⇒σ" ("∃ᴱ") where               "∃ᴱΦ ≡ λw.∃x. x@w ∧ Φ x w"
39  abbreviation Mexiactb (binder "∃ᴱ" [8]9) where          "∃ᴱx. φ(x) ≡ ∃ᴱφ"
40 —‹Leibniz equality (polymorphic):›
41  abbreviation Mleibeq::"'a⇒'a⇒σ" ("_≡_") where              "x≡y ≡ ∀P. P x ⊃ P y"
42 —‹Meta-logical predicate for global validity:›
43  abbreviation Mvalid::"σ⇒bool" ("⌊_⌋") where                 "⌊ψ⌋ ≡ ∀w. ψ w"
44 end
```

Fig. 3 Shallow embedding of higher-order modal logic (HOML) in the classical higher-order logic (HOL) of Isabelle/HOL utilizing the LogiKEy methodology

λ-terms that use an accessibility guard based on the accessibility relation r between possible worlds (lines 22–23). The relation r itself is introduced as an uninterpreted constant symbol (line 11), which is constrained by axioms so that it denotes an equivalence relation between worlds (line 13), with the effect that □ and ◇ thus denote S5 modal operators; see also the respective tests performed in Fig. 4, where prominent S5 schematic reasoning principles and axioms are validated.[19] Note also that both possibilist quantifiers (cf. the polymorphic declarations in lines 30–33, with $'a$ being a type variable) and actualist quantifiers (lines 35–39, for individuals/entities of type e only) are provided. The latter uses an explicit existence predicate @ as a guard. Polymorphic Leibniz equality is defined by a possibilist universal quantifier (line 41).

---

[19] However, as we will exploit in our experiments, the exact dependencies of proofs and proof steps on individual modal principles like B, D, M, 4 and 5 (or on their semantic counterparts) can be made explicit. In fact, it will turn out that Gödel's argument can be validated using only the B principle (i.e., symmetry of the accessibility relation r).



```
theory TestsHOML imports HOMLinHOL
begin
—‹Test for S5 modal logic›
 lemma axM:  "⌊□φ ⊃ φ⌋"                         using Rrefl by blast
 lemma axD:  "⌊□φ ⊃ ◇φ⌋"                        using Rrefl by blast
 lemma axB:  "⌊φ ⊃ □◇φ⌋"                      using Rsymm by blast
 lemma ax4:  "⌊□φ ⊃ □□φ⌋"                      using Rtrans by blast
 lemma ax5:  "⌊◇φ ⊃ □◇φ⌋"                 using Rsymm Rtrans by blast
—‹Test for Barcan and converse Barcan formula:›
 lemma BarcanAct:          "⌊(∀ᴱx.□(φ x)) ⊃ □(∀ᴱx.(φ x))⌋"   nitpick oops —‹Countermodel found›
 lemma ConvBarcanAct:     "⌊□(∀ᴱx.(φ x)) ⊃ (∀ᴱx.□(φ x))⌋"   nitpick oops —‹Countermodel found›
 lemma BarcanPoss:         "⌊(∀x.□(φ x)) ⊃ □(∀x. φ x)⌋"      by blast
 lemma ConvBarcanPoss:  "⌊□(∀x.(φ x)) ⊃ (∀x.□(φ x))⌋"      by blast
—‹A simple Hilbert system for classical propositional logic is derived›
 lemma Hilbert_A1:  "⌊A ⊃ (B ⊃ A)⌋"                              by blast
 lemma Hilbert_A2:  "⌊(A ⊃ (B ⊃ C))  ⊃ ((A ⊃ B) ⊃ (A ⊃ C))⌋"     by blast
 lemma Hilbert_MP: assumes "⌊A⌋" and "⌊A ⊃ B⌋" shows "⌊B⌋"  using assms by blast
—‹We have a polymorphic possibilist quantifier for which existential import holds›
 lemma Quant_1: assumes "⌊A⌋" shows "⌊∀x::'a. A⌋"    using assms by auto
—‹Existential import holds for possibilist quantifiers›
 lemma ExImPossibilist1:  "⌊∃x::e. x = x⌋"                     by blast
 lemma ExImPossibilist2:  "⌊∃x::e. x ≡ x⌋"                     by blast
 lemma ExImPossibilist3:  "⌊∃x::e. x = t⌋"                     by blast
 lemma ExImPossibilist4:  "⌊∃x::'a. x ≡ t::'a⌋"                by blast
 lemma ExImPossibilist:   "⌊∃x::'a. ⊤⌋"                        by blast
—‹We have an actualist quantifier for individuals for which existential import does not hold›
 lemma Quant_2: assumes "⌊A⌋" shows "⌊∀ᴱx::e. A⌋" using assms by auto
—‹Existential import does not hold for our actualist quantifiers (for individuals)›
 lemma ExImActualist1:  "⌊∃ᴱx::e. x = x⌋"      nitpick[card=1] oops —‹Countermodel found›
 lemma ExImActualist2:  "⌊∃ᴱx::e. x ≡ x⌋"      nitpick[card=1] oops —‹Countermodel found›
 lemma ExImActualist3:  "⌊∃ᴱx::e. x = t⌋"      nitpick[card=1] oops —‹Countermodel found›
 lemma ExImActualist:   "⌊∃ᴱx::e. ⊤⌋"          nitpick[card=1] oops —‹Countermodel found›
—‹Properties of the embedded primitive equality, which coincides with Leibniz equality›
 lemma EqRefl:        "⌊x = x⌋"                                 by blast
 lemma EqSym:         "⌊(x = y) ↔ (y = x)⌋"                     by blast
 lemma EqTrans:       "⌊((x = y) ∧ (y = z)) ⊃ (x = z)⌋"         by blast
 lemma EQCong:        "⌊(x = y) ⊃ ((φ x) = (φ y))⌋"             by blast
 lemma EQFuncExt:   "⌊(φ = ψ) ⊃ (∀x. ((φ x) = (ψ x)))⌋"      by blast
 lemma EQBoolExt1:  "⌊(φ = ψ) ⊃ (φ ↔ ψ)⌋"                     by blast
 lemma EQBoolExt2:  "⌊(φ ↔ ψ) ⊃ (φ = ψ)⌋"       nitpick[card=2] oops  —‹Countermodel found›
 lemma EQBoolExt3:  "⌊(φ ↔ ψ)⌋  ⟶ ⌊(φ = ψ)⌋"                 by blast
 lemma EqPrimLeib:   "⌊(x = y) ↔ (x ≡ y)⌋"                     by auto
—‹Comprehension is natively supported in HOL (due to λ-abstraction)›
 lemma Comprehension1:  "⌊∃φ. ∀x. (φ x) ↔ A⌋"                 by force
 lemma Comprehension2:  "⌊∃φ. ∀x. (φ x) ↔ (A x)⌋"             by force
 lemma Comprehension3:  "⌊∃φ. ∀x y. (φ x y) ↔ (A x y)⌋"       by force
—‹Modal collapse does not hold›
 lemma ModalCollapse: "⌊∀φ. φ ⊃ □φ⌋"          nitpick[card=2] oops  —‹Countermodel found›
—‹Empty property and self-difference›
 lemma TruePropertyAndSelfIdentity:          "⌊(λx::e. ⊤) = (λx. x = x)⌋"   by blast
 lemma EmptyPropertyAndSelfDifference:  "⌊(λx::e. ⊥) = (λx. x ≠ x)⌋"   by blast
 lemma EmptyProperty2: "⌊∃x. φ x⌋ ⟶ ⌊φ ≠ (λx::e. ⊥)⌋"                 by blast
 lemma EmptyProperty3: "⌊∃ᴱx. φ x⌋ ⟶ ⌊φ ≠ (λx::e. ⊥)⌋"                by blast
 lemma EmptyProperty4: "⌊φ ≠ (λx::e. ⊥)⌋ ⟶ ⌊∃x. φ x⌋"      nitpick oops  —‹Countermodel found›
 lemma EmptyProperty5: "⌊φ ≠ (λx::e. ⊥)⌋  ⟶  ⌊∃ᴱx. φ x⌋"  nitpick oops  —‹Countermodel found›
end
```

Fig. 4 Tests and verifications of properties for the embedding of HOML in HOL from Fig. 3

Finally, the validity of embedded, type-lifted formulas φ of type σ, denoted as $\lfloor\varphi\rfloor$, is defined in the obvious way by asking for the truth of φ for all possible worlds (line 43); thus, instead of $\lfloor\varphi\rfloor$ we could have also used $\models \varphi$ as notation. Since it can be shown that this embedding of HOML in HOL is faithful (sound and complete for Henkin's generalized semantics; see e.g. [19]), we can now use the tools provided in Isabelle/HOL to automate reasoning in HOML and develop e.g. Isar style proofs in HOML interactively.

The main properties of the presented embedding of HOML in HOL are briefly summarized:

- The alethic operators $\Box$ and $\Diamond$ are modeled as the operators of an S5 modal logic. In the formal proofs presented in this paper, dependencies on certain S5 principles are explicitly mentioned, and it turns out that most proofs steps actually require only base logic K, and some other steps require logic KB. The introduction of all S5 principles is thus essentially unnecessary.
- The interpretation of the logical connectives $\bot$, $\top$, $\neg$, $\vee$, $\wedge$, $\Rightarrow$, $\leftrightarrow$ and $\vee^e$ (exclusive or) is classical; in addition, the logic provides a (world independent, polymorphic) primitive (in-)equality ($=$ and $\neq$) and Leibniz equality ($\equiv$).
- Exploiting the flexibility of the LogiKEy approach two different notions of modal quantifiers [22, 97, 99–101] are introduced:
  - Possibilist quantifiers $\forall$ and $\exists$, corresponding to constant domain semantics, are provided for quantification over arbitrary types (polymorphic definitions are given).
  - Actualist quantifiers $\forall^E$ and $\exists^E$, corresponding to varying domain semantics, are provided for quantification over the domain of individuals/entities only; to model those an explicit existence predicate @ (read $x@w$ as x exists in world w) is used as a relativizing guard in their definitions. This modeling of actualist quantifiers is related to quantification in free logics [102–104], whose formalization using the LogiKEy approach has been explored in related work [105].

  In line with e.g. Fitting [17], in our subsequent formalizations we use by default the actualist quantifiers for quantification over individuals and the possibilist quantifiers for quantification over propositions and properties. However, we also add experiments that consider only possibilist quantifiers or a mixed used of actualist and possibilist quantifiers for individuals.[20]
- Existential import holds for the possibilist quantifiers, but not for the actualist ones. Thus later in the main theorem, when we use an existential actualist quantifier to state that there is an x that has the property of being God-like, we do not assume that there is any individual a priori.
- Validity of a modal logic formula φ, denoted in this paper as $\lfloor\varphi\rfloor$, is modeled as truth of φ in all possible worlds.
- As intended, the embedded modal logic HOML does not have modal collapse.

The above statements are briefly illustrated with some corresponding tests and verifications in Fig. 4 (note that the file “HOMLinHOL” from Fig. 3 is imported in line 1): The prominent axiom schemata M, D, B, 4, and 5, which are all valid in an S5

---

[20] Anderson [12, p.301], for example, motivates a mixed use of actualist and possibilist quantifiers for individuals in Gödel’s argument: *“Note well that the appropriate translation of the argument into Cocchiarella’s notation will take the existential quantifier in the definition of necessary existence to be an e-quantifier (an “existence”, rather than “subsistence”, quantifier) and so too the quantifier in the conclusion of the argument. All others individual quantifiers may be taken to be subsistential and hence to range over all possibles.”* See the respective experiments mentioned in Footnotes 25 and 29, and in Subsection 5.2.

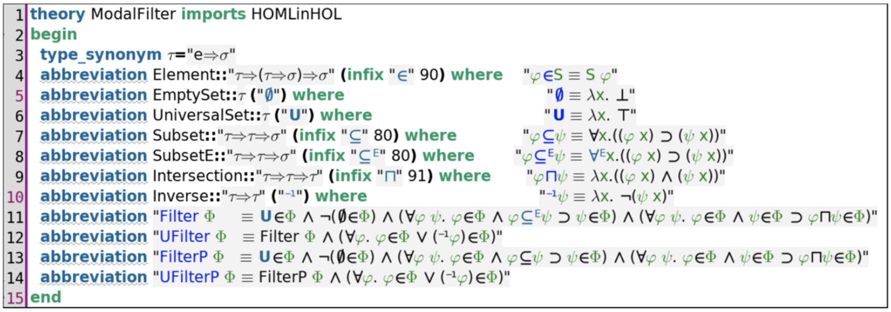
modal logic, are proved as implied lemmata (in lines 4–8) by Isabelle's in-built proof automation routine "blast" (a trusted tableaux prover).

Note that the variable symbol φ (and ψ and x later on), displayed in light blue color, are here (implicitly universally quantified) variables in the metalogic HOL. Equivalently, we could have explicitly quantified over these variables in our logic HOML using our (polymorphic) possibilist universal quantifiers from Fig. 3. This comment applies to all such occurrences of unbound variable symbols in our subsequent formalizations.

The Barcan formulas hold for the possibilist quantifiers (lines 12–13), but not for the actualist quantifiers (lines 10–11); this is confirmed by the model finder Nitpick [68] integrated with Isabelle/HOL, which reports simple counterexamples (not shown here) consisting of two worlds i_1 and i_2 and one entity e_1. The derivability of a simple Hilbert system (consisting of axiom schemata A1-A4 and the schematic modus ponens rule for classical propositional logic) illustrates that the interpretation of the embedded logical connective $\supset$ is classical (lines 15–16); again, the proofs are obtained automatically using "blast" and "simp" (a trusted simplifier of Isabelle/HOL). Existential import holds for possibilist existential quantification $\exists$ (lines 21–25). Existential import does not hold for the actualist existential quantification $\exists^E$ (lines 29–32); Nitpick reports an obvious counterexample for all existential import lemmata shown here: consider a world i_1 and an individual e_1, so that e_1 does not exist in i_1, i.e., so that $e_1@i_1$ is false, where @ is the explicit existence predicate used as guard in the definition of $\exists^E$. The rigidly embedded primitive equality = is shown to be an equivalence relation, satisfying also congruence and functional extensionality (lines 34–38). As expected for a modal setting, Boolean extensionality fails to hold in the non-trivial direction from equivalence to equality (line 40), since otherwise substitution of equivalents in arbitrary modal formulas would be supported; Nitpick reports a counterexamples in this case with two possible worlds i_1 and i_2, where φ is false in i_2 and ψ is true in i_2, while they are are both false in the actual world i_1. The trivial opposite direction of Boolean extensionality is proven valid (line 41). Leibniz equality and primitive equality coincide (line 42). Full comprehension[21] is natively supported in our embedding of HOML in HOL (lines 44–46, these lines are proven with another trusted tactic of Isabelle/HOL, called "force"); this is due to the native support for λ-abstractions in the metalanguage HOL. For modal collapse (line 48) Nitpick reports a simple counterexample: consider two mutually (self-)connected worlds i_1 and i_2, choose formula φ to hold in the actual world $i1$ but not in world i_2. We also present some tests regarding some specific modal properties that will be of interest later: Self-identity $\lambda x.\, x{=}x$ is equivalent to the tautologous, universally true property $\lambda x.\, \top$ (line 50). Analogously, self-difference $\lambda x.\, x{\neq}x$ is equivalent to the absurd, empty property $\lambda x.\, \bot$ (line 51). Further tests illustrate that exemplification statements on φ (using possibilist or actualist quantifiers) do imply that φ is not the empty property (lines 53–54), but the opposite does not hold (lines 54–55).

---

[21] The LogiKEy methodology also supports embeddings of object logics lacking full comprehension; see, for example, the work by Kirchner et al. [106, 107]. Computer-supported investigations of Gödel's ontological argument using the LogiKEy methodology for object logics without full comprehension is future work though.



```
 1 theory ModalFilter imports HOMLinHOL
 2 begin
 3   type_synonym τ="e⇒σ"
 4   abbreviation Element::"τ⇒(τ⇒σ)⇒σ" (infix "∈" 90) where    "φ∈S ≡ S φ"
 5   abbreviation EmptySet::τ ("∅") where                          "∅ ≡ λx. ⊥"
 6   abbreviation UniversalSet::τ ("U") where                      "U ≡ λx. ⊤"
 7   abbreviation Subset::"τ⇒τ⇒σ" (infix "⊆" 80) where          "φ⊆ψ ≡ ∀x.((φ x) ⊃ (ψ x))"
 8   abbreviation SubsetE::"τ⇒τ⇒σ" (infix "⊆ᴱ" 80) where       "φ⊆ᴱψ ≡ ∀ᴱx.((φ x) ⊃ (ψ x))"
 9   abbreviation Intersection::"τ⇒τ⇒τ" (infix "⊓" 91) where    "φ⊓ψ ≡ λx.((φ x) ∧ (ψ x))"
10   abbreviation Inverse::"τ⇒τ" ("⁻¹") where                      "⁻¹ψ ≡ λx. ¬(ψ x)"
11   abbreviation "Filter Φ    ≡ U∈Φ ∧ ¬(∅∈Φ) ∧ (∀φ ψ. φ∈Φ ∧ φ⊆ᴱψ ⊃ ψ∈Φ) ∧ (∀φ ψ. φ∈Φ ∧ ψ∈Φ ⊃ φ⊓ψ∈Φ)"
12   abbreviation "UFilter Φ   ≡ Filter Φ ∧ (∀φ. φ∈Φ ∨ (⁻¹φ)∈Φ)"
13   abbreviation "FilterP Φ   ≡ U∈Φ ∧ ¬(∅∈Φ) ∧ (∀φ ψ. φ∈Φ ∧ φ⊆ψ ⊃ ψ∈Φ) ∧ (∀φ ψ. φ∈Φ ∧ ψ∈Φ ⊃ φ⊓ψ∈Φ)"
14   abbreviation "UFilterP Φ ≡ FilterP Φ ∧ (∀φ. φ∈Φ ∨ (⁻¹φ)∈Φ)"
15 end
```

Fig. 5 Set filter and ultrafilter formalized for our modal logic setting

Since in this paper we will also study the relation of Gödel's positive properties to set ultrafilters, we introduce corresponding formalizations of these notions in our modal logic context; see Fig. 5. Again we utilize the LogiKEy methodology and introduce the respective notions as shorthand notations in HOL. Note, that the logical connectives used in these shorthand definitions now refer to HOML logical connectives, which are themselves introduced as shorthand notation for HOL terms in Fig. 3. The notions introduced are modal variants (note the σ-type) of elementhood (line 4), empty set (line 5), universal set (of objects of type e, line 6), subset relation stated with possibilist and actualist quantifiers (lines 7–8), set intersection (line 9) and inverse relation (line 10); note that these definitions utilize the idea (as mentioned in Sect. 2) to encode sets as characteristic functions. A (modal) set filter is then defined as a set of sets (resp. set of properties) Φ containing the universal set but not the empty set and which is closed for supersets and set intersection (line 11); note that in this shorthand notation the subset relation formulated with actualist quantification is used. An analogous shorthand notation is introduced (in line 13), called "FilterP", using the subset relation with possibilist quantification. A (modal) set ultrafilter is then introduced as a (modal) set filter that is maximal (line 12, resp. line 14).

4 Gödel's ontological argument—1970 manuscript

In this section we present a formal study of Gödel's ontological argument as it appears in his 1970 manuscript; see Fig. 1. In contrast to other prior works, we stick here as closely as possible to this manuscript of Gödel and explain any deviations where necessary. The detailed properties of the logical presuppositions were discussed in the previous section. We explain our formalizations as shown in Figs. 6, 7 and 8 in some detail below, address relevant questions along the way, and implicitly provide answers to some frequently asked questions. Note that the formalizations given in these figures coincide up to line 22, where axiom A4 is stated, with the following important exceptions (marked with "(**)" in red): in Fig. 7 the notion of essence is changed, and in Fig. 8 the notion of necessary property implication. While Fig. 6 demonstrates an inconsistency in Gödel's assumptions starting from line 23, this is not the case for the slightly modified modelings presented in Figs. 7 and 8, where consistency is shown in line 23 and the logical validity of Gödel's ontological argument is subsequently confirmed. Note that the presented results also hold when only possibilist quantifiers are used in the formalization. This is documented in Figs. 16, 17 and 18 in Appendices A-C (where relevant modifications are again marked with "(**)" in red, this time with respect to their actualist counterparts in Figs. 6, 7 and 8). Further experiments, as briefly mentioned in Footnotes 25 and Footnote 29 but not displayed in this document, also confirm our results for a mixed use of quantifiers.

We now discuss the formal modeling of Gödel's axioms and definitions step by step.

4.1 Formalization of Gödel's axioms and definitions

Positive properties: P

The only primitive symbol in Gödel's manuscript is P, for "positive", and P(φ) stands for φ is a positive property. A relevant question is whether Gödel intended P to range over extensions or intensions of properties. In our modeling the type declaration for P is chosen such that P ranges over intensions of properties (cf. line 3 in Figs. 6, 7 and 8): P is applied to terms φ of type $e \Rightarrow \sigma$, that is, modal predicates (recall that type σ in our embedding of HOML in HOL stands for the predicate type $i \Rightarrow$`bool`, where i is the type of possible worlds). The modal predicates φ, when applied to an argument term of type e (remember that this type is denoting the domain of individuals/entities) thus constitutes a world-dependent modal term of type σ and it may thus be evaluated differently for different possible worlds. In this sense, our formalization is such that it models P to range over intensions of properties. In Scott's meeting with Gödel, this issue was not discussed further, and the interpretation chosen here is also consistent with what various other authors have found to be an appropriate choice.[22]

Axiom Ax1: $P\varphi \wedge P\psi \supset P(\varphi\,.\,\psi)$

This axiom states that the conjunction of any two positive properties is positive (line 4).[23] Note that Gödel has a footnote in his manuscript adding *"and for any number of summands"*. As we will later see Gödel's comment to generalize this axiom so that it includes also infinite number of conjuncts is actually relevant for deriving the possible and necessary existence of a God-like entity. Therefore, in Fig. 7 we will generalize Axiom Ax1 to Axiom Ax1Gen (line 24–26) to formally address and analyze this footnote. Such a generalized axiom in third-order logic has been suggested

---

[22] Fitting [17], however, assumes that P should range over extensions (resp. rigidly intensionalised extensions) of properties; in fact, the connection between the two choices and their different effects on the validity of modal collapse has been studied in detail, using the LogiKEy methodology, by Benzmüller and Fuenmayor [35].

[23] For the conjunction symbol between modal propositions shown on the left in this formula, Gödel uses a "." symbol in his manuscript; in this article we have decided to use $\wedge$ instead. However, we keep the "." on the right, which denotes the conjunction of properties; see also line 27 in Fig. 3. Also, we generally avoid parentheses and write P φ instead of P(φ). In addition, we could alternatively use possibilist universal quantifiers in the encoding to explicitly quantify over φ and ψ. To stay close to Gödel's manuscript, however, we formalize them here as schematic variables in the metalanguage HOL; technically, after unfolding all embedded terms in the metalogic HOL, the difference boils down to an alternation in the leading universal quantifiers of this formula; this remark applies similarly to other schematic axioms in the rest of this article.

by Anderson and Gettings [13, p. 170]. However, a related proposal can already be found in earlier notes by Gödel on the ontological argument, which were recently discovered in his Nachlass; see Kanckos and Lethen [4, p. 1019]. As it turns out, for the proof that Gödel's original axioms and definitions in HOML are inconsistent (see Sect. 4.2 below), it is irrelevant whether Axiom Ax1 or the more general Axiom Ax1Gen is considered.

Note that all axioms, lemmas, and theorems in the Isabelle/HOL encodings of the various arguments studied in this article are always wrapped in the validity operator "$\lfloor\ \rfloor$". We thus postulate their validity for all possible worlds, i.e. we model them as "global" modal assumptions.

Axiom Ax2a: $\boldsymbol{P}\varphi \vee^e \boldsymbol{P}{\sim}\varphi$

The label "Ax.2" actually occurs twice in Gödel's manuscript and we therefore distinguish these two parts in the manuscript by labeling the first occurrence Ax2a and the second one Ax2b. Ax2a states that either a property φ or its negation ${\sim}\varphi$ is positive. Using our logical connective $\vee^e$, for exclusive or (cf. Gödel's comment on the left bottom in the manuscript), this axiom is straightforward to encode (line 5).

Definition of God-like: $\boldsymbol{G}\,x \equiv \forall\varphi.\, \boldsymbol{P}\varphi \supset \varphi\, x$

For Gödel an entity x has the property of being God-like (G) if (and only if) that entity possesses all positive properties. Using a possibilist universal quantifier $\forall$ ranging over properties φ this can easily be encoded as shown (line 6).

Definition of Essence: $\varphi\, \textbf{\textit{Ess.}}\, x \equiv \forall\psi.\, \psi\, x \supset (\varphi \supset_N \psi)$

For Gödel (in the tradition of the work of Leibniz, whom Gödel studied extensively), entities are uniquely determined by their essential properties, by which all their other properties are implied. This idea is captured in Gödel's definition of essence (Ess.; read $\varphi\,\mathrm{Ess.}\,x$ as φ is an essential property of entity x): φ is an essential property of entity x if (and only if) for all (possibly other) properties ψ that x also has, it holds that ψ is necessarily implied by φ.[24] The latter condition, necessary property implication $\varphi \supset_N \psi$, mentioned in Gödel's manuscript on the left, will also be used in the axiom Ax4 below. $\varphi \supset_N \psi$ is formalized as (line 7, see also Fitting [17])

$$\Box(\forall^E y.\, \varphi\, y \supset \psi\, y)$$

Note that our entire formalization of essence (in line 8 of Fig. 6) thus uses a possibilist universal quantifier to quantify over the properties ψ, and an actualist universal quantifier to quantify over the entities y in $\varphi \supset_N \psi$ (but remember that we have conducted alternative experiments as mentioned in Appendices A-C and Footnotes 25 and 29). Note also that Gödel does not explicitly require in this definition that property φ applies to x, which leads to an inconsistency in his postulates, as explained in Sect. 4.2 below, and which will therefore be modified in Sects. 4.3 and 4.4 to arrive at a consistent set of assumptions. Alternatively, one could argue that the necessary property implication $\varphi \supset_N \psi$ in this definition of essence should be formalized differently, so

---

[24] As noted before, we use the symbol $\Box$ for necessity (N) and $\Diamond$ for possibility (M).

```
 1 theory GoedelVariantHOML1 imports HOMLinHOL
 2 begin
 3   consts PositiveProperty::"(e⇒σ)⇒σ" ("P")
 4   axiomatization where Ax1: "⌊P φ ∧ P ψ ⊃ P (φ . ψ)⌋"
 5   axiomatization where Ax2a: "⌊P φ ∨ᵉ P ~φ⌋"
 6   definition God ("G") where "G x ≡ ∀φ. P φ ⊃ φ x"
 7   abbreviation PropertyInclusion ("_⊃N_") where "φ ⊃N ψ  ≡  □(∀ᴱy.  φ y ⊃ ψ y)"
 8   definition Essence ("_Ess._") where "φ Ess. x ≡ ∀ψ. ψ x ⊃ (φ ⊃N ψ)"
 9   axiomatization where Ax2b: "⌊P φ ⊃ □ P φ⌋"
10   lemma Ax2b': "⌊¬P φ ⊃ □(¬P φ)⌋" using Ax2a Ax2b by blast
11   theorem Th1: "⌊G x ⊃ G Ess. x⌋" using Ax2a Ax2b Essence_def God_def by (smt (verit))
12   definition NecExist ("E") where "E x  ≡ ∀φ. φ Ess. x ⊃ □(∃ᴱx. φ x)"
13   axiomatization where Ax3: "⌊P E⌋"
14   theorem Th2: "⌊G x ⊃ □(∃ᴱy. G y)⌋" using Ax3 Th1 God_def NecExist_def by smt
15   theorem Th3: "⌊◇(∃ᴱx. G x) ⊃ □(∃ᴱy. G y)⌋" sledgehammer(Th2 Rsymm)  —‹Proof found›
16      proof -
17       have 1: "⌊(∃ᴱx. G x) ⊃ □(∃ᴱy. G y)⌋" using Th2 by blast
18       have 2: "⌊◇(∃ᴱx. G x) ⊃ ◇□(∃ᴱy. G y)⌋" using 1 by blast
19       have 3: "⌊◇(∃ᴱx. G x) ⊃ □(∃ᴱy. G y)⌋" using 2 Rsymm by blast
20       thus ?thesis by blast
21     qed
22   axiomatization where Ax4: "⌊P φ ∧ (φ ⊃N ψ) ⊃ P ψ⌋"
23   lemma True nitpick[satisfy] oops  —‹No model found›
24   lemma EmptyEssL: "⌊(λy.⊥) Ess. x⌋" using Essence_def by auto
25   theorem Inconsistency: False sledgehammer(Ax2a Ax3 Ax4 EmptyEssL NecExist_def) —‹Proof found›
26     proof -
27       have 1: "⌊¬(P (λx.⊥))⌋" using Ax2a Ax4 by blast
28       have 2: "⌊P (λx.(λy.⊥) Ess. x ⊃ □(∃ᴱz. (λy.⊥)z))⌋" using Ax3 Ax4 NecExist_def by smt
29       have 3: "⌊P (λx.□(∃ᴱz. (λx.⊥) z))⌋" using 2 EmptyEssL by simp
30       have 4: "⌊P (λx.□⊥)⌋" using 3 by auto
31       have 5: "⌊P (λx.⊥)⌋" using 4 Ax2a Ax4 by smt
32       have 6: "⌊⊥⌋" using 1 5 by blast
33       thus ?thesis by blast
34     qed
35 end
```

Fig. 6 Gödel's axioms and definitions, as presented in the 1970 manuscript, are inconsistent

that the absurd, empty property does not necessarily imply all other properties. Our new results on this option to fix the inconsistency in Gödel's assumptions, which were experimentally explored, are reported in Sect. 4.5 below.

Axiom Ax2b: $P\,\varphi \supset \Box P\,\varphi$ and $\neg P\,\varphi \supset \Box(\neg P\,\varphi)$

Gödel provides two versions of this axioms, which we label Ax2b and Ax2b'. The former (in line 9) states that every positive property is necessarily positive, and the latter that every non-positive property is necessarily non-positive. Since Ax2b' is actually implied by Ax2a and Ax2b, as our experiments confirm, we introduce Ax2b' as a lemma (line 10) and prove it with the trusted theorem prover blast. Our experiments also show that Ax2b' is not needed at all (it is not referred to again in Figs. 6, 7 and 8 or Figs. 16, 17 and 18) and can thus be avoided altogether.

Theorem Th1: $G\,x \supset G\,\mathit{Ess.}\,x$

This theorem states that the property of being God-like is an essence of any God-like entity (line 11). It easily follows from Ax2a and Ax2b using the definitions of essence and being God-like (here the trusted SMT solver "verit" built into Isabelle/HOL proves this result using the stated dependencies).

Definition of Necessary Existence: $\boldsymbol{E}\,x \equiv \forall\varphi.\, \varphi\, \boldsymbol{Ess.}\, x \supset \Box(\exists^E x.\, \varphi\, x)$

For Gödel, an entity x has the property of necessary existence (E) if (and only if) all essences of x are necessarily exemplified (line 12). The reader may be tempted to criticize that this notion introduces some kind of circularity into the argument. Note, however, that the exemplification statement $\Box(\exists^E x.\, \varphi\, x)$, which can obviously be replaced by $\Box(\exists^E z.\, \varphi\, z)$, refers to arbitrary exemplifying entities x, resp. z, which here could be different from the entity x to which the predicate E is applied.

Axiom Ax3: $\boldsymbol{P}\,\boldsymbol{E}$

This axiom states that E, necessary existence, is a positive property (line 13).

Theorem Th2: $\boldsymbol{G}\,x \supset \Box(\exists^E y.\, \boldsymbol{G}\, y)$

If x is God-like, then a God-like entity necessarily exists (line 14). This can be easily proved from axiom Ax3 and theorem Th1 using the definitions of God-like and necessary existence; in our Isabelle/HOL formalization this proof is contributed by a trusted, smt solver. Note that the existential quantifier used is an actualist quantifier.

Gödel then derives in his manuscript the following formula, which we call theorem Th3 and formalize using actualist quantifiers:

Theorem Th3: $\Diamond(\exists^E x.\, \boldsymbol{G}\, x) \supset \Box(\exists^E y.\, \boldsymbol{G}\, y)$

If a God-like entity possibly exists, then a God-like entity necessarily exists (line 15). This follows from Theorem Th2 in modal logic S5, as we confirm here both with Sledgehammer and with an interactive Isar style proof that is following exactly the steps in Gödel's manuscript (lines 16–21), which are then confirmed individually with calls to the trusted theorem prover blast. It is important to note that in the proof constructed in Isabelle/HOL, only the symmetry of the accessibility relation, corresponding to the modal axiom scheme B, is actually required. Hence, S5 modal logic is not needed, modal logic KB is already sufficient. A relevant question that arises is whether logic S4 is eventually also suited to for proving theorem T3 ? However, further experiments, which are presented in Appendix A answer this negatively in the given context, since Nitpick produces a counterexample.

It thus remains for Gödel to prove that a God-like entity possibly exists. For this he introduces another axiom.

Axiom Ax4: $\boldsymbol{P}\,\varphi \wedge (\varphi \supset_N \psi) \supset \boldsymbol{P}\,\psi$

Every property ψ that is necessarily implied by a positive property φ is positive (line 22). Here we reuse the modeling of $\varphi \supset_N \psi$ as already discussed above for the formalization of essence.[25]

After postulating Ax4, Gödel actually arrives at an inconsistent set of assumptions in classical HOML (lines 23–34), which he apparently was not aware of. This inconsistency is analyzed and explained further in the next subsection.

---

[25] Additional experiments have shown that we could instead use a possibilist universal quantifier in axiom Ax4 and in the definition of essence while keeping the other actualist individual quantifiers as suggested by Anderson [12, p.301] (see also Footnote 20) without affecting the validity of the results presented in Figs. 6 and 7. A small modification needed for the proof step verifications to succeed in Fig. 7 is that the proof of theorem Th2 now requires symmetry of the accessibility relation (or modal axiom B), as an additional dependency (reflexivity would actually also do the job).

4.2 Inconsistency in Gödel's 1970 manuscript

As discussed in prior related work [9, 108], the inconsistency of Gödel's postulates from his 1970 manuscript was detected automatically by the theorem prover Leo-II [79]. This inconsistency, unknown to the authors before the discovery of Leo-II, is reconstructed here in some detail. The inconsistency is due to a controversial omission in Gödel's definition of essence: Gödel does not require in his 1970 manuscript that essential properties φ of x actually apply to x, i.e., hold for x (line 8).[26] Unaware of the thread of inconsistency hidden in this omission, Scott found it natural to add such a clause in his variant of Gödel's ontological argument, which we will present later in Sect. 5.[27]

The inconsistency follows quite easily from what is called in prior work [9, 108] the "empty essence lemma" (line 24 of Fig. 6): $(\lambda x.\bot)\ \text{Ess.}\ x$. This lemma states that the absurd, empty property $\lambda x.\bot$ (or, alternatively, the extensionally equivalent property of self-difference $\lambda x.x \neq x$, see line 51 in Fig. 4), which does not apply to any x, is nevertheless an essential property of every x.[28]

Starting from the additional result that the empty property (or self-difference) cannot be a positive property due to Ax2a and Ax4 (line 27), we can now conclude falsity (lines 25 and 32) by deriving from the axioms Ax3, Ax4 and Ax2a with the help of the empty essence lemma and the definition of necessary existence that the opposite also holds, namely that the empty property must also be positive (line 31). Note that the proof sketch of falsity as displayed does not require any specific properties of modal logic S5. This derivation is instead valid already in base modal logic K.

4.3 Consistent variant: modifying essence

To arrive at a consistent set of assumptions in Gödel's work, only a small addition of the conjunct "$\varphi\, x\ \wedge$" in the definition of essence is required. Hence, as suggested by Scott, instead of defining $\varphi\ \text{Ess.}\ x$ as $\forall\psi.\ \psi\, x \supset (\varphi \supset_N \psi))$, we now define (line 8 of Fig. 7)

$$\varphi\ \text{Ess.}\ x \equiv \varphi\, x \wedge \forall\psi.\ \psi\, x \supset (\varphi \supset_N \psi))$$

Since this addition forces essential properties φ of an entity x to actually hold for x, it obviously blocks the empty essence lemma, since the absurd empty property cannot not be true for any entity. Note that all other axioms and definitions of Gödel's manuscript remain unchanged (except for the modified definition of Ess. in line 8, lines 3–22 are completely identical). This modified set of assumptions is now satisfiable,

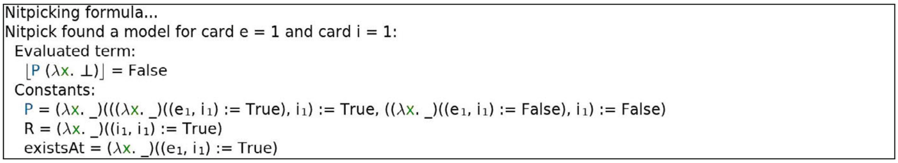
---

[26] Adams [1, p. 392] actually mentions "that if there is a property that is necessarily false of everything, then it must be an essence of x" and he notes that undesirable conclusions can be drawn from this. Although he does not use the word inconsistency directly, he apparently recognized the problem, which was independently discovered by the theorem prover Leo-II, who reported a derivation of falsity from Gödel's premises in experiments.

[27] However, as discussed by Kanckos and Lethen [4], such an additional conjunct in the definition of essence can in fact be found in earlier notes on the ontological argument in Gödel's Nachlass. Whether the omission was actually intended by Gödel remains an interesting open question.

[28] Adams [1] mentions the property that is necessarily false of everything, and which is an essence of x. However, Adams does not point out the inconsistency caused by it.

which is actually confirmed by the model finder Nitpick (line 23), which automatically computes the following model of cardinality one:



```
Nitpicking formula...
Nitpick found a model for card e = 1 and card i = 1:
 Evaluated term:
  ⌊P (λx. ⊥)⌋ = False
 Constants:
  P = (λx. _)(((λx. _)((e₁, i₁) := True), i₁) := True, ((λx. _)((e₁, i₁) := False), i₁) := False)
  R = (λx. _)((i₁, i₁) := True)
  existsAt = (λx. _)((e₁, i₁) := True)
```

This model consists of one self-connected world i_1 with one God-like entity e_1 which exists at world i_1. The modal properties $\lambda x.\top$ and $\lambda x.\bot$ in this world constitute positive and non-positive properties, respectively (as can be seen from the displayed interpretation of P). Note also that the requested evaluation of formula $\lfloor \mathrm{P}\,(\lambda x.\bot)\rfloor$ in the call to Nitpick (in line 23) returns false.

4.4 The possible and necessary existence of a God-like entity

Unfortunately attempts to continue Gödel's proof after fixing the inconsistency as reported above do not yet succeed, and the automated provers fail to prove the crucial theorem Th4 demonstrating the possible existence of a God-like entity (line 28 of Fig. 7), from which, due to theorem Th3, the intended result, the necessary existence of a God-like entity, would immediately follow.

Theorem Th4: $\Diamond(\exists^E x.\, \boldsymbol{G}\, x)$

This theorem, shown above right in Gödel's 1970 manuscript, responds to a prominent criticism of Anselm's ontological argument, which implicitly assumes that the existence of a God-like entity is possible. With theorem Th4, Gödel instead wants to derive this explicitly as a theorem from his assumptions. For this purpose, Gödel's footnote addition to axiom Ax1 (shown in the lower left corner of the manuscript) now actually becomes relevant. It suggests a generalization of axiom Ax1 for any number of summands/conjuncts. A formal modeling, as suggested by Anderson and Gettings [13] and used also by Fitting [17], is possible in the following way: First, introduce a predicate "PosProps" (line 24) that holds for a set of properties Φ if (and only if) Φ contains only positive properties. Second, define a relation "ConjOfPropsFrom" (line 25) that holds for a property φ and a set of properties Φ if (and only if) φ is equivalent to an arbitrary and possibly infinite conjunction of properties from φ. Next, we use these definitions to formalize the following generalization of axiom Ax1:

Axiom Ax1Gen: $(\boldsymbol{PosProps}\ \Phi \wedge \boldsymbol{ConjOfPropsFrom}\ \varphi\ \Phi) \supset \boldsymbol{P}\ \varphi$

This axiom says that every conjunction of positive properties, including all infinite conjunctions, is positive. It turns out that from this axiom Ax1Gen together with the definition of a God-like entity one can now prove lemma L:

Lemma L: P G

This lemma states that being God-like is a positive property (line 27).

```
 1 theory GoedelVariantHOML2 imports HOMLinHOL ModalFilter
 2 begin
 3   consts PositiveProperty::"(e⇒σ)⇒σ" ("P")
 4   axiomatization where Ax1: "⌊P φ ∧ P ψ ⊃ P (φ . ψ)⌋"
 5   axiomatization where Ax2a: "⌊P φ ∨ᵉ P ∼φ⌋"
 6   definition God ("G") where "G x ≡ ∀φ. P φ ⊃ φ x"
 7   abbreviation PropertyInclusion ("_⊃N_") where "φ ⊃N ψ  ≡  □(∀Ey.  φ y ⊃ ψ y)"
 8 (**)  definition Essence ("_Ess._") where "φ Ess. x ≡ φ x ∧ (∀ψ. ψ x ⊃ (φ ⊃N ψ))"
 9   axiomatization where Ax2b: "⌊P φ ⊃ □ P φ⌋"
10   lemma Ax2b': "⌊¬P φ ⊃ □(¬P φ)⌋" using Ax2a Ax2b by blast
11   theorem Th1: "⌊G x ⊃ G Ess. x⌋" using Ax2a Ax2b Essence_def God_def by (smt (verit))
12   definition NecExist ("E") where "E x  ≡ ∀φ. φ Ess. x ⊃ □(∃Ex. φ x)"
13   axiomatization where Ax3: "⌊P E⌋"
14   theorem Th2: "⌊G x ⊃ □(∃Ey. G y)⌋" using Ax3 Th1 God_def NecExist_def by smt
15   theorem Th3: "⌊◇(∃Ex. G x) ⊃ □(∃Ey. G y)⌋" sledgehammer(Th2 Rsymm)  —‹Proof found›
16     proof -
17       have 1: "⌊(∃Ex. G x) ⊃ □(∃Ey. G y)⌋" using Th2 by blast
18       have 2: "⌊◇(∃Ex. G x) ⊃ ◇□(∃Ey. G y)⌋" using 1 by blast
19       have 3: "⌊◇(∃Ex. G x) ⊃ □(∃Ey. G y)⌋" using 2 Rsymm by blast
20       thus ?thesis by blast
21     qed
22   axiomatization where Ax4: "⌊P φ ∧ (φ ⊃N ψ) ⊃ P ψ⌋"
23   lemma True nitpick[satisfy,card=1,eval="⌊P (λx.⊥)⌋"] oops  —‹One model found of cardinality one›
24   abbreviation "PosProps Φ ≡ ∀φ. Φ φ ⊃ P φ"
25   abbreviation "ConjOfPropsFrom φ Φ ≡ □(∀Ez. φ z ↔ (∀ψ. Φ ψ ⊃ ψ z))"
26   axiomatization where Ax1Gen: "⌊(PosProps Φ ∧ ConjOfPropsFrom φ Φ) ⊃ P φ⌋"
27   lemma L:       "⌊P G⌋"              using Ax1Gen God_def by smt
28   theorem Th4: "⌊◇(∃Ex. G x)⌋"  using Ax2a Ax4 L by blast
29   theorem Th5: "⌊□(∃Ex. G x)⌋"  using Th3 Th4 by blast
30   lemma MC:     "⌊φ ⊃ □φ⌋"  sledgehammer(Ax2a Ax2b Th5 God_def Rsymm)  —‹Proof found›
31     proof - {fix w fix Q
32       have 1: "∀x.(G x w ⟶ (∀Z. Z x ⊃ □(∀Ez. G z ⊃ Z z)) w)" using Ax2a Ax2b God_def by smt
33       have 2: "(∃x. G x w)⟶((Q ⊃ □(∀Ez. G z ⊃ Q)) w)" using 1 by force
34       have 3: "(Q ⊃ □Q) w" using 2 Th5 Rsymm by blast}
35       thus ?thesis by auto
36     qed
37   lemma PosProps:     "⌊P (λx.T) ∧  P (λx. x = x)⌋"    using Ax2a Ax4 by blast
38   lemma NegProps:     "⌊¬P(λx.⊥) ∧ ¬P(λx. x ≠ x)⌋"  using Ax2a Ax4 by blast
39   lemma UniqueEss1:  "⌊φ Ess. x ∧ ψ Ess. x ⊃  □(∀Ey. φ y ↔ ψ y)⌋" using Essence_def by smt
40   lemma UniqueEss2:  "⌊φ Ess. x ∧ ψ Ess. x ⊃  □(φ ≡ ψ)⌋" nitpick[card i=1] oops   —‹Countermodel found›
41   lemma UniqueEss3:  "⌊φ Ess. x ⊃  □(∀Ey. φ y ⊃ y ≡ x)⌋" using Essence_def MC by auto
42   lemma Monotheism: "⌊G x ∧ G y ⊃  x ≡ y⌋" using Ax2a God_def by smt
43   lemma Filter:        "⌊Filter P⌋"     using Ax1 Ax4 MC NegProps PosProps Rsymm by smt
44   lemma UltraFilter: "⌊UFilter P⌋"  using Ax2a Filter by smt
45   lemma True nitpick[satisfy,card=1,eval="⌊P (λx.⊥)⌋"] oops  —‹One model found of cardinality one›
46 end
```

Fig. 7 After an appropriate modification of the definition of essence in Gödel’s 1970 ontological proof, the inconsistency revealed in Fig. 6 is avoided, and the argument can be successfully verified in modal logic S5 (indeed, as shown, only symmetry of the accessibility relation is actually needed)

Using lemma L in combination with axioms Ax2a and Ax4, it is now possible to provide a proof of Gödel’s Theorem Th4 as intended; this is e.g. an easy exercise for the trusted tableaux theorem prover blast (line 28). Note that the axiom Ax1Gen (or A1) is only needed to introduce this lemma for the proof of theorem Th4. It is therefore possible to omit Ax1Gen and A1 altogether and simply postulate L as an axiom. This is actually the simplifying approach that Scott used in his variant of Gödel’s ontological proof, where L is then called axiom A3; see Sect. 5.

Now we are in the position to state and prove the main theorem.

Theorem Th5: $\Box(\exists^E x. G\, x)$

The necessary existence of a God-like entity is now provable as a simple corollary of Th3 and Th4 (line 29).

It is easy to see that the entire derivation of the necessary existence of a God-like entity from Gödel's assumptions, as shown in Fig. 7, does not use all the reasoning principles of modal logic S5. Only the symmetry of the accessibility relation between possible worlds is required, which corresponds to the (rather uncontroversial) modal axiom scheme B: $\varphi \supset \Box\Diamond\varphi$, expressing that if φ holds, then it is necessarily possible that φ holds.

There are various other experiments that can now be additionally conducted with the automated theorem provers built into Isabelle/HOL to explore further implications of Gödel's set of postulates. We report some of the relevant results.

Modal Collapse MC: $\varphi \supset \Box\varphi$

Modal collapse, which states that what holds that holds necessarily, is implied by Gödel's axioms and definitions (line 30–36). It is relevant to note that modal collapse also followed for all alternative variants considered in this paper, where actualist individual quantifiers are replaced by possibilist ones. It is reasonable to assume that modal collapse, which eliminates contingency and expresses that there are no other ways the world could be, was in fact not seen by Gödel as a defect of his theory, but as a much desired consequence of his postulates; see also Kovač [109].

Examples of positive properties: $\boldsymbol{P}\,(\lambda x.\top) \wedge \boldsymbol{P}\,(\lambda x.x{=}x)$

The tautological property $\lambda x.\top$ and self-identity $\lambda x.x{=}x$ are examples of positive properties (line 37) in addition to those already mentioned in the proof; see also the middle right part of Gödel's manuscript.

Examples of non-positive properties: $\neg\boldsymbol{P}\,(\lambda x.\bot) \wedge \neg\boldsymbol{P}\,(\lambda x.x{\neq}x)$

The absurd, empty property $\lambda x.\bot$ and self-difference $\lambda x.x{\neq}x$ are examples of non-positive properties (line 38); see also the middle right part of Gödel's manuscript.

Unique essences

Any two essences of x are necessarily equivalent (for all entities y, line 39), but not necessarily identical (line 40). This statement can be found in the left bottom part of Gödel's 1970 manuscript. Another statement is verified (in line 41) expressing that entities are uniquely determined by their essences.

Monotheism: $\boldsymbol{G}\,x \wedge \boldsymbol{G}\,y \supset x \equiv y$

Monotheism follows directly from axiom A2a and definition of God-like (line 42).

Ultrafilter

The set of positive properties forms a (modal) set ultrafilter (lines 43–44). See also Hazen [110] and the previous experimentally confirmed studies by Benzmüller and Fuenmayor [35] for a deeper discussion of this aspect.

We finally perform another consistency check with model finder Nitpick (in line 45 of Fig. 7), which computes the same model of cardinality one as discussed already in Sect. 4.3 above.

Logic S4

A natural question is whether the axiom schemes M, D or 4 of the logic S4, respectively their semantic counterparts reflexivity and transitivity of the accessibility relation, could be used instead of the axiom scheme B or symmetry (called "Rsymm" in our formalization files). This question actually only concerns the proof of Gödel's Theorem Th3, since it is only there that the symmetry axiom is actually needed. While a proof for theorem Th3 based on reflexivity alone is indeed found by the theorem provers in the context of the (inconsistent) variant as shown in Fig. 6, no such proof has yet been reported, nor a counterexample found in logic S4 for the variant shown in Fig. 7, where the definition of essence has been adapted. This question thus remains open.

4.5 Consistent variant: modifying property implication

There is actually another way to modify Gödel's ontological proof to avoid the reported inconsistency. This newly explored modification is shown in Fig. 8. The only relevant modification in this variant concerns the notion of necessary property implication, which is used in the definition of essence and in the axiom Ax4. Recall that necessary property implication $\varphi \supset_N \psi$ has been formalized so far (also in line with Fitting [17]) as

$$\Box(\forall^E y.\, \varphi\, y \supset \psi\, y)$$

An alternative suggestion is to encode $\varphi \supset_N \psi$ as

$$\Box(\varphi \neq (\lambda x.\bot) \land \forall^E y.\, \varphi\, y \supset \psi\, y)$$

With this modification (see line 7 of Fig. 8), which prevents the absurd, empty property from necessarily implying all other properties, the relevant proof steps as reported in Fig. 7 can be successfully replayed and consistency is again shown by Nitpick (line 23).[29]

However, while Nitpick in Fig. 7 reported exactly one model (for cardinality one, i.e. when assuming one world and one entity), Nitpick now reports two models, see Fig. 9.

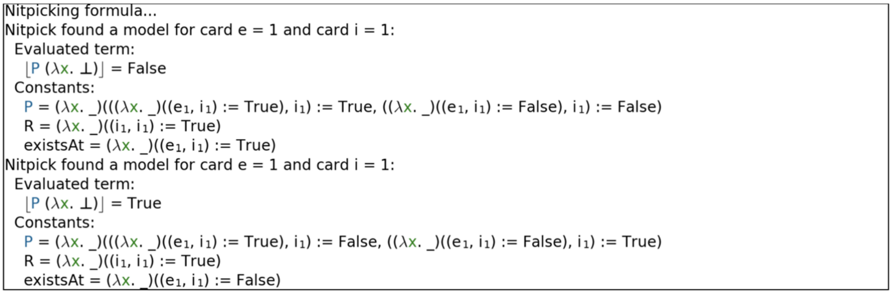
The first of these models is exactly the same as the one previously reported (for line 23 in Fig. 7). In addition, we are now confronted with another interesting model that satisfies the axioms and definitions introduced so far in Fig. 8. This additional model consists of a self-connected world i_1 with an entity e_1 that is "non-existent" (in a free logic sense modulo our explicit existence operator @) in world i_1; most interestingly, this entity e_1 is not a God-like entity in i_1 and $\lambda x.\bot$ is now a positive property in i_1, while a $\lambda x.\top$ is not. Note also that the requested evaluation of the formula $\lfloor \mathrm{P}\,(\lambda x.\bot)\rfloor$ in the call to Nitpick (line 23) now returns True. As we can

---

[29] Additional experiments have shown that we can alternatively use a possibilist universal quantifier in the modified definition of necessary property implication, used in axiom Ax4 and the definition of essence, without affecting the validity of the results shown in Fig. 8; different to the observation mentioned in Footnote 25, no further dependencies need to be added for the proof of Theorem Th4 to succeed.

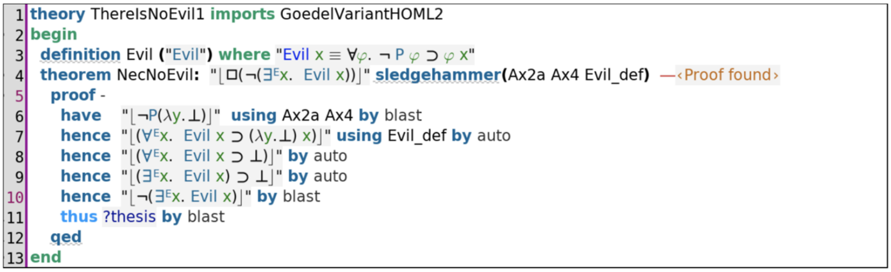
```
 1 theory GoedelVariantHOML3 imports HOMLinHOL ModalFilter
 2 begin
 3   consts PositiveProperty::"(e⇒σ)⇒σ" ("P")
 4   axiomatization where Ax1: "⌊P φ ∧ P ψ ⊃ P (φ . ψ)⌋"
 5   axiomatization where Ax2a: "⌊P φ ∨e P ~φ⌋"
 6   definition God ("G") where "G x ≡ ∀φ. P φ ⊃ φ x"
 7 (**)  abbreviation PropertyInclusion ("_⊃N_") where "φ ⊃N ψ  ≡ □(φ ≠ (λx. ⊥) ∧ (∀Ey.  φ y ⊃ ψ y))"
 8   definition Essence ("_Ess._") where "φ Ess. x ≡ ∀ψ. ψ x ⊃ (φ ⊃N ψ)"
 9   axiomatization where Ax2b: "⌊P φ ⊃ □ P φ⌋"
10   lemma Ax2b': "⌊¬P φ ⊃ □(¬P φ)⌋" using Ax2a Ax2b by blast
11   theorem Th1: "⌊G x ⊃ G Ess. x⌋" using Ax2a Ax2b Essence_def God_def by (smt (verit))
12   definition NecExist ("E") where "E x  ≡ ∀φ. φ Ess. x ⊃ □(∃Ex. φ x)"
13   axiomatization where Ax3: "⌊P E⌋"
14   theorem Th2: "⌊G x ⊃ □(∃Ey. G y)⌋" using Ax3 Th1 God_def NecExist_def by smt
15   theorem Th3: "⌊◇(∃Ex. G x) ⊃ □(∃Ey. G y)⌋" sledgehammer(Th2 Rsymm)  —‹Proof found›
16     proof -
17       have 1: "⌊(∃Ex. G x) ⊃ □(∃Ey. G y)⌋" using Th2 by blast
18       have 2: "⌊◇(∃Ex. G x) ⊃ ◇□(∃Ey. G y)⌋" using 1 by blast
19       have 3: "⌊◇(∃Ex. G x) ⊃ □(∃Ey. G y)⌋" using 2 Rsymm by blast
20       thus ?thesis by blast
21    qed
22   axiomatization where Ax4: "⌊P φ ∧ (φ ⊃N ψ) ⊃ P ψ⌋"
23   lemma True nitpick[satisfy,card=1,eval="⌊P (λx.⊥)⌋"]  oops  —‹Two models found of cardinality one›
24   abbreviation "PosProps Φ ≡ ∀φ. Φ φ ⊃ P φ"
25   abbreviation "ConjOfPropsFrom φ Φ ≡ □(∀Ez. φ z ↔ (∀ψ. Φ ψ ⊃ ψ z))"
26   axiomatization where Ax1Gen: "⌊(PosProps Φ ∧ ConjOfPropsFrom φ Φ) ⊃ P φ⌋"
27   lemma L:       "⌊P G⌋"  using Ax1Gen God_def by (smt (verit))
28   theorem Th4: "⌊◇(∃Ex. G x)⌋" sledgehammer[timeout=200](Ax2a L Ax1Gen)  sorry  —‹Proof found›
29   theorem Th5: "⌊□(∃Ex. G x)⌋" using Th3 Th4 by blast
30   lemma MC:     "⌊φ ⊃ □φ⌋"  sledgehammer(Ax2a Ax2b Th5 God_def Rsymm)  —‹Proof found›
31     proof - {fix w fix Q
32       have 1: "∀x.(G x w ⟶ (∀Z. Z x ⊃ □(∀Ez. G z ⊃ Z z)) w)" using Ax2a Ax2b God_def by smt
33       have 2: "(∃x. G x w)⟶((Q ⊃ □(∀Ez. G z ⊃ Q)) w)" using 1 by force
34       have 3: "(Q ⊃ □Q) w" using 2 Th5 Rsymm by blast}
35       thus ?thesis by auto
36      qed
37   lemma PosProps:      "⌊P (λx.T) ∧  P(λx. x = x)⌋"     using Ax2a Ax4 L Th4 by smt
38   lemma NegProps:      "⌊¬P(λx.⊥) ∧ ¬P(λx. x ≠ x)⌋"  using Ax2a Ax4 L Th4 by smt
39   lemma UniqueEss1:  "⌊φ Ess. x ∧ ψ Ess. x ⊃  □(∀Ey. φ y ↔ ψ y)⌋"  oops  —‹Unclear, open question›
40   lemma UniqueEss2:  "⌊φ Ess. x ∧ ψ Ess. x ⊃  □(φ ≡ ψ)⌋"   oops  —‹Unclear, open question›
41   lemma UniqueEss3:  "⌊φ Ess. x ⊃  □(∀Ey. φ y ⊃ y ≡ x)⌋" using Essence_def MC by auto
42   lemma Monotheism: "⌊G x ∧ G y ⊃  x ≡ y⌋" using Ax2a God_def by smt
43   lemma Filter:        "⌊Filter P⌋"     using Ax1 Ax4 MC NegProps PosProps Rsymm by smt
44   lemma UltraFilter: "⌊UFilter P⌋"  using Ax2a Filter by smt
45   lemma True nitpick[satisfy,card=1,eval="⌊P (λx.T)⌋"]  oops  —‹One model found of cardinality one›
46 end
```

Fig. 8 After an appropriate modification of the notion of necessary property inclusion in Gödel's 1970 ontological proof, the inconsistency revealed in Fig. 6 is avoided, and the argument can be successfully verified in modal logic S5 (indeed, as shown, only symmetry of the accessibility relation is actually needed)



```
Nitpicking formula...
Nitpick found a model for card e = 1 and card i = 1:
  Evaluated term:
    ⌊P (λx. ⊥)⌋ = False
  Constants:
    P = (λx. _)(((λx. _)((e1, i1) := True), i1) := True, ((λx. _)((e1, i1) := False), i1) := False)
    R = (λx. _)((i1, i1) := True)
    existsAt = (λx. _)((e1, i1) := True)
Nitpick found a model for card e = 1 and card i = 1:
  Evaluated term:
    ⌊P (λx. ⊥)⌋ = True
  Constants:
    P = (λx. _)(((λx. _)((e1, i1) := True), i1) := False, ((λx. _)((e1, i1) := False), i1) := True)
    R = (λx. _)((i1, i1) := True)
    existsAt = (λx. _)((e1, i1) := False)
```

Fig. 9 Nitpick reports two models of cardinality one for the postulates up to line 23 in Fig. 8



```
 1 theory ThereIsNoEvil1 imports GoedelVariantHOML2
 2 begin
 3   definition Evil ("Evil") where "Evil x ≡ ∀φ. ¬ P φ ⊃ φ x"
 4   theorem NecNoEvil: "⌊□(¬(∃ᴱx.  Evil x))⌋" sledgehammer(Ax2a Ax4 Evil_def) —‹Proof found›
 5     proof -
 6       have   "⌊¬P(λy.⊥)⌋"  using Ax2a Ax4 by blast
 7       hence "⌊(∀ᴱx.  Evil x ⊃ (λy.⊥) x)⌋" using Evil_def by auto
 8       hence "⌊(∀ᴱx.  Evil x ⊃ ⊥)⌋" by auto
 9       hence "⌊(∃ᴱx.  Evil x) ⊃ ⊥⌋" by auto
10       hence "⌊¬(∃ᴱx. Evil x)⌋" by blast
11       thus ?thesis by blast
12     qed
13 end
```

Fig. 10 Importing Gödel's modified axioms from Fig. 7 we can prove that necessarily there exists no entity that possesses all non-positive (=negative) properties

see (line 45), after postulating the generalized axiom A1Gen (line 25), however, this second model without a God-like entity is ruled out.

Another relevant difference concerns the proof of theorem Th4 (line 28), the possible existence of a God-like entity. This proof becomes now more difficult. Instead of proving this result from Ax2a and Ax4 and lemma L, a proof using Ax2a, L and A1Gen is now reported (line 28). Independently the automated theorem provers Vampire and Zipperposition (and also Leo-II and Leo-III), integrated with Isabelle/HOL via the Sledgehammer tool, report proofs for Th4 from these dependencies.[30]

The subsequent proofs of theorem Th5, modal collapse MC and the lemmata PosProps and NegProps are again analogous to the previous ones. The situation for the unique essence lemmata one and two (lines 39–40), however, remains open; neither did the model finder report a counterexample nor did the automated theorem provers come up with a proof in both cases (though we expect them to be provable). The set of positive properties constitutes a (modal) set ultrafilter as before. Moreover, as in the previous subsection, the question about a proof for theorem Th3 in logic S4 remains an open question.

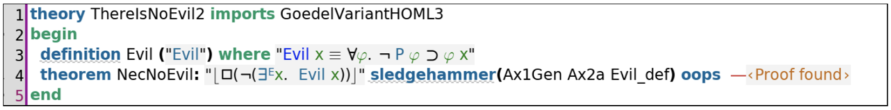
4.6 Necessarily there is no Evil—if we accept Gödel's axioms

It might be tempting to try to prove the existence of an Evil-like entity in an analogous way, by simply defining an entity as Evil-like if (and only if) it possesses all non-positive (=negative) properties. However, if one accepts only two axioms of Gödel from Fig. 7, namely axiom Ax2a (resp. Scott's axiom A1) and axiom Ax4 (resp. Scott's axiom A2) about positive properties, then the derivation shown in Fig. 10 is possible (lines 4–12), which demonstrates that necessarily there is no Evil-like entity under these assumptions.[31]

---

[30] Unfortunately, however, their proofs could not yet be reconstructed by weaker but trusted provers integrated into Isabelle/HOL; this explains the use of the keyword "sorry" highlighted in red. Hence, some further work is needed here to fully verify these novel results also within the trusted kernel of the Isabelle/HOL system.

[31] The idea was actually suggested to Benzmüller by Chad Brown in a bar conversation in Berlin.



```
theory ThereIsNoEvil2 imports GoedelVariantHOML3
begin
  definition Evil ("Evil") where "Evil x ≡ ∀φ. ¬ P φ ⊃ φ x"
  theorem NecNoEvil: "⌊□(¬(∃Ex.  Evil x))⌋" sledgehammer(Ax1Gen Ax2a Evil_def) oops —‹Proof found›
end
```

Fig. 11 Importing Gödel's modified axioms from Fig. 8 we can prove that necessarily there exists no entity that possesses all non-positive (=negative) properties

An analogous derivation is possible when importing the corresponding axioms from Fig. 8, see Fig. 11. However, since in this variant axiom Ax4 has been weakened, the axioms A1Gen and Ax2a are now used by the provers to obtain the result.

However, the necessary existence of an Evil-like entity is derivable if we reject Gödel's assumptions and formulate them instead completely for negative properties. This is shown in Fig. 20 in Appendix E. The properties of this Evil-like entity are however identical to those of Gödel's God-like entity.

5 Scott's variant

A formalization of Scott's variant of Gödel's ontological argument is presented in Fig. 12. The presentation is close to the notes of Scott as published in Sobel's book [3, 6]. We briefly discuss the commonalities and differences to Gödel's variant.

Positive properties: P
The notion of positive properties is the same as in Sect. 4.

Axiom A1: $\neg \boldsymbol{P}\,\varphi \leftrightarrow \boldsymbol{P} \sim\!\varphi$
Properties are either positive or non-positive. Axiom A1 correspond to Gödel's axiom Ax2a: $\mathrm{P}\varphi \vee^e \mathrm{P}\!\sim\!\varphi$ from Sect. 4. However, here the equivalent formulation $\neg\mathrm{P}\,\varphi \leftrightarrow \mathrm{P} \sim\!\varphi$ is preferred.

Axiom A2: $\boldsymbol{P}\,\varphi \wedge \Box(\forall^E y.\, \varphi\, y \supset \psi\, y) \supset \boldsymbol{P}\,\psi$
Axiom A2 is identical to Gödel's axiom Ax4 from Sect. 4 (modulo the unfolding of $\varphi \supset_N \psi$)

Theorem T1: $\boldsymbol{P}\,\varphi \supset \Diamond(\exists^E x.\, \varphi\, x)$
Positive properties are possibly exemplified. This theorem follows directly from axioms A1 and A2.

While a corresponding theorem is not directly included in Gödel's ontological proof in Sect. 4, this theorem is also valid for the variants presented in Sects. 4.4 and 4.5 (as confirmed by tests not shown here).

Definition of God-like: $\boldsymbol{G}\,x \equiv \forall\varphi.\, \boldsymbol{P}\,\varphi \supset \varphi\, x$
A God-like entity possesses all positive properties. This is identical to Gödel's definition from Sect. 4.

```
theory ScottVariantHOML imports HOMLinHOL ModalFilter
begin
  consts PositiveProperty::"(e⇒σ)⇒σ" ("P")
  axiomatization where A1:           "⌊¬P φ ↔ P ~φ⌋"
  axiomatization where A2:           "⌊P φ ∧ □(∀ᴱy.  φ y ⊃ ψ y) ⊃ P ψ⌋"
  theorem T1:                        "⌊P φ ⊃ ◇(∃ᴱx. φ x)⌋"  using A1 A2  by blast
  definition God ("G") where         "G x ≡ ∀φ. P φ ⊃ φ x"
  axiomatization where A3:           "⌊P G⌋"
  theorem Coro:                      "⌊◇(∃ᴱx. G x)⌋"  using A3 T1 by blast
  axiomatization where A4:           "⌊P φ ⊃ □ P φ⌋"
  definition Ess ("_Ess._") where    "φ Ess. x ≡  φ x ∧ (∀ψ. ψ x ⊃ □(∀ᴱy.  φ y ⊃ ψ y))"
  theorem T2:                        "⌊G x ⊃ G Ess. x⌋"  using A1 A4 Ess_def God_def by fastforce
  definition NecExist ("NE") where   "NE x  ≡ ∀φ. φ Ess. x ⊃ □(∃ᴱx. φ x)"
  axiomatization where A5:           "⌊P NE⌋"
  lemma True nitpick[satisfy,card=1,eval="⌊P (λx. T)⌋"]  oops  —‹One model found of cardinality one›
  theorem  T3:  "⌊□(∃ᴱx. G x)⌋" sledgehammer(A5 Coro God_def NecExist_def Rsymm T2) —‹Proof found›
    proof -
      have   1: "⌊(G x ⊃ NE x) ∧ (G Ess. x ⊃ □(∃ᴱx.  G x))⌋" using A5 Ess_def God_def NecExist_def by smt
      hence 2: "⌊(∃ᴱx. G x) ⊃ □(∃ᴱx. G x)⌋"  using A5 God_def NecExist_def T2 by smt
      hence 3: "⌊◇(∃ᴱx. G x) ⊃ (◇(□(∃ᴱx. G x)) ⊃ □(∃ᴱx. G x))⌋"  using Rsymm by blast
      thus ?thesis using 2 Coro by blast
    qed
  lemma MC: "⌊φ ⊃ □φ⌋"   sledgehammer(A1 A4 God_def Rsymm T3)  —‹Proof found›
    proof - {fix w fix Q
       have 1: "∀x.(G x w  ⟶  (∀Z. Z x ⊃ □(∀ᴱz.((G z) ⊃ (Z z)))) w)" using A1 A4 God_def by smt
       have 2: "(∃x. G x w)⟶((Q ⊃ □(∀ᴱz.((G z) ⊃ Q))) w)" using 1 by force
       have 3: "(Q ⊃ □Q) w" using 2 T3 Rsymm by blast}
       thus ?thesis by auto
     qed
  lemma PosProps:      "⌊P (λx. T) ∧  P (λx. x = x)⌋"    using A1 A2 by blast
  lemma NegProps:      "⌊¬P(λx. ⊥) ∧ ¬P(λx. x ≠ x)⌋"   using A1 A2 by blast
  lemma UniqueEss1:  "⌊φ Ess. x ∧ ψ Ess. x ⊃  □(∀ᴱy. φ y ↔ ψ y)⌋" using Ess_def  by smt
  lemma UniqueEss2:  "⌊φ Ess. x ∧ ψ Ess. x ⊃  □(φ ≡ ψ)⌋" nitpick[card i=1] oops   —‹Countermodel found›
  lemma UniqueEss3:  "⌊φ Ess. x ⊃  □(∀ᴱy. φ y ⊃ y ≡ x)⌋" using Ess_def MC by auto
  lemma Monotheism: "⌊G x ∧ G y ⊃  x ≡ y⌋" using A1 God_def by smt
  lemma Filter:        "⌊Filter P⌋"     using A1 God_def Rsymm T1 T3 by (smt (verit, best))
  lemma UltraFilter: "⌊UFilter P⌋"  using Filter A1 by blast
  lemma True nitpick[satisfy,card=1,eval="⌊P (λx. ⊥)⌋"] oops  —‹One model found of cardinality one›
end
```

Fig. 12 Scott's variant of Gödel's ontological argument

Axiom A3: P G

Being God-like is a positive property. This axiom replaces Gödel's axiom Ax1, respectively its generalized version Ax1Gen, which was needed for the proofs to succeed in Sects. 4.4 and 4.5. Since Ax1Gen was used there only to derive the lemma L: P G, this lemma can be taken directly as an axiom. However, it is also possible to use the axiom Ax1Gen instead and derive P G from it.

Theorem Coro: $\Diamond(\exists^E x . \boldsymbol{G}\, x)$

It is possible for a God-like entity to exist. This follows directly from theorem T1 and axiom A3.

Axiom A4: $\boldsymbol{P}\,\varphi \supset \Box \boldsymbol{P}\,\varphi$

Positive properties are necessarily positive. This axiom is identical to Gödel's axiom Ax2b from Sect. 4.

Definition of essence: $\varphi\, \boldsymbol{Ess.}\, x \equiv \varphi\, x \wedge \forall \psi .\, \psi\, x \supset \Box(\forall^E y . \varphi\, y \supset \psi\, y)$

φ is an essential property of entity x if (and only if) if x exemplifies φ and for all (possibly other) properties ψ that x also has holds that ψ is necessarily implied by φ.

Unaware of the inconsistency reported in Sect. 4.2, Scott found it natural to require that essences of an entity are actually exemplified by the entity. Thus, he added the conjunct $\psi\, x \wedge$ to the definition of essence in his variant, thereby unknowingly avoiding the reported inconsistency.

Theorem T2: $\boldsymbol{G}\, x \supset \boldsymbol{G}\, \boldsymbol{Ess.}\, x$

Being God-like is an essence of any God-like entity. This is identical to Gödel's theorem Th2 from Sect. 4. It follows here from axioms A1 and A4 using the definitions of Essence and God-like.

Definition of Necessary Existence: $\boldsymbol{NE}\, x \equiv \forall \varphi .\, \varphi\, \boldsymbol{Ess.}\, x \supset \Box(\exists^E x .\, \varphi\, x)$

Entity x has necessary existence (NE) if (and only if) all essences of x are necessarily exemplified. This definition is again identical to Gödel's definition from Sect. 4, except that instead of using E as name, the concept is now called NE.

Axiom A5: P NE

Necessary Existence is a positive property. Modulo the renaming of E into NE this axiom is identical to Gödel's axiom Ax3 from Sect. 4.

Consistency of the assumptions

After all assumptions have been introduced we use the model finder Nitpick to confirm their consistency (line 15). Nitpick computes exactly the same model as presented before in Sect. 4.3.

Theorem T3: $\Box(\exists^E x .\, \boldsymbol{G}\, x)$

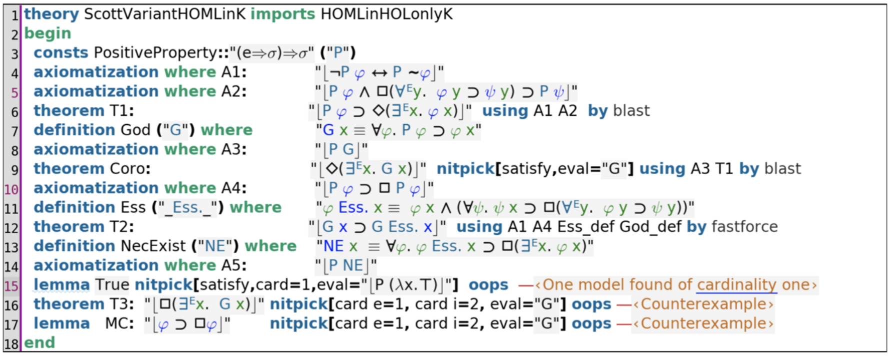
In correspondence to the main theorem Th5 in Sects. 4.4 and 4.5 we are now in the position to prove the main theorem T3: Necessarily, there exists a God-like entity. This can be done fully automatically using automated theorem provers integrated with Isabelle/HOL via the Sledgehammer tool (line 16), or more interactively following the notes of the second author (lines 17–22). By the use of an actualist individual existential quantifier the controversial issue of existential import is avoided.

Further Results

Modal Collapse MC: $\varphi \supset \Box\varphi$ holds also for this variant (lines 23–29), and $\lambda x.\top$ and $\lambda x.x{=}x$ are examples of positive properties (besides NE and G), while $\lambda x.\bot$ and $\lambda x.x{\neq}x$ are negative properties. Also the results for uniqueness of essences (lines 32–34), monotheism(line 35), the filter and ultrafilter structure of P (lines 36–37) and consistency (lines 38) are confirmed analogously to what we have reported in Sect. 4.4 before for the corresponding variant of Gödel.

5.1 Counterexamples in base logic K

A natural question concerns the possible validity of the argument already for basic modal logic K. This is not the case, however, as can be verified with the model finder Nitpick. Nitpick produces counterexamples(s) to theorem T3: $\Box(\exists^E x .\, \mathrm{G}\, x)$ and also to modal collapse MC: $\varphi \supset \Box\varphi$, while verifying all the previous steps, as shown in



```
 1 theory ScottVariantHOMLinK imports HOMLinHOLonlyK
 2 begin
 3   consts PositiveProperty::"(e⇒σ)⇒σ" ("P")
 4   axiomatization where A1:              "⌊¬P φ ↔ P ~φ⌋"
 5   axiomatization where A2:              "⌊P φ ∧ □(∀ᴱy.  φ y ⊃ ψ y) ⊃ P ψ⌋"
 6   theorem T1:                          "⌊P φ ⊃ ◇(∃ᴱx. φ x)⌋"  using A1 A2  by blast
 7   definition God ("G") where            "G x ≡ ∀φ. P φ ⊃ φ x"
 8   axiomatization where A3:              "⌊P G⌋"
 9   theorem Coro:                         "⌊◇(∃ᴱx. G x)⌋"  nitpick[satisfy,eval="G"] using A3 T1 by blast
10   axiomatization where A4:              "⌊P φ ⊃ □ P φ⌋"
11   definition Ess ("_Ess._") where       "φ Ess. x ≡  φ x ∧ (∀ψ. ψ x ⊃ □(∀ᴱy.  φ y ⊃ ψ y))"
12   theorem T2:                          "⌊G x ⊃ G Ess. x⌋"  using A1 A4 Ess_def God_def by fastforce
13   definition NecExist ("NE") where      "NE x  ≡ ∀φ. φ Ess. x ⊃ □(∃ᴱx. φ x)"
14   axiomatization where A5:              "⌊P NE⌋"
15   lemma True nitpick[satisfy,card=1,eval="⌊P (λx.T)⌋"]  oops  —‹One model found of cardinality one›
16   theorem T3:  "⌊□(∃ᴱx.  G x)⌋" nitpick[card e=1, card i=2, eval="G"] oops —‹Counterexample›
17   lemma   MC:  "⌊φ ⊃ □φ⌋"       nitpick[card e=1, card i=2, eval="G"] oops —‹Counterexample›
18 end
```

Fig. 13 Scott’s variant of Gödel’s argument fails for base logic K (but only it the last step)

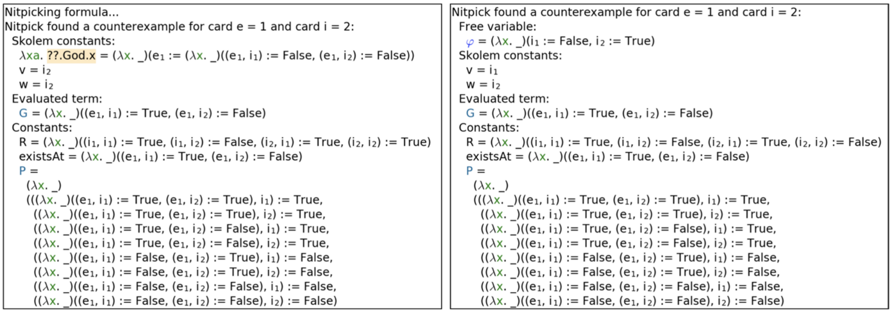
Fig. 13. Note the change in the imported file (see line 1); this file is now “HOMLinHOLonlyK” instead of “HOMLinHOL” as used in Fig. 12 and shown in Fig. 3. The only difference in “HOMLinHOLonlyK” (which is not shown here) in comparison to “HOMLinHOL” is that the postulation of the (semantic counterparts of the) S5 axioms (in lines 12–13) has been removed, resp. commented. This also has the effect that the respective schematic S5 principles displayed in Fig. 4 (in lines 4–8) are not valid anymore.

Counterexamples as produced by Nitpick for theorem T3 and modal collapse MC are shown in Fig. 14: In both counterexamples there are two possible worlds i_1 and i_2, and an entity e_1 that actually exists only in i_1 but not in i_2. Entity e_1 is God-like in world i_1 but not i_2. Both worlds are self-connected, and additionally world i_1 is reachable from world i_2, but not vice versa. The actual world is i_2. The universal property is positive, while the absurd, empty property is not. It is also easy to see that the property of existing in a world is positive. This situation obviously creates a counterexample to theorem T3, since in the actual world i_2, which is reachable from itself, the entity e_1 is not God-like. The counterexample presented on the right to modal collapse MC now suggests to consider a proposition φ, which holds in the actual world i_2, but not in the other possible world i_1 accessible from i_2 (and the non-existence of a God-like entity in a given world would be one possible choice).

Similar observations apply to Gödel’s variants of the argument as presented in Sect. 4.

5.2 Mixed use of actualist and possibilist individual quantifiers

Scott’s variant of Gödel’s argument also works for the mixed use of quantifiers as suggested by Anderson [12, p.301] (see also Footnote 20). The formal modifications required to accommodate Anderson’s view (that actualist individual quantifiers should be used only in the definition of essence and in the final theorem establishing the necessary existence of a God-like entity) are marked with “(**)” in red in Fig. 15. As reported for the analogous experiments (see Footnotes 25 and 29) conducted for the



```
Nitpicking formula...
Nitpick found a counterexample for card e = 1 and card i = 2:
 Skolem constants:
  λxa. ??.God.x = (λx. _)(e₁ := (λx. _)((e₁, i₁) := False, (e₁, i₂) := False))
  v = i₂
  w = i₂
 Evaluated term:
  G = (λx. _)((e₁, i₁) := True, (e₁, i₂) := False)
 Constants:
  R = (λx. _)((i₁, i₁) := True, (i₁, i₂) := False, (i₂, i₁) := True, (i₂, i₂) := True)
  existsAt = (λx. _)((e₁, i₁) := True, (e₁, i₂) := False)
  P =
   (λx. _)
   (((λx. _)((e₁, i₁) := True, (e₁, i₂) := True), i₁) := True,
    ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True), i₂) := True,
    ((λx. _)((e₁, i₁) := True, (e₁, i₂) := False), i₁) := True,
    ((λx. _)((e₁, i₁) := True, (e₁, i₂) := False), i₂) := True,
    ((λx. _)((e₁, i₁) := False, (e₁, i₂) := True), i₁) := False,
    ((λx. _)((e₁, i₁) := False, (e₁, i₂) := True), i₂) := False,
    ((λx. _)((e₁, i₁) := False, (e₁, i₂) := False), i₁) := False,
    ((λx. _)((e₁, i₁) := False, (e₁, i₂) := False), i₂) := False)
```

```
Nitpick found a counterexample for card e = 1 and card i = 2:
 Free variable:
  φ = (λx. _)(i₁ := False, i₂ := True)
 Skolem constants:
  v = i₁
  w = i₂
 Evaluated term:
  G = (λx. _)((e₁, i₁) := True, (e₁, i₂) := False)
 Constants:
  R = (λx. _)((i₁, i₁) := True, (i₁, i₂) := False, (i₂, i₁) := True, (i₂, i₂) := False)
  existsAt = (λx. _)((e₁, i₁) := True, (e₁, i₂) := False)
  P =
   (λx. _)
   (((λx. _)((e₁, i₁) := True, (e₁, i₂) := True), i₁) := True,
    ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True), i₂) := True,
    ((λx. _)((e₁, i₁) := True, (e₁, i₂) := False), i₁) := True,
    ((λx. _)((e₁, i₁) := True, (e₁, i₂) := False), i₂) := True,
    ((λx. _)((e₁, i₁) := False, (e₁, i₂) := True), i₁) := False,
    ((λx. _)((e₁, i₁) := False, (e₁, i₂) := True), i₂) := False,
    ((λx. _)((e₁, i₁) := False, (e₁, i₂) := False), i₁) := False,
    ((λx. _)((e₁, i₁) := False, (e₁, i₂) := False), i₂) := False)
```

Fig. 14 Counterexamples reported by Nitpick to T3: $\Box(\exists^E x.\, \mathrm{G}\, x)$ and MC: $\varphi \supset \Box\varphi$ in base logic K

```
 1 theory ScottVariantHOMLAndersonQuant imports HOMLinHOL ModalFilter
 2 begin
 3   consts PositiveProperty::"(e⇒σ)⇒σ" ("P")
 4   axiomatization where A1:          "⌊¬P φ ↔ P ∼φ⌋"
 5 (**) axiomatization where A2:       "⌊P φ ∧ □(∀y. φ y ⊃ ψ y) ⊃ P ψ⌋"
 6 (**) theorem T1:                    "⌊P φ ⊃ ◇(∃x. φ x)⌋"  using A1 A2 by smt
 7   definition God ("G") where        "G x ≡ ∀φ. P φ ⊃ φ x"
 8   axiomatization where A3:          "⌊P G⌋"
 9 (**) theorem Coro:                   "⌊◇(∃x. G x)⌋"   using A3 T1 by blast
10   axiomatization where A4:          "⌊P φ ⊃ □ P φ⌋"
11 (**) definition Ess ("_Ess._") where  "φ Ess. x ≡  φ x ∧ (∀ψ. ψ x ⊃ □(∀y::e. φ y ⊃ ψ y))"
12   theorem T2:                       "⌊G x ⊃ G Ess. x⌋"  using A1 A4 Ess_def God_def by smt
13   definition NecExist ("NE") where  "NE x  ≡ ∀φ. φ Ess. x ⊃ □(∃ᴱx. φ x)"
14   axiomatization where A5:          "⌊P NE⌋"
15   lemma True nitpick[satisfy,card=1,eval="⌊P (λx.T)⌋"]  oops  —‹One model found of cardinality one›
16   theorem  T3:  "⌊□(∃ᴱx.  G x)⌋" sledgehammer(A5 Coro God_def NecExist_def Rsymm T2) —‹Proof found›
17    proof -
18     have   1: "⌊(G x ⊃ NE x) ∧ (G Ess. x ⊃ □(∃ᴱx.  G x))⌋" using A5 Ess_def God_def NecExist_def by smt
19 (**) hence 2: "⌊(∃x. G x) ⊃ □(∃ᴱx. G x)⌋"  using A5 God_def NecExist_def T2 by smt
20 (**) hence 3: "⌊◇(∃x.  G x) ⊃ (◇(□(∃x. G x)) ⊃ □(∃ᴱx. G x))⌋"  using Rsymm by blast
21     thus ?thesis using 2 Coro by blast
22    qed
23   lemma MC: "⌊φ ⊃ □φ⌋"   sledgehammer(A1 A4 God_def Rsymm T3)  —‹Proof found›
24    proof - {fix w fix Q
25 (**)  have 1: "∀x.(G x w ⟶ (∀Z. Z x ⊃ □(∀z.((G z) ⊃ (Z z)))) w)" using A1 A4 God_def by smt
26 (**)  have 2: "(∃x. G x w)⟶((Q ⊃ □(∀z.((G z) ⊃ Q))) w)" using 1 by force
27      have 3: "(Q ⊃ □Q) w" using 2 T3 Rsymm by blast}
28      thus ?thesis by auto
29     qed
30   lemma PosProps: "⌊P (λx.T) ∧  P (λx. x = x)⌋"       using A1 A2 by blast
31   lemma NegProps:  "⌊¬P(λx.⊥) ∧ ¬P(λx. x ≠ x)⌋"   using A1 A2 by blast
32 (**) lemma UniqueEss1: "⌊φ Ess. x ∧ ψ Ess. x ⊃  □(∀y. φ y ↔ ψ y)⌋" using Ess_def  by smt
33 (**) lemma UniqueEss2: "⌊φ Ess. x ∧ ψ Ess. x ⊃  □(φ = ψ)⌋" nitpick[card i=2] oops  —‹Countermodel found›
34 (**) lemma UniqueEss3:  "⌊φ Ess. x ⊃  □(∀y. φ y ⊃ y ≡ x)⌋" using Ess_def MC by auto
35   lemma Monotheism: "⌊G x ∧ G y ⊃  x ≡ y⌋" using A1 God_def by smt
36   lemma Filter:        "⌊Filter P⌋"     using A1 God_def Rsymm T1 T3 by (smt (verit, best))
37   lemma UltraFilter: "⌊UFilter P⌋"  using Filter A1 by blast
38   lemma True nitpick[satisfy,card=1,eval="⌊P (λx.⊥)⌋"] oops  —‹One model found of cardinality one›
39 end
```

Fig. 15 Scott's variant of Gödel's argument with a mixed use of actualist and possibilist single quantifiers. Modified lines compared to Fig. 12 are marked with "(**)" in red

variants presented earlier, the verification of the argument still succeeds modulo these modifications, and only small, rather obvious changes are required in the interactive proof of theorem T3 (in lines 19–20) and MC (lines 25–26).

5.3 Further work: logic S4 and modified property inclusion

In some further experiments, Nitpick has reported counterexamples for theorem T3 in Figs. 12, 15 and 19 when logic S4 is used. However, these counterexamples could not yet be reproduced if Scott's axiom A3 is replaced by his generalized axiom Ax1Gen from Sect. 4, but also no proof could be found. This inconclusive situation thus requires further study, which was not possible in the time frame of this paper. These studies should also pay attention to the proposed modified notion of property inclusion from Sect. 4.5 as an alternative to the modification of the notion of essence.

6 Conclusion

Several variants of Gödel's ontological argument, including the one proposed by Scott, originally formulated with pen and paper in modal higher-order logic, were studied in detail with a modern proof assistant for classical higher-order logic. The bridge between these two logic contexts and the multiple formalization variants considered (e.g. with respect to different notions of quantification) was made possible by the use of a flexible, logico-pluralistic knowledge representation and reasoning methodology, which distinguishes between a metalogic (classical higher-order logic) and various object logics (variants of higher-order modal logics) encoded in it.[32]

Our systematic experiments have reconfirmed some known results from previous related work, but they have also revealed some new aspects that are included here for the first time. Moreover, this work presents the first formalization work that actually stays as close as possible to what Gödel originally presented in his 1970 manuscript. Gödel, as stated in his "philosophical viewpoints", was apparently convinced that *"There is a scientific (exact) philosophy and theology, which deals with concepts of the highest abstractness; and this is also most highly fruitful for science."* [2, p.316, 9.4.17].[33] In this light, our contribution can be seen as a consistent next step, which Gödel eventually already envisioned: from informal metaphysical studies and arguments, to formalization of the concepts in appropriate logics, to systematic experimentation, and finally to verification (or refutation) on the computer.

Gödel's philosophical notebooks (see, e.g., [114]), while still being under investigation in various research projects, provide increasing evidence that Gödel was searching for and developing, mostly in private, a holistic and comprehensive view of the foun-

---

[32] It is also worth noting that the techniques presented in this article have (i) been very fruitfully used and tested for about a decade by the first author and colleagues in the classroom to support a hands-on university-level logics education, and (ii) they should be adaptable to eventually support, together with students, the formalization and evaluation of recent contributions in the area, e.g., such as [111] or [112].

[33] Note the corrections made to Wang's transcription [2, p.316, 9.4.17] of Gödel's list of 14 philosophical viewpoints as presented by Crocco and Engelen [113]. Unfortunately, there are also errors in Crocco and Engelen's transcription, as one anonymous reviewer pointed out: *Item 9 has to be "Das formal Rechte ist eine Wirklichkeitswissenschaft". This relates to Gödel's lecture "The modern development of the foundations of math. in the light of phil.", where he mentions a linear order of philosophical viewpoints from left to right, theology standing at the far right. This is an important point for the ontological proof as it is an outstanding example of das formal Rechte (i.e. formal theology, formal metaphysics), which Gödel regards as eine "Wirklichkeitswissenschaft".*

dations of mathematics, metaphysics and theology, and so it is not surprising that the notion of God that he tries to capture with the axioms and definitions presented in his 1970 manuscript on the ontological proof is a maximally abstract and maximally consistent, respectively rational, entity, whose set of properties bears a connection to the mathematical structure of an ultrafilter. In this light, it is also not a big surprise that modal collapse, as (presumably) intended by Gödel, follows from his postulates.[34] The mathematical structure that Gödel models here as his notion of positive properties that characterize a God-like entity is therefore a very abstract counterpart to what most religions would accept as their usually overburdened notion of God. Recall that for Gödel *"Religions are, for the most part, bad – but religion is not."* [ibid].

However, whether Gödel's postulates, as presented in his 1970 paper, are indeed convincing and suitable to adequately model this abstract notion of God he was aiming for is still controversially debated, and represents an interesting and challenging opportunity for further research, eventually also supported by further experimentation with modern mathematical proof-assistant systems, as demonstrated in this article … since for Gödel *"There are systematic methods for the solution of all problems (also art etc.)"* [ibid].

Supplementary information

The Isabelle/HOL source files as shown or mentioned in this article are attached as a supplement. In addition these sources are available for download in the Archive of Formal Proofs (AFP) at https://www.isa-afp.org/entries/Notes_On_Goedels_Ontological_Argument.html [44] (the AFP version of the sources is slightly modified to comply with AFP requirements).

Appendix A: Inconsistency of Gödel's 1970 ontological proof: using possibilist quantifiers only

Using only possibilist quantifiers in the formalization of his 1970 manuscript does not prevent the inconsistency reported in Sect. 4.2; see Fig. 16.

Appendix B: Verification of Gödel's 1970 ontological proof: modified notion of essence and using possibilist quantifiers only

The results reported in Sect. 4.4 are still valid when using possibilist quantifiers only; see Fig. 17. The only difference is that in lemma UniqEss2 we now find counterexamples only when we consider two possible worlds (or more). Previously, in Fig. 7, a counterexample was reported already for one possible world.

---

[34] It must be acknowledged that some philosophers take a different view here, e.g., blaming full comprehension as a possible cause of modal collapse; see also Footnote 21.

```
 1 theory GoedelVariantHOML1poss imports HOMLinHOL
 2 begin
 3   consts PositiveProperty::"(e⇒σ)⇒σ" ("P")
 4   axiomatization where Ax1: "⌊P φ ∧ P ψ ⊃ P (φ . ψ)⌋"
 5   axiomatization where Ax2a: "⌊P φ ∨ᵉ P ~φ⌋"
 6   definition God ("G") where "G x ≡ ∀φ. P φ ⊃ φ x"
 7 (**) abbreviation PropertyInclusion ("_⊃N_") where "φ ⊃N ψ  ≡  □(∀y::e.  φ y ⊃ ψ y)"
 8   definition Essence ("_Ess._") where "φ Ess. x ≡ ∀ψ. ψ x ⊃ (φ ⊃N ψ)"
 9   axiomatization where Ax2b: "⌊P φ ⊃ □ P φ⌋"
10   lemma Ax2b': "⌊¬P φ ⊃ □(¬P φ)⌋" using Ax2a Ax2b by blast
11   theorem Th1: "⌊G x ⊃ G Ess. x⌋" using Ax2a Ax2b Essence_def God_def by (smt (verit))
12 (**)  definition NecExist ("E") where "E x  ≡ ∀φ. φ Ess. x ⊃ □(∃x. φ x)"
13   axiomatization where Ax3: "⌊P E⌋"
14 (**)  theorem Th2: "⌊G x ⊃ □(∃y. G y)⌋" using Ax3 Th1 God_def NecExist_def by smt
15 (**)  theorem Th3: "⌊◇(∃x. G x) ⊃ □(∃y. G y)⌋" sledgehammer(Th2 Rsymm)  —‹Proof found›
16         proof -
17 (**)      have 1: "⌊(∃x. G x) ⊃ □(∃y. G y)⌋" using Th2 by blast
18 (**)      have 2: "⌊◇(∃x. G x) ⊃ ◇□(∃y. G y)⌋" using 1 by blast
19 (**)      have 3: "⌊◇(∃x. G x) ⊃ □(∃y. G y)⌋" using 2 Rsymm by blast
20           thus ?thesis by blast
21         qed
22   axiomatization where Ax4: "⌊P φ ∧ (φ ⊃N ψ) ⊃ P ψ⌋"
23   lemma True nitpick[satisfy] oops  —‹No model found›
24   lemma EmptyEssL: "⌊(λy.⊥) Ess. x⌋" using Essence_def by metis
25   theorem Inconsistency: False sledgehammer(Ax2a Ax3 Ax4 EmptyEssL NecExist_def) —‹Proof found›
26    proof -
27     have 1: "⌊¬(P (λx.⊥))⌋" using Ax2a Ax4 by blast
28 (**)   have 2: "⌊P (λx.(λy.⊥) Ess. x ⊃ □(∃z::e.(λy.⊥)z))⌋" using Ax3 Ax4 NecExist_def by smt
29 (**)   have 3: "⌊P (λx.□(∃z. (λx.⊥) z))⌋" using 2 EmptyEssL Ax4 by smt
30     have 4: "⌊P (λx.□⊥)⌋" using 3 by auto
31     have 5: "⌊P (λx.⊥)⌋" using 4 Ax2a Ax4 by smt
32     have 6: "⌊⊥⌋" using 1 5 by blast
33     thus ?thesis by blast
34    qed
35 end
```

Fig. 16 Gödel's axioms and definitions, as presented in the 1970 manuscript, are inconsistent. In contrast to Fig. 6 we here use only possibilist quantifiers and still derive falsity. Modified lines compared to Fig. 6 are marked with "(**)" in red

Appendix C: Verification of Gödel's 1970 ontological proof: modified notion of necessary property implication and using possibilist quantifiers only

The results reported in Sect. 4.5 are still valid when using only possibilist quantifiers; see Fig. 18. The only difference (in line 23) is that now only one model of cardinality one is found for Gödel's postulates.

```
theory GoedelVariantHOML2poss imports HOMLinHOL ModalFilter
begin
  consts PositiveProperty::"(e⇒σ)⇒σ" ("P")
  axiomatization where Ax1: "⌊P φ ∧ P ψ ⊃ P (φ . ψ)⌋"
  axiomatization where Ax2a: "⌊P φ ∨ᵉ P ~φ⌋"
  definition God ("G") where "G x ≡ ∀φ. P φ ⊃ φ x"
(**)  abbreviation PropertyInclusion ("_⊃N_") where "φ ⊃N ψ  ≡  □(∀y::e.  φ y ⊃ ψ y)"
  definition Essence ("_Ess._") where "φ Ess. x ≡ φ x ∧ (∀ψ. ψ x ⊃ (φ ⊃N ψ))"
  axiomatization where Ax2b: "⌊P φ ⊃ □ P φ⌋"
  lemma Ax2b': "⌊¬P φ ⊃ □(¬P φ)⌋" using Ax2a Ax2b by blast
  theorem Th1: "⌊G x ⊃ G Ess. x⌋" using Ax2a Ax2b Essence_def God_def by (smt (verit))
(**)  definition NecExist ("E") where "E x  ≡ ∀φ. φ Ess. x ⊃ □(∃x. φ x)"
  axiomatization where Ax3: "⌊P E⌋"
(**)  theorem Th2: "⌊G x ⊃ □(∃y. G y)⌋" using Ax3 Th1 God_def NecExist_def by smt
(**)  theorem Th3: "⌊◇(∃x. G x) ⊃ □(∃y. G y)⌋" sledgehammer(Th2 Rsymm)  —‹Proof found›
          proof -
(**)        have 1: "⌊(∃x. G x) ⊃ □(∃y. G y)⌋" using Th2 by blast
(**)        have 2: "⌊◇(∃x. G x) ⊃ ◇□(∃y. G y)⌋" using 1 by blast
(**)        have 3: "⌊◇(∃x. G x) ⊃ □(∃y. G y)⌋" using 2 Rsymm by blast
            thus ?thesis by blast
          qed
  axiomatization where Ax4: "⌊P φ ∧ (φ ⊃N ψ) ⊃ P ψ⌋"
  lemma True nitpick[satisfy,card=1,eval="⌊P (λx.⊥)⌋"] oops  —‹One model found of cardinality one›
  abbreviation "PosProps Φ ≡ ∀φ. Φ φ ⊃ P φ"
(**)  abbreviation "ConjOfPropsFrom φ Φ ≡ □(∀z. φ z ↔ (∀ψ. Φ ψ ⊃ ψ z))"
  axiomatization where Ax1Gen: "⌊(PosProps Φ ∧ ConjOfPropsFrom φ Φ) ⊃ P φ⌋"
  lemma L:       "⌊P G⌋"             using Ax1Gen God_def by smt
(**)  theorem Th4: "⌊◇(∃x. G x)⌋"  using Ax2a Ax4 L by blast
(**)  theorem Th5: "⌊□(∃x. G x)⌋"  using Th3 Th4 by blast
  lemma MC:     "⌊φ ⊃ □φ⌋"  sledgehammer(Ax2a Ax2b Th5 God_def Rsymm)  —‹Proof found›
     proof - {fix w fix Q
(**)  have 1: "∀x.(G x w ⟶ (∀Z. Z x ⊃ □(∀z. G z ⊃ Z z)) w)" using Ax2a Ax2b God_def by smt
(**)  have 2: "(∃x. G x w)⟶((Q ⊃ □(∀z. G z ⊃ Q)) w)" using 1 by force
      have 3: "(Q ⊃ □Q) w"using 2 Th5 Rsymm by blast}
      thus ?thesis by auto
     qed
  lemma PosProps:        "⌊P (λx.T) ∧  P (λx. x = x)⌋"    using Ax2a Ax4 by blast
  lemma NegProps:        "⌊¬P(λx.⊥) ∧ ¬P(λx. x ≠ x)⌋"  using Ax2a Ax4 by blast
(**)  lemma UniqueEss1: "⌊φ Ess. x ∧ ψ Ess. x ⊃  □(∀y. φ y ↔ ψ y)⌋" using Essence_def  by smt
(**)  lemma UniqueEss2: "⌊φ Ess. x ∧ ψ Ess. x ⊃  □(φ ≡ ψ)⌋" nitpick[card i=2] oops —‹Countermodel found›
(**)  lemma UniqueEss3:  "⌊φ Ess. x ⊃  □(∀y. φ y ⊃ y ≡ x)⌋" using Essence_def MC by auto
  lemma Monotheism:     "⌊G x ∧ G y ⊃  x ≡ y⌋" using Ax2a God_def by smt
  lemma Filter:        "⌊FilterP P⌋"     using Ax1 Ax4 MC NegProps PosProps Rsymm by smt
  lemma UltraFilter: "⌊UFilterP P⌋"  using Ax2a Filter by smt
  lemma True nitpick[satisfy,card=1,eval="⌊P (λx.⊥)⌋"] oops  —‹One model found of cardinality one›
end
```

Fig. 17 After an appropriate modification of the definition of essence, the inconsistency revealed in Fig. 16 is avoided, and the argument can be successfully verified in modal logic S5 (indeed, as shown, only the modal schema B is actually needed). In contrast to Fig. 7 we here use only possibilist quantifiers to obtain these results. Modified lines compared to Fig. 7 are marked with "(**)" in red

Appendix D: Verification of Scott's variant of Gödel's ontological proof: using possibilist quantifiers only

The results reported for Scott's variant in Sect. 5 are still valid when using only possibilist quantifiers; see Fig. 19. The only difference is that in lemma UniqEssence2 we now find a counterexample only when considering at least two possible worlds, but not for a single possible world as before.

```
theory GoedelVariantHOML3poss imports HOMLinHOL ModalFilter
begin
  consts PositiveProperty::"(e⇒σ)⇒σ" ("P")
  axiomatization where Ax1: "⌊P φ ∧ P ψ ⊃ P (φ . ψ)⌋"
  axiomatization where Ax2a: "⌊P φ ∨ᵉ P ~φ⌋"
  definition God ("G") where "G x ≡ ∀φ. P φ ⊃ φ x"
(**)  abbreviation PropertyInclusion ("_⊃N_") where "φ ⊃N ψ  ≡ □(φ ≠ (λx. ⊥) ∧ (∀y::e.  φ y ⊃ ψ y))"
  definition Essence ("_Ess._") where "φ Ess. x ≡ ∀ψ. ψ x ⊃ (φ ⊃N ψ)"
  axiomatization where Ax2b: "⌊P φ ⊃ □ P φ⌋"
  lemma Ax2b': "⌊¬P φ ⊃ □(¬P φ)⌋" using Ax2a Ax2b by blast
  theorem Th1: "⌊G x ⊃ G Ess. x⌋" using Ax2a Ax2b Essence_def God_def by (smt (verit))
(**)  definition NecExist ("E") where "E x  ≡ ∀φ. (φ Ess. x) ⊃ □(∃x. φ x)"
  axiomatization where Ax3: "⌊P E⌋"
(**)  theorem Th2: "⌊G x ⊃ □(∃y. G y)⌋" using Ax3 Th1 God_def NecExist_def by smt
(**)  theorem Th3: "⌊◇(∃x. G x) ⊃ □(∃y. G y)⌋" sledgehammer(Th2 Rsymm)  —‹Proof found›
          proof -
(**)        have 1: "⌊(∃x. G x) ⊃ □(∃y. G y)⌋" using Th2 by blast
(**)        have 2: "⌊◇(∃x. G x) ⊃ ◇□(∃y. G y)⌋" using 1 by blast
(**)        have 3: "⌊◇(∃x. G x) ⊃ □(∃y. G y)⌋" using 2 Rsymm by blast
            thus ?thesis by blast
         qed
  axiomatization where Ax4: "⌊P φ ∧ (φ ⊃N ψ) ⊃ P ψ⌋"
  lemma True nitpick[satisfy,card=1,eval="⌊P (λx.⊥)⌋"]  oops  —‹One model found of cardinality one›
  abbreviation "PosProps Φ ≡ ∀φ. Φ φ ⊃ P φ"
(**)  abbreviation "ConjOfPropsFrom φ Φ ≡ □(∀z. φ z ↔ (∀ψ. Φ ψ ⊃ ψ z))"
  axiomatization where Ax1Gen: "⌊(PosProps Φ ∧ ConjOfPropsFrom φ Φ) ⊃ P φ⌋"
  lemma L:       "⌊P G⌋"  using Ax1Gen God_def by (smt (verit))
(**)  theorem Th4: "⌊◇(∃x. G x)⌋" sledgehammer[timeout=200](Ax2a L Ax1Gen) sorry  —‹Proof found›
(**)  theorem Th5: "⌊□(∃x. G x)⌋" using Th3 Th4 by blast
  lemma MC:    "⌊φ ⊃ □φ⌋"  sledgehammer(Ax2a Ax2b Th5 God_def Rsymm)  —‹Proof found›
    proof - {fix w fix Q
(**)  have 1: "∀x.(G x w ⟶ (∀Z. Z x ⊃ □(∀z. G z ⊃ Z z)) w)" using Ax2a Ax2b God_def by smt
(**)  have 2: "(∃x. G x w)⟶((Q ⊃ □(∀z. G z ⊃ Q)) w)" using 1 by force
      have 3: "(Q ⊃ □Q) w"using 2 Th5 Rsymm by blast}
      thus ?thesis by auto
     qed
  lemma PosProps:  "⌊P (λx.T) ∧  P(λx. x = x)⌋"     using Ax2a Ax4 L Th4 by smt
  lemma NegProps:  "⌊¬P(λx.⊥) ∧ ¬P(λx. x ≠ x)⌋"  using Ax2a Ax4 L Th4 by smt
(**)  lemma UniqueEss1: "⌊(φ Ess. x) ∧ (ψ Ess. x) ⊃  □(∀y. φ y ↔ ψ y)⌋"  oops  —‹Unclear, open question›
  lemma UniqueEss2: "⌊(φ Ess. x) ∧ (ψ Ess. x) ⊃  □(φ ≡ ψ)⌋"   oops  —‹Unclear, open question›
(**)  lemma UniqueEss3:  "⌊φ Ess. x ⊃  □(∀y. φ y ⊃ y ≡ x)⌋"  using Essence_def MC by auto
  lemma Monotheism: "⌊G x ∧ G y ⊃  x ≡ y⌋" using Ax2a God_def by smt
  lemma Filter:       "⌊FilterP P⌋"      using Ax1 Ax4 MC NegProps PosProps Rsymm by smt
  lemma UltraFilter: "⌊UFilterP P⌋"  using Ax2a Filter by smt
  lemma True nitpick[satisfy,card=1,eval="⌊P (λx.T)⌋"]  oops  —‹One model found of cardinality one›
end
```

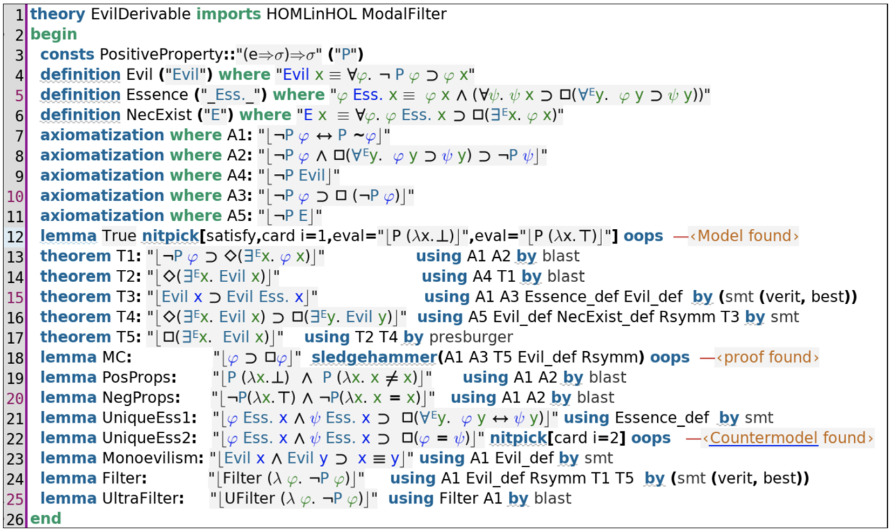
Fig. 18 After an appropriate modification of the definition of necessary property implication, the inconsistency shown in Fig. 16 is avoided, and the argument can be successfully verified. As shown here, this still holds when using possibilist quantifiers only. Modified lines compared to Fig. 8 are marked with "(**)" in red

Appendix E: Necessary existence of an Evil-like entity derived from a modified set of axioms

By rejecting Gödel's assumptions and instead postulating corresponding negative versions of them, as shown in the Fig. 20, the necessary existence of Evil becomes derivable. Note that the empty property $\lambda x.\bot$ and self-difference $\lambda x.x{\neq}x$ now become positive properties, and the tautologous property $\lambda x.\top$ and self-identity $\lambda x.x{=}x$ are now non-positive. The non-positive properties of this "Evil-like" entity are thus apparently identical to the positive properties of Gödel's God-like entity.

```
theory ScottVariantHOMLposs imports HOMLinHOL ModalFilter
begin
  consts PositiveProperty::"(e⇒σ)⇒σ" ("P")
  axiomatization where A1:              "⌊¬P φ ↔ P ~φ⌋"
(**)  axiomatization where A2:          "⌊P φ ∧ □(∀y.  φ y ⊃ ψ y) ⊃ P ψ⌋"
(**)  theorem T1:                       "⌊P φ ⊃ ◇(∃x. φ x)⌋"  using A1 A2  by blast
  definition God ("G") where            "G x ≡ ∀φ. P φ ⊃ φ x"
  axiomatization where A3:              "⌊P G⌋"
(**)  theorem Coro:                    "⌊◇(∃x. G x)⌋"  using A3 T1 by blast
  axiomatization where A4:              "⌊P φ ⊃ □ P φ⌋"
(**)  definition Ess ("_Ess._") where   "φ Ess. x ≡  φ x ∧ (∀ψ. ψ x ⊃ □(∀y::e.  φ y ⊃ ψ y))"
  theorem T2:                           "⌊G x ⊃ G Ess. x⌋"  using A1 A4 Ess_def God_def by fastforce
(**)  definition NecExist ("NE") where  "NE x  ≡ ∀φ. φ Ess. x ⊃ □(∃x. φ x)"
  axiomatization where A5:              "⌊P NE⌋"
  lemma True nitpick[satisfy,card=1,eval="⌊P (λx.⊥)⌋"]  oops  —‹One model found of cardinality one›
(**)  theorem  T3:  "⌊□(∃x. G x)⌋" sledgehammer(A5 Coro God_def NecExist_def Rsymm T2) —‹Proof found›
        proof -
(**)      have   1: "⌊(G x ⊃ NE x) ∧ (G Ess. x ⊃ □(∃x.  G x))⌋" using A5 Ess_def God_def NecExist_def by smt
(**)      hence 2: "⌊(∃x.  G x) ⊃ □(∃x.  G x)⌋"  using A5 God_def NecExist_def T2 by smt
(**)      hence 3: "⌊◇(∃x. G x) ⊃ (◇(□(∃x. G x)) ⊃ □(∃x. G x))⌋"  using Rsymm by blast
          thus ?thesis using 2 Coro by blast
        qed
  lemma MC: "⌊φ ⊃ □φ⌋"   sledgehammer(A1 A4 God_def Rsymm T3)  —‹Proof found›
    proof - {fix w fix Q
(**)  have 1: "∀x.(G x w ⟶ (∀Z. Z x ⊃ □(∀z.((G z) ⊃ (Z z)))) w)" using A1 A4 God_def by smt
(**)  have 2: "(∃x. G x w)⟶((Q ⊃ □(∀z.((G z) ⊃ Q))) w)" using 1 by force
      have 3: "(Q ⊃ □Q) w" using 2 T3 Rsymm by blast}
      thus ?thesis by auto
    qed
  lemma PosProps: "⌊P (λx.⊤) ∧  P (λx. x = x)⌋"       using A1 A2 by blast
  lemma NegProps:  "⌊¬P(λx.⊥) ∧ ¬P(λx. x ≠ x)⌋"   using A1 A2 by blast
(**)  lemma UniqueEss1: "⌊φ Ess. x ∧ ψ Ess. x ⊃  □(∀y. φ y ↔ ψ y)⌋" using Ess_def  by smt
(**)  lemma UniqueEss2: "⌊φ Ess. x ∧ ψ Ess. x ⊃  □(φ ≡ ψ)⌋" nitpick[card i=2] oops —‹Countermodel found›
(**)  lemma UniqueEss3:  "⌊φ Ess. x ⊃  □(∀y. φ y ⊃ y ≡ x)⌋" using Ess_def MC by auto
  lemma Monotheism: "⌊G x ∧ G y ⊃  x ≡ y⌋" using A1 God_def by smt
  lemma Filter:       "⌊FilterP P⌋"     using A1 God_def Rsymm T1 T3 by (smt (verit, best))
  lemma UltraFilter: "⌊UFilterP P⌋"  using Filter A1 by blast
  lemma True nitpick[satisfy,card=1,eval="⌊P (λx.⊥)⌋"] oops  —‹One model found of cardinality one›
end
```

Fig. 19 Scott's variant of Gödel's ontological proof is still valid when using possibilist quantifiers only. Modified lines compared to Fig. 12 are marked with "(**)" in red



```
theory EvilDerivable imports HOMLinHOL ModalFilter
begin
  consts PositiveProperty::"(e⇒σ)⇒σ" ("P")
  definition Evil ("Evil") where "Evil x ≡ ∀φ. ¬ P φ ⊃ φ x"
  definition Essence ("_Ess._") where "φ Ess. x ≡  φ x ∧ (∀ψ. ψ x ⊃ □(∀ᴱy.  φ y ⊃ ψ y))"
  definition NecExist ("E") where "E x  ≡ ∀φ. φ Ess. x ⊃ □(∃ᴱx. φ x)"
  axiomatization where A1: "⌊¬P φ ↔ P ~φ⌋"
  axiomatization where A2: "⌊¬P φ ∧ □(∀ᴱy.  φ y ⊃ ψ y) ⊃ ¬P ψ⌋"
  axiomatization where A4: "⌊¬P Evil⌋"
  axiomatization where A3: "⌊¬P φ ⊃ □ (¬P φ)⌋"
  axiomatization where A5: "⌊¬P E⌋"
  lemma True nitpick[satisfy,card i=1,eval="⌊P (λx.⊥)⌋",eval="⌊P (λx.⊤)⌋"] oops  —‹Model found›
  theorem T1: "⌊¬P φ ⊃ ◇(∃ᴱx. φ x)⌋"                   using A1 A2 by blast
  theorem T2: "⌊◇(∃ᴱx. Evil x)⌋"                        using A4 T1 by blast
  theorem T3: "⌊Evil x ⊃ Evil Ess. x⌋"                  using A1 A3 Essence_def Evil_def  by (smt (verit, best))
  theorem T4: "⌊◇(∃ᴱx. Evil x) ⊃ □(∃ᴱy. Evil y)⌋"  using A5 Evil_def NecExist_def Rsymm T3 by smt
  theorem T5: "⌊□(∃ᴱx.  Evil x)⌋"    using T2 T4 by presburger
  lemma MC:            "⌊φ ⊃ □φ⌋"  sledgehammer(A1 A3 T5 Evil_def Rsymm) oops  —‹proof found›
  lemma PosProps:      "⌊P (λx.⊥)  ∧  P (λx. x ≠ x)⌋"      using A1 A2 by blast
  lemma NegProps:      "⌊¬P(λx.⊤) ∧ ¬P(λx. x = x)⌋"   using A1 A2 by blast
  lemma UniqueEss1:   "⌊φ Ess. x ∧ ψ Ess. x ⊃  □(∀ᴱy.  φ y ↔ ψ y)⌋" using Essence_def  by smt
  lemma UniqueEss2:   "⌊φ Ess. x ∧ ψ Ess. x ⊃  □(φ = ψ)⌋" nitpick[card i=2] oops   —‹Countermodel found›
  lemma Monoevilism: "⌊Evil x ∧ Evil y ⊃  x ≡ y⌋" using A1 Evil_def by smt
  lemma Filter:        "⌊Filter (λ φ. ¬P φ)⌋"     using A1 Evil_def Rsymm T1 T5  by (smt (verit, best))
  lemma UltraFilter:   "⌊UFilter (λ φ. ¬P φ)⌋"  using Filter A1 by blast
end
```

Fig. 20 The necessary existence of an Evil-like entity proved from (controversially) modified assumptions

Acknowledgements We thank Bruno Woltzenlogel-Paleo and David Fuenmayor for their contributions to previous related work. We are also grateful to the reviewers of this article for their very valuable feedback. We also thank Deepl, which was used to slightly rewrite some textual representations.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Adams, R.M.: Introductory note to *1970. In: Kurt Gödel: Collected Works. Oxford University Press, Oxford (1995). https://doi.org/10.1093/oso/9780195072556.003.0017
2. Wang, H.: A Logical Journey: From Gödel to Philosophy. The MIT Press, Cambridge (1997). https://doi.org/10.7551/mitpress/4321.001.0001
3. Sobel, J.H.: Logic and Theism: Arguments for and Against Beliefs in God. Cambridge University Press, Cambridge (2004). https://doi.org/10.1017/CBO9780511497988
4. Kanckos, A., Lethen, T.: The development of Gödel's ontological proof. Rev. Symbol. Logic **14**(4), 1011–1029 (2021). https://doi.org/10.1017/S1755020319000479
5. Gödel, K.: Appendix A. Notes in Kurt Gödel's Hand. In: Sobel, J.H. (ed.) Logic and Theism: Arguments for and Against Beliefs in God, pp. 144–145. Cambridge University Press, Cambridge (2004)
6. Scott, D.S.: Appendix B: Notes in Dana Scott's Hand. In: Sobel, J.H. (ed.) Logic and Theism: Arguments for and Against Beliefs in God, pp. 145–146. Cambridge University Press, Cambridge (2004). https://doi.org/10.1017/CBO9780511497988

7. Benzmüller, C., Woltzenlogel Paleo, B.: Automating Gödel's ontological proof of God's existence with higher-order automated theorem provers. In: ECAI 2014. Frontiers in Artificial Intelligence and Applications, vol. 263, pp. 93–98. IOS Press, Amsterdam (2014). https://doi.org/10.3233/978-1-61499-419-0-93
8. Benzmüller, C., Woltzenlogel Paleo, B.: Interacting with modal logics in the Coq proof assistant. In: Beklemishev, L.D., Musatov, D.V. (eds.) Computer Science—Theory and Applications—10th International Computer Science Symposium in Russia, CSR 2015, Listvyanka, Russia, July 13–17, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9139, pp. 398–411. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20297-6_25
9. Benzmüller, C., Woltzenlogel Paleo, B.: The inconsistency in Gödel's ontological argument: a success story for AI in metaphysics. In: Kambhampati, S. (ed.) IJCAI 2016, vol. 1–3, pp. 936–942. AAAI Press, Palo Alto (2016). http://www.ijcai.org/Proceedings/16/Papers/137.pdf
10. Bentkamp, A., Blanchette, J., Nummelin, V., Tourret, S., Vukmirovic, P., Waldmann, U.: Mechanical mathematicians. Commun. ACM **66**(4), 80–90 (2023). https://doi.org/10.1145/3557998
11. Benzmüller, C., Miller, D.: Automation of higher-order logic. In: Gabbay, D.M., Siekmann, J.H., Woods, J. (eds.) Handbook of the History of Logic. Computational Logic, vol. 9, pp. 215–254. Elsevier, North Holland (2014). https://doi.org/10.1016/B978-0-444-51624-4.50005-8
12. Anderson, C.A.: Some emendations of Gödel's ontological proof. Faith Philos. **7**(3), 291–303 (1990). https://doi.org/10.5840/faithphil19907325
13. Anderson, C.A., Gettings, M.: Gödel's ontological proof revisited. In: Hájek, P. (ed.) Gödel'96: Logical Foundations of Mathematics, Computer Science, and Physics: Lecture Notes in Logic 6, pp. 167–172. Lecture Notes in Logic. Cambridge University Press, Cambridge (2017). https://doi.org/10.1017/9781316716939.011
14. Hájek, P.: Magari and others on Gödel's ontological proof. In: Ursini, A., Agliano, P. (eds.) Logic and Algebra, pp. 125–135. Routledge, New York (1996). https://doi.org/10.1201/9780203748671
15. Hájek, P.: Der Mathematiker und die Frage der Existenz Gottes. In: Al., B. (ed.) Kurt Gödel. Wahrheit und Beweisbarkeit, pp. 325–336. öbv & hpt Verlagsgesellschaft mbH, Wien (2001)
16. Hájek, P.: A new small emendation of Gödel's ontological proof. Stud. Logica **71**(2), 149–164 (2002). https://doi.org/10.1023/A:1016583920890
17. Fitting, M.: Types, Tableaus, and Gödel's God. Springer, Dordrecht (2002). https://doi.org/10.1007/978-94-010-0411-4
18. Bjørdal, F.A.: All properties are divine or god exists—the sacred thesis and its ontological argument. Logic Logic. Philos. **27**, 3 (2018). https://doi.org/10.12775/LLP.2017.027
19. Benzmüller, C., Paulson, L.: Quantified multimodal logics in simple type theory. Log. Univers. **7**(1), 7–20 (2013). https://doi.org/10.1007/s11787-012-0052-y
20. Gallin, D. (ed.): Intensional and Higher-Order Modal Logic. North-Holland Mathematics Studies. Elsevier, North Holland (1975). https://www.sciencedirect.com/bookseries/north-holland-mathematics-studies/vol/19/
21. Muskens, R.: Higher order modal logic. In: Blackburn, P., Benthem, J., Wolter, F. (eds.) Handbook of Modal Logic Studies in Logic and Practical Reasoning, vol. 3, pp. 621–653. Elsevier, North-Holland (2007). https://doi.org/10.1016/S1570-2464(07)80013-9
22. Fitting, M., Mendelsohn, R.L.: First-Order Modal Logic. Synthese Library, 2nd edn. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-40714-7
23. Church, A.: A formulation of the simple theory of types. J. Symb. Log. **5**, 56–68 (1940). https://doi.org/10.2307/2266170
24. Andrews, P.B.: An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof. Applied Logic Series, vol. 27. Springer, Dordrecht (2002). https://doi.org/10.1007/978-94-015-9934-4
25. Andrews, P.B.: Classical type theory. In: Robinson, J.A., Voronkov, A. (eds.) Handbook of Automated Reasoning, vol. 2, pp. 965–1007. Elsevier and MIT Press, Cambridge (2001). https://doi.org/10.1016/B978-044450813-3/50017-5
26. Benzmüller, C., Brown, C., Kohlhase, M.: Higher-order semantics and extensionality. J. Symb. Log. **69**(4), 1027–1088 (2004). https://doi.org/10.2178/jsl/1102022211
27. Benzmüller, C., Andrews, P.: Church's type theory. In: Zalta, E.N., Nodelman, U. (eds.) The Stanford Encyclopedia of Philosophy, Spring 2024 edn. Metaphysics Research Lab, Stanford University (2024). https://plato.stanford.edu/archives/spr2024/entries/type-theory-church/

28. Farmer, W.M.: Simple Type Theory—A Practical Logic for Expressing and Reasoning About Mathematical Ideas. Birkhäuser, Cham (2023). https://doi.org/10.1007/978-3-031-21112-6
29. Benzmüller, C.: Universal (meta-)logical reasoning: recent successes. Sci. Comput. Program. **172**, 48–62 (2019). https://doi.org/10.1016/j.scico.2018.10.008
30. Kirchner, D., Benzmüller, C., Zalta, E.N.: Computer science and metaphysics: a cross-fertilization. Open Philos. **2**, 230–251 (2019). https://doi.org/10.1515/opphil-2019-0015
31. Benzmüller, C., Parent, X., Torre, L.: Designing normative theories for ethical and legal reasoning: LogiKEy framework, methodology, and tool support. Artif. Intell. **287**, 103348 (2020). https://doi.org/10.1016/j.artint.2020.103348
32. Benzmüller, C., Weber, L., Woltzenlogel Paleo, B.: Computer-assisted analysis of the Anderson–Hájek controversy. Log. Univers. **11**(1), 139–151 (2017). https://doi.org/10.1007/s11787-017-0160-9
33. Benzmüller, C., Woltzenlogel Paleo, B.: Experiments in computational metaphysics: Gödel's proof of God's existence. Savijnanam: scientific exploration for a spiritual paradigm. J. Bhaktivedanta Inst. **9**, 43–57 (2017)
34. Fuenmayor, D., Benzmüller, C.: Automating emendations of the ontological argument in intensional higher-order modal logic. In: KI 2017: Advances in Artificial Intelligence. LNAI, vol. 10505, pp. 114–127. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67190-1_9
35. Benzmüller, C., Fuenmayor, D.: Computer-supported analysis of positive properties, ultrafilters and modal collapse in variants of Gödel's ontological argument. Bull. Sect. Logic **49**(2), 127–148 (2020). https://doi.org/10.18778/0138-0680.2020.08
36. Benzmüller, C.: Symbolic AI and Gödel's ontological argument. Zygonr **57**, 953–962 (2022). https://doi.org/10.1111/zygo.12830
37. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL—A Proof Assistant for Higher-Order Logic. LNCS, vol. 2283. Springer, Berlin (2002). https://doi.org/10.1007/3-540-45949-9
38. Bayer, J., Benzmüller, C., Buzzard, K., David, M., Lamport, L., Matiyasevich, Y., Paulson, L., Schleicher, D., Stock, B., Zelmanov, E.: Mathematical proof between generations. Notices AMS **71**(1), 79–92 (2024). https://doi.org/10.1090/noti2860
39. Aigner, M., Ziegler, G.M.: Proofs from THE BOOK, 6th edn. Springer, Berlin (2018). https://doi.org/10.1007/978-3-662-57265-8
40. Vestrucci, A., Benzmüller, C.: Reflections on the ontological proof of Kurt Gödel. In: Marcacci, F., Allen, P. (eds.) Divined Explanations. The Theological and Philosophical Context for the Development of the Sciences (1600–2000). Philosophical Studies in Science and Religion, pp. 255–285. Brill, Leiden (2024). https://doi.org/10.1163/9789004701908_013
41. Passon, O., Benzmüller, C., Falkenburg, B. (eds.): On Gödel and the Nonexistence of Time - Gödel und die Nichtexistenz der Zeit, pp. 1–153. Springer, Heidelberg (2023). https://doi.org/10.1007/978-3-662-67045-3
42. Mühlenbeck, C.: Der Zusammenhang zwischen Gödels Unvollständigkeitssätzen, der Kontinuumshypothese und dem Ontologischen Gottesbeweis und seine Auswirkungen auf die metamathematische Grundlegung der Mathematik. Bachelorarbeit, Freie Universität Berlin (2024). https://doi.org/10.17169/refubium-43924
43. Mühlenbeck, C., Benzmüller, C.: On the maximum of positive properties and modal collapse in Gödel's ontological argument compared to its variants. In: Formal Methods and Science in Philosophy V, International Conference, Book of Abstracts, pp. 31–32 (2024). https://iuc.hr/file/1802
44. Benzmüller, C., Scott, D.S.: Notes on Gödel's and Scott's variants of the ontological argument (Isabelle/HOL dataset). Archive of Formal Proofs (2025). https://www.isa-afp.org/entries/Notes_On_Goedels_Ontological_Argument.html
45. Nipkow, T., Klein, G.: Concrete Semantics-With Isabelle/HOL. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10542-0
46. Bertot, Y., Castéran, P.: Interactive Theorem Proving and Program Development-Coq'Art: The Calculus of Inductive Constructions. Texts in Theoretical Computer Science. An EATCS Series, Springer, Berlin (2004). https://doi.org/10.1007/978-3-662-07964-5
47. Moura, L., Ullrich, S.: The Lean 4 theorem prover and programming language. In: Platzer, A., Sutcliffe, G. (eds.) Automated Deduction—CADE 28—28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12699, pp. 625–635. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79876-5_37

48. Avigad, J.: Automated reasoning for mathematics. In: Benzmüller, C., Heule, M.J.H., Schmidt, R.A. (eds.) Automated Reasoning—12th International Joint Conference, IJCAR 2024, Nancy, France, July 3–6, 2024, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14739, pp. 3–20. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-63498-7_1
49. Slind, K., Norrish, M.: A brief overview of HOL4. In: Mohamed, O.A., Muñoz, C.A., Tahar, S. (eds.) Theorem Proving in Higher Order Logics, 21st International Conference, TPHOLs 2008, Montreal, Canada, August 18–21, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5170, pp. 28–32. Springer, Berlin (2008). https://doi.org/10.1007/978-3-540-71067-7_6
50. Harrison, J.: HOL light: An overview. In: Berghofer, S., Nipkow, T., Urban, C., Wenzel, M. (eds.) Theorem Proving in Higher Order Logics, 22nd International Conference, TPHOLs 2009, Munich, Germany, August 17–20, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5674, pp. 60–66. Springer, Berlin (2009). https://doi.org/10.1007/978-3-642-03359-9_4
51. Owre, S., Rushby, J.M., Shankar, N.: PVS: A prototype verification system. In: Kapur, D. (ed.) Automated Deduction—CADE-11, 11th International Conference on Automated Deduction, Saratoga Springs, NY, USA, June 15–18, 1992, Proceedings. Lecture Notes in Computer Science, vol. 607, pp. 748–752. Springer, Berlin (1992). https://doi.org/10.1007/3-540-55602-8_217
52. Constable, R.L., Allen, S.F., Bromley, M., Cleaveland, R., Cremer, J.F., Harper, R., Howe, D.J., Knoblock, T.B., Mendler, N.P., Panangaden, P., Sasaki, J.T., Smith, S.F.: Implementing Mathematics with the Nuprl Proof Development System. Prentice Hall (1986). https://nuprl-web.cs.cornell.edu/book/
53. Bove, A., Dybjer, P., Norell, U.: A brief overview of Agda—a functional language with dependent types. In: Berghofer, S., Nipkow, T., Urban, C., Wenzel, M. (eds.) Theorem Proving in Higher Order Logics, 22nd International Conference, TPHOLs 2009, Munich, Germany, August 17–20, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5674, pp. 73–78. Springer, Berlin (2009). https://doi.org/10.1007/978-3-642-03359-9_6
54. Hales, T.C., Adams, M., Bauer, G.J., Dang, D.T., Harrison, J., Le Hoang, T., Kaliszyk, C., Magron, V., McLaughlin, S., Nguyen, T.T., Nguyen, T.Q., Nipkow, T., Obua, S., Pleso, J., Rute, J.M., Solovyev, A., Thi Ta, A.H., Tran, T.N., Trieu, D.T., Urban, J., Vu, K.K., Zumkeller, R.: A formal proof of the Kepler conjecture. Forum Math. (2015). https://doi.org/10.1017/fmp.2017.1
55. Dalrymple, D., Skalse, J., Bengio, Y., Russell, S., Tegmark, M., Seshia, S., Omohundro, S., Szegedy, C., Goldhaber, B., Ammann, N., Abate, A., Halpern, J., Barrett, C., Zhao, D., Zhi-Xuan, T., Wing, J., Tenenbaum, J.: Towards guaranteed safe AI: a framework for ensuring robust and reliable AI systems (2024). https://arxiv.org/abs/2405.06624
56. Lenat, D., Marcus, G.: Getting from generative AI to trustworthy AI: what LLMs might learn from Cyc (2023). arXiv:2308.04445
57. Platzer, A.: Intersymbolic AI: interlinking symbolic AI and subsymbolic AI (2024). https://arxiv.org/abs/2406.11563
58. Benzmüller, C., Lomfeld, B.: Reasonable machines: a research manifesto. In: Schmid, U., Klügl, F., Wolter, D. (eds.) KI 2020: Advances in Artificial Intelligence—43rd German Conference on Artificial Intelligence, Bamberg, Germany, September 21–25, 2020, Proceedings. Lecture Notes in Artificial Intelligence, vol. 12352, pp. 251–258. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58285-2_20
59. Arkin, R.C.: Governing lethal behavior: embedding ethics in a hybrid deliberative/reactive robot architecture. In: Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction. HRI '08, pp. 121–128. Association for Computing Machinery, New York (2008).https://doi.org/10.1145/1349822.1349839
60. Moor, J.: Four kinds of ethical robots. Philos. Now **72**, 12–14 (2009)
61. Arkin, R.C.: Civilized collaboration: ethical architectures for enforcing legal requirements and mediating social norms in HRI. In: 2015 International Conference on Collaboration Technologies and Systems, CTS 2015, Atlanta, GA, USA, June 1–5, 2015, p. 38. IEEE, Piscataway (2015). https://doi.org/10.1109/CTS.2015.7210394
62. Govindarajulu, N.S., Bringsjord, S.: Ethical regulation of robots must be embedded in their operating systems. In: Trappl, R. (ed.) A Construction Manual for Robots' Ethical Systems—Requirements, Methods, Implementations. Cognitive Technologies, pp. 85–99. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21548-8_5
63. Scheutz, M.: The case for explicit ethical agents. AI Mag. **38**(4), 57–64 (2017). https://doi.org/10.1609/aimag.v38i4.2746

64. Zhou, J.P., Staats, C., Li, W., Szegedy, C., Weinberger, K.Q., Wu, Y.: Don't trust: verify - grounding LLM quantitative reasoning with autoformalization. In: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7–11, 2024 (2024). https://openreview.net/forum?id=V5tdi14ple
65. Russell, B.: Logical positivism. Revue Internationale de Philosophie **4**(11), 3–19 (1950)
66. Leibniz, G., Gerhardt, C.I.: Die Philosophischen Schriften, vol. 7. Adegi Graphics LLC, New York (1999)
67. Blanchette, J.C., Böhme, S., Paulson, L.C.: Extending Sledgehammer with SMT solvers. J. Autom. Reason. **51**(1), 109–128 (2013). https://doi.org/10.1007/978-3-642-22438-6_11
68. Blanchette, J.C., Nipkow, T.: Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In: Kaufmann, M., Paulson, L.C. (eds.) Interactive Theorem Proving—ITP 2010. LNCS, vol. 6172, pp. 131–146. Springer, Berlin (2010).https://doi.org/10.1007/978-3-642-14052-5_11
69. Wenzel, M.: Isabelle, Isar—a versatile environment for human readable formal proof documents. PhD thesis, Technical University Munich, Germany (2002). https://d-nb.info/963850776/34
70. Rothgang, C., Rabe, F., Benzmüller, C.: Theorem proving in dependently-typed higher-order logic. In: Pientka, B., Tinelli, C. (eds.) Automated Deduction—CADE 29—29th International Conference on Automated Deduction, Rome, Italy, July 1–4, 2023, Proceedings. Lecture Notes in Computer Science, vol. 14132, pp. 438–455. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-38499-8_25
71. Niederhauser, J., Brown, C.E., Kaliszyk, C.: Tableaux for automated reasoning in dependently-typed higher-order logic. In: Benzmüller, C., Heule, M.J.H., Schmidt, R.A. (eds.) Automated Reasoning—12th International Joint Conference, IJCAR 2024, Nancy, France, July 3–6, 2024, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14739, pp. 86–104. Springer, Chame (2024). https://doi.org/10.1007/978-3-031-63498-7_6
72. Sutcliffe, G., Benzmüller, C.: Automated reasoning in higher-order logic using the TPTP THF infrastructure. J. Formal. Reason. **3**(1), 1–27 (2010). https://doi.org/10.6092/issn.1972-5787/1710
73. Sutcliffe, G.: The TPTP problem library and associated infrastructure—from CNF to TH0, TPTP v6.4.0. J. Autom. Reason. **59**(4), 483–502 (2017). https://doi.org/10.1007/S10817-017-9407-7
74. Sutcliffe, G.: The CADE ATP system competition—CASC. AI Mag. **37**(2), 99–101 (2016). https://doi.org/10.1609/AIMAG.V37I2.2620
75. Kaliszyk, C., Urban, J.: Hol(y)hammer: online ATP service for HOL light. Math. Comput. Sci. **9**(1), 5–22 (2015). https://doi.org/10.1007/S11786-014-0182-0
76. Bentkamp, A., Blanchette, J., Tourret, S., Vukmirovic, P.: Superposition for higher-order logic. J. Autom. Reason. **67**(1), 10 (2023). https://doi.org/10.1007/S10817-022-09649-9
77. Bhayat, A., Suda, M.: A higher-order vampire (short paper). In: Benzmüller, C., Heule, M.J.H., Schmidt, R.A. (eds.) Automated Reasoning—12th International Joint Conference, IJCAR 2024, Nancy, France, July 3–6, 2024, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14739, pp. 75–85. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-63498-7_5
78. Schulz, S., Cruanes, S., Vukmirovic, P.: Faster, higher, stronger: E 2.3. In: Fontaine, P. (ed.) Automated Deduction—CADE 27—27th International Conference on Automated Deduction, Natal, Brazil, August 27–30, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11716, pp. 495–507. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29436-6_29
79. Benzmüller, C., Sultana, N., Paulson, L.C., Theiß, F.: The higher-order prover LEO-II. J. Autom. Reason. **55**(4), 389–404 (2015). https://doi.org/10.1007/s10817-015-9348-y
80. Steen, A., Benzmüller, C.: Extensional higher-order paramodulation in Leo-III. J. Autom. Reason. **65**(6), 775–807 (2021). https://doi.org/10.1007/S10817-021-09588-X
81. Barbosa, H., Barrett, C.W., Brain, M., Kremer, G., Lachnitt, H., Mann, M., Mohamed, A., Mohamed, M., Niemetz, A., Nötzli, A., Ozdemir, A., Preiner, M., Reynolds, A., Sheng, Y., Tinelli, C., Zohar, Y.: cvc5: a versatile and industrial-strength SMT solver. In: Fisman, D., Rosu, G. (eds.) Tools and Algorithms for the Construction and Analysis of Systems—28th International Conference, TACAS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2–7, 2022, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13243, pp. 415–442. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-99524-9_24
82. Brown, C.E.: Satallax: an automatic higher-order prover. In: Gramlich, B., Miller, D., Sattler, U. (eds.) Automated Reasoning—6th International Joint Conference, IJCAR 2012, Manchester, UK, June 26–29, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7364, pp. 111–117. Springer, Berlin (2012). https://doi.org/10.1007/978-3-642-31365-3_11

83. Heule, M., Kullmann, O.: The science of brute force. Commun. ACM **60**(8), 70–79 (2017). https://doi.org/10.1145/3107239
84. Yolcu, E., Aaronson, S., Heule, M.: An automated approach to the Collatz conjecture. J. Autom. Reason. **67**(2), 15 (2023). https://doi.org/10.1007/S10817-022-09658-8
85. Andrews, P.B., Bishop, M., Issar, S., Nesmith, D., Pfenning, F., Xi, H.: Tps: a theorem-proving system for classical type theory. J. Autom. Reason. **16**, 321–353 (1996). https://doi.org/10.1007/BF00252180
86. Dechesne, F., Nederpelt, R.P.: N.g. de bruijn (1918–2012) and his road to automath, the earliest proof checker. Math. Intell. **34**, 4–11 (2012). https://doi.org/10.1007/s00283-012-9324-x
87. Andrews, P.B.: Some reflections on proof transformations. In: Hutter, D., Stephan, W. (eds.) Mechanizing Mathematical Reasoning: Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday, pp. 14–29. Springer, Berlin (2005). https://doi.org/10.1007/978-3-540-32254-2_2
88. Andrews, P.B., Brown, C.E.: Tps: a hybrid automatic-interactive system for developing proofs. J. Appl. Log. **4**(4), 367–395 (2006). https://doi.org/10.1016/j.jal.2005.10.002
89. Henkin, L.: Completeness in the theory of types. J. Symb. Log. **15**(2), 81–91 (1950). https://doi.org/10.2307/2266967
90. Andrews, P.B.: General models and extensionality. J. Symb. Log. **37**(2), 395–397 (1972). https://doi.org/10.2307/2272982
91. Andrews, P.B.: General models, descriptions, and choice in type theory. J. Symb. Log. **37**(2), 385–394 (1972). https://doi.org/10.2307/2272981
92. Gödel, K.: Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme i. Monatshefte für Mathematik und Physik **38**, 173–198 (1931). https://doi.org/10.1007/BF01700692
93. Brown, C.E.: Set comprehension in Church's type theory. PhD thesis, Department of Mathematical Sciences, Carnegie Mellon University (2004)
94. Benzmüller, C., Fuenmayor, D.: Cantor's theorem without Reductio Ad Absurdum (2021). https://doi.org/10.13140/RG.2.2.31069.95201/1
95. Sutcliffe, G.: Stepping stones in the TPTP world. In: Benzmüller, C., Heule, M.J.H., Schmidt, R.A. (eds.) Automated Reasoning—12th International Joint Conference, IJCAR 2024, Nancy, France, July 3–6, 2024, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14739, pp. 30–50. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-63498-7_3
96. Garson, J.: Modal logic. In: Zalta, E.N., Nodelman, U. (eds.) The Stanford Encyclopedia of Philosophy, Spring 2024 edn. Metaphysics Research Lab, Stanford University (2024). https://plato.stanford.edu/archives/spr2024/entries/logic-modal/
97. Cresswell, M.J., Hughes, G.E.: A New Introduction to Modal Logic. Routledge, London (1996). https://doi.org/10.4324/9780203028100
98. Fuenmayor, D.: Semantical investigations on non-classical logics with recovery operators: negation. Logic J. IGPL **8**, 1–42 (2023). https://doi.org/10.1093/jigpal/jzad013
99. Uzquiano, G.: Quantifiers and quantification. In: Zalta, E.N., Nodelman, U. (eds.) The Stanford Encyclopedia of Philosophy, Winter 2022 edn. Metaphysics Research Lab, Stanford University (2022). https://plato.stanford.edu/archives/win2022/entries/quantification/
100. Menzel, C.: The Possibilism-actualism debate. In: Zalta, E.N., Nodelman, U. (eds.) The Stanford Encyclopedia of Philosophy, Summer 2024 edn. Metaphysics Research Lab, Stanford University (2024). https://plato.stanford.edu/archives/sum2024/entries/possibilism-actualism/
101. Cocchiarella, N.B.: A completeness theorem in second order modal logic. Theoria **35**(2), 81–103 (1969). https://doi.org/10.1111/j.1755-2567.1969.tb00361.x
102. Scott, D.S.: Existence and description in formal logic. In: Schoenman, R. (ed.) Bertrand Russell: Philosopher of the Century, pp. 181–200. George Allen & Unwin, London (1967). (Reprinted with additions in: Philosophical Applications of Free Logic, edited by K. Lambert. Oxford Universitry Press, 1991, pp. 28–48) https://github.com/CMU-HoTT/scott/blob/main/pdfs/1967-existence-and-description-in-formal-logic.pdf
103. Lambert, K.: The definition of e(xistence)! in free logic. In: Abstracts: The International Congress for Logic, Methodology and Philosophy of Science. Stanford University Press, Stanford (1960)
104. Lambert, K.: Existential import revisited. Notre Dame J. Formal Logic **4**(4), 288–292 (1963). https://doi.org/10.1305/ndjfl/1093957655
105. Benzmüller, C., Scott, D.S.: Automating free logic in HOL, with an experimental application in category theory. J. Autom. Reason. **64**(1), 53–72 (2020). https://doi.org/10.1007/s10817-018-09507-7

106. Kirchner, D., Benzmüller, C., Zalta, E.N.: Mechanizing principia logico-metaphysica in functional type theory. Rev. Symbol. Logic **13**(1), 206–218 (2020). https://doi.org/10.1017/S1755020319000297
107. Kirchner, D.: Computer-verified foundations of metaphysics and an ontology of natural numbers in Isabelle/HOL. Dissertation, Freie Universität Berlin (2022). https://doi.org/10.17169/refubium-35141
108. Benzmüller, C., Woltzenlogel Paleo, B.: An object-logic explanation for the inconsistency in Gödel's ontological theory (extended abstract, sister conferences). In: Helmert, M., Wotawa, F. (eds.) KI 2016: Advances in Artificial Intelligence, Proceedings. Lecture Notes in Computer Science, pp. 244–250. Springer, Berlin (2016). https://doi.org/10.1007/978-3-319-46073-4
109. Kovač, S.: Modal collapse in Gödel's ontological proof. In: Szatkowski, M. (ed.) Ontological Proofs Today, pp. 50–323. Ontos Verlag, Heusenstamm (2012). https://doi.org/10.1515/9783110325881.323
110. Hazen, A.P.: On Gödel's ontological proof. Australas. J. Philos. **76**(3), 361–377 (1998). https://doi.org/10.1080/00048409812348501
111. Studtmann, P.: God and the numbers. J. Philos. **120**, 641–655 (2023). https://doi.org/10.5840/jphil20231201235
112. Friedman, H.: In: Szatkowski, M. (ed.) A Divine Consistency Proof for Mathematics, pp. 645–696. De Gruyter, Berlin. https://doi.org/10.1515/9783111332536-034
113. Crocco, G., Engelen, E.-M.: Kurt Gödel's philosophical remarks (max phil). In: Crocco, G., Engelen, E.-M. (eds.) Kurt Gödel Philosopher-Scientist. Presses universitaires de Provence, Aix-en-Provence (2016). https://doi.org/10.4000/books.pup.53560
114. Gödel, K.: Philosophische Notizbücher/Philosophical Notebooks vol. 1–6. De Gruyter, Berlin, Boston (2019). (Edited by Eva-Maria Engelen, Translation by Carl Merlin). https://www.degruyter.com/serial/goedel-b/html