Applying Second-Order Quantifier Elimination in Inspecting Gödel's Ontological Proof

Christoph Wernhard

University of Potsdam, Germany

Abstract. In recent years, Gödel's ontological proof and variations of it were formalized and analyzed with automated tools in various ways. We supplement these analyses with a modeling in an automated environment based on first-order logic extended by predicate quantification. Formula macros are used to structure complex formulas and tasks. The analysis is presented as a generated type-set document where informal explanations are interspersed with pretty-printed formulas and outputs of reasoners for first-order theorem proving and second-order quantifier elimination. Previously unnoticed or obscured aspects and details of Gödel's proof become apparent. Practical application possibilities of second-order quantifier elimination are shown and the encountered elimination tasks may serve as benchmarks.

1 Introduction

Kurt Gödel's ontological proof is bequeathed in hand-written notes by himself [14] and by Dana Scott [25]. Since transcriptions of these notes were published in 1987 [26], the proof was analyzed, formalized and varied in many different logical settings.[1] Books by John Howard Sobel [27] and Melvin Fitting [13] include comprehensive discussions. Branden Fitelson, Paul E. Oppenheimer and Edward N. Zalta [12,24] investigated various metaphysical arguments in an automated first-order setting based on *Prover9* [20]. Christoph Benzmüller and Bruno Woltzenlogel Paleo [7] initiated in 2014 the investigation of Gödel's argument with automated systems, which led to a large number of follow-up studies concerning its verification with automated tools and the human-readable yet formal representation, e.g., [8,16,6,17,5]. Here we add to these lines of work an inspection of Gödel's proof in a logical setting that so far has not been considered for this purpose. The expectation is that further, previously unnoticed or obscured aspects and details of the proof become apparent. The used framework, *PIE* ("Proving, Interpolating, Eliminating") [29,30] embeds automated reasoners, in particular for first-order theorem proving and second-order quantifier elimination, in a system for defining formula macros and rendering LaTeX-formatted

[1] Recently discovered further sources by Gödel are presented in [15].

presentations of formula macro definitions and reasoner outputs. In fact, the present paper is the generated output of such a *PIE* document.[2]

The target logic of the macros is second-order logic, or, more precisely classical first-order logic extended by predicate quantifiers. The macro layer and the formulas obtained as expansions can be strictly separated. In our modeling of Gödel's proof we proceed by expressing large-scale steps (axioms, theorems) with macros whose relationships are verified by invocations of embedded reasoners. In this sense our formalization of may be considered as *semi*-automated.

Aside of providing further material for the study of Gödel's proof, the work shows possibilities of applying second-order quantifier elimination in a practical system. It appears that the functionality of the macro mechanism is necessary to express nontrivial applications on the basis of first- and second-order logic. The elimination problems that suggested themselves in the course of the investigation, some of which could not be solved by the current version of *PIE*, may be useful as benchmarks for implemented elimination systems.[3]

The rest of the paper is structured as follows: After introducing preliminaries in Sect. 2, Gödel's proof in the version of Scott is developed in Sect. 3. An approach to obtain the weakest sufficient precondition on the accessibility relation for Gödel's proof with second-order quantifier elimination is then discussed in Sect. 4. Section 5 concludes the paper. Supplementary material is provided in the report version [31] of the paper.

2 Preliminaries

A *PIE* document is a Prolog source file that contains declarative formula macro definitions and specifications of reasoner invocations, interspersed with LaTeX comments in the manner of literate programming [18]. The *PIE* processor expands the formula macros, invokes the reasoners, and compiles a LaTeX document where the formula macro definitions and the results of reasoner invocations are pretty-printed. Alternatively, the processor's functionality is accessible from Prolog, via the interpreter and in programs. The overall processing time for the present paper, including reasoner invocations and LaTeX processing to produce a PDF, is about 2.5 seconds.

Formula macros without parameters can play the role of formula names. Expressions with macros expand into formulas of first-order logic extended with predicate quantifiers. Hence, some means of expression that would naturally be used in a higher-order logic formalization of Gödel's proof are not available in the expansion results. Specifically, predicates in argument position are not permitted and there is no abstraction mechanism to construct predicates from formulas.

[2] The *PIE* source of this paper is available at http://cs.christophwernhard.com/pie.

[3] Another recent system for second-order quantifier elimination on the basis of first-order logic is *DLS-Forgetter* [2], which, like the implementation in *PIE*, is based on the *DLS* algorithm [10]. An older resolution-based system, *SCAN*, [22,23] can currently be invoked via a Web interface.

However, these higher-order features are in Gödel's proof actually only required with respect to specific instances that can be expressed in first-order logic.

As embedded reasoners we used the first-order theorem provers *Prover9* [20] and *CMProver* [9,29], the first-order model generator *Mace4* [20], and an implementation [29,30] of the *DLS* algorithm [10] for second-order quantifier elimination, which is based on Ackermann's Lemma [1]. Reasoner outputs computed during processing of the *PIE* document are presented with the introductory phrases **This formula is valid:**, **This formula is not valid:** and **Result of elimination:**. In addition, various methods for formula simplification, clausification and un-Skolemization are applied in preprocessing, inprocessing and for output presentation.

The generated LaTeX presentation of formulas and macro definitions bears some footprint inherited from the Prolog syntax that is used to write formulas in *PIE* documents. As in Prolog, predicate and constant symbols are written in lower case. Macro parameters and bound logical variables that are to be instantiated with fresh symbols at macro expansion are printed like Prolog variables with a capitalized initial. Where-clauses in macro definitions are used to display in abstracted form auxiliary Prolog code executed at macro expansion.

We write formulas of modal predicate logic as formulas of classical first-order logic with one additional free world variable v by applying the standard translation from [4, Sec. 11.4] (see also [3, Chap. XII]), which can be defined as

$$\begin{aligned}
ST(P(t_1,\ldots,t_n)) &\stackrel{\text{def}}{=} P(v,t_1,\ldots,t_n)\\
ST(\neg F) &\stackrel{\text{def}}{=} \neg ST(F)\\
ST(F \vee G) &\stackrel{\text{def}}{=} ST(F) \vee ST(G)\\
ST(\exists x F) &\stackrel{\text{def}}{=} \exists x\,(\mathsf{e}(v,x) \wedge ST(F))\\
ST(\Diamond F) &\stackrel{\text{def}}{=} \exists w\,(\mathsf{r}(v,w) \wedge \textstyle\bigwedge_{i \text{ s.th. } x_i \text{ free in } F} \mathsf{e}(w,x_i) \wedge ST(F)\{v \mapsto w\})
\end{aligned}$$

An n-ary predicate P in the modal logic is translated into an $n+1$-ary predicate, where the first argument represents a world. The binary predicates r and e are used for world accessibility and membership in the domain of a world. The logic operators $\wedge, \rightarrow, \leftrightarrow, \forall, \Box$ can be understood as shorthands defined in terms of the shown operators. As target logic we neither use a two-sorted logic nor encode two-sortedness explicitly with relativizer predicates. However, the translation of modal formulas yields formulas in which all quantifications are relativized by r or by e, which seems to subsume the effect of such relativizer predicates. To express that free individual symbols are of sort *world* we use the unary predicate world. Macro 4, defined below, can be used as an axiom that relates world and r as far as needed for our purposes. The standard translation realizes with respect to the represented modal logic *varying domain semantics* (*actualist notion of quantification*), expressed with the existence predicate e. *Constant domain semantics* (*possibilist notion of quantification*) can be achieved with axioms that state domain increase and decrease.

As technical basis for Gödel's proof we use the presentation of Scott's version [25] in [7, Fig. 1], shown here as Fig. 1. The identifiers **A1**–**A5**, **T1**–**T3**, **D1**–**D3** and **C** of the involved axioms, theorems, definitions and corollary follow [7]. In addition, Lemma **L** is taken from [6, Fig. 1], where it is appears as **L2**.

A1 Either a property or its negation is positive, but not both
$$\forall P(\mathsf{Pos}(\neg P) \leftrightarrow \neg\mathsf{Pos}(P))$$
A2 A property necessarily implied by a positive property is positive
$$\forall P \forall Q\,((\mathsf{Pos}(P) \wedge \Box\forall x\,(P(x) \rightarrow Q(x))) \rightarrow \mathsf{Pos}(Q))$$
T1 Positive properties are possibly exemplified
$$\forall P\,(\mathsf{Pos}(P) \rightarrow \Diamond\exists x\,P(x))$$
D1 A *God-like* being possesses all positive properties
$$\mathsf{G}(x) \leftrightarrow \forall P\,(\mathsf{Pos}(P) \rightarrow P(x))$$
A3 The property of being God-like is positive
$$\mathsf{Pos}(\mathsf{G})$$
C Possibly, God exists
$$\Diamond\exists x\,\mathsf{G}(x)$$
A4 Positive properties are necessarily positive
$$\forall P\,(\mathsf{Pos}(P) \rightarrow \Box\mathsf{Pos}(P))$$
D2 An *essence* of an individual is a property possessed by it and necessarily implying any of its properties
$$\mathsf{Ess}(P,x) \leftrightarrow P(x) \wedge \forall Q\,(Q(x) \rightarrow \Box\forall y\,(P(y) \rightarrow Q(y)))$$
T2 Being God-like is an essence of any God-like being
$$\forall x\,(\mathsf{G}(x) \rightarrow \mathsf{Ess}(\mathsf{G},x))$$
D3 *Necessary existence* of an individual is the necessary exemplification of all its essences
$$\mathsf{NE}(x) \leftrightarrow \forall P(\mathsf{Ess}(P,x) \rightarrow \Box\exists y\,P(y))$$
A5 Necessary existence is a positive property
$$\mathsf{Pos}(\mathsf{NE})$$
L If a God-like being exists, then necessarily a God-like being exists
$$\exists x\,\mathsf{G}(x) \rightarrow \Box\exists x\,\mathsf{G}(x)$$
T3 Necessarily, God exists
$$\Box\exists x\,\mathsf{G}(x)$$

Fig. 1. Scott's version of Gödel's ontological argument [25], adapted from [7,6].

3 Rendering Gödel's Ontological Proof

3.1 Positiveness – Proving Theorem T1

The first two axioms in Gödel's proof, **A1** and **A2**, are about *positiveness* of properties. Theorem **T1** follows from them. The following macros render the left-to-right direction of **A1** and **A2**, respectively.

Macro 1 $ax_1^{\rightarrow}(V,P)$ is defined as
$$\mathsf{world}(V) \rightarrow (\mathsf{pos}(V,N') \rightarrow \neg\mathsf{pos}(V,P')),$$
where
$$N' := \dot{\neg}P,$$
$$P' := \dot{P}.$$

Is positive is represented here by the binary predicate pos, which has a world and an individual representing a predicate as argument. P' and N' are individual constants that represent a supplied predicate P and its complement $\lambda vx.\neg P(v,x)$, respectively. The *where* clause specifies that at macro expansion they are replaced by individual constants $\dot{P}$ and $\dot{\neg}P$, available for each predicate symbol P.

Throughout this analysis, we expose the current world as macro parameter V, which facilitates identifying proofs steps where an axiom is not just applied with respect to the initially given current world but to some other reachable world.

Macro 2 $ax_2(V,P,Q)$ is defined as

$$\begin{array}{l}\mathsf{world}(V) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \rightarrow \\ (\mathsf{pos}(V,P') \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \wedge \\ \forall W\,(\mathsf{r}(V,W) \qquad\qquad\qquad\qquad\qquad\quad \rightarrow \\ \qquad \forall X\,(\mathsf{e}(W,X) \rightarrow (P(W,X) \rightarrow Q(W,X)))) \qquad\quad \rightarrow \\ \mathsf{pos}(V,Q')),\end{array}$$

where $P' := \dot{P}$, $Q' := \dot{Q}$.

As an insight-providing intermediate step for proving **T1** we can now derive the following lemma using just a single instance of each of $ax_1^{\rightarrow}$ and ax_2, where verum ($\lambda x.x = x$) and falsum ($\lambda x.x \neq x$) are represented as binary predicates $\top$ and $\bot$, whose first argument is a world.

Macro 3 $lemma_1(V)$ is defined as

$$\mathsf{world}(V) \rightarrow \neg\mathsf{pos}(V,\dot{\bot}).$$

Macro 4 r_world_1 is defined as $\forall v \forall w\,(\mathsf{r}(v,w) \rightarrow \mathsf{world}(w))$.

Macro 5 $topbot\_def$ is defined as

$$\begin{array}{ll}\forall v \forall x\,(\mathsf{world}(v) \rightarrow (\top(v,x) \leftrightarrow \mathsf{e}(v,x))) & \wedge \\ \forall v \forall x\,(\mathsf{world}(v) \rightarrow (\bot(v,x) \leftrightarrow \neg\mathsf{e}(v,x))). & \end{array}$$

To express the precondition for $lemma_1$ we need some auxiliary macros concerning $\top$ and $\bot$. The following expresses equivalence of $\top$ and $\lambda vx.\neg\bot(v,x)$.

Macro 6 $topbot\_equiv$ is defined as $\forall v \forall x\,(\mathsf{world}(v) \rightarrow (\top(v,x) \leftrightarrow \neg\bot(v,x)))$.

This formula is valid: $topbot\_def \rightarrow topbot\_equiv$.

The constants $\dot{\top}$, $\dot{\bot}$ and $\neg\dot{\top}$ designate the individuals associated with $\top$, $\bot$ and $\lambda vx.\neg\top(v,x)$, respectively. The following axiom leads from the equivalence expressed by Macro 6 to equality of the associated individuals.

Macro 7 $topbot\_equiv\_equal$ is defined as $topbot\_equiv \rightarrow \dot{\bot} = \neg\dot{\top}$.

Equality is understood there with respect to first-order logic, not qualified by a world parameter. In [31] alternatives are shown, where equality is replaced by a weaker substitutivity property. We can now give the precondition for $lemma_1$.

Macro 8 $pre\_lemma_1(V)$ is defined as

$$\begin{array}{ll} r\_world_1 & \wedge \\ topbot\_def & \wedge \\ topbot\_equiv\_equal & \wedge \\ ax_1^{\rightarrow}(V,\top) & \wedge \\ ax_2(V,\bot,\top). & \end{array}$$

This formula is valid: $pre\_lemma_1(\mathsf{v}) \rightarrow lemma_1(\mathsf{v})$.

T1 can be rendered by the following macro with a predicate parameter.

Macro 9 $thm_1(V,P)$ is defined as

$$\begin{array}{l} \mathsf{world}(V) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \rightarrow \\ (\mathsf{pos}(V,P') \qquad\qquad\qquad\qquad\qquad\qquad\quad \rightarrow \\ \exists W\,(\mathsf{r}(V,W) \wedge \exists X\,(\mathsf{e}(W,X) \wedge P(W,X)))), \end{array}$$

where $P' := \dot{P}$.

Macro 10 $pre\_thm_1(V,P)$ is defined as $lemma_1(V) \wedge ax_2(V,P,\bot)$.

This formula is valid: $pre\_thm_1(\mathsf{v},\mathsf{p}) \rightarrow thm_1(\mathsf{v},\mathsf{p})$.

Instances of $thm_1(V,P)$ can be proven for arbitrary worlds V and predicates P, from the respective instance of the precondition $pre\_thm_1(V,P)$. A further instance of ax_2 – beyond that used to prove $lemma_1$ – is required there, with respect to $\bot$ and the given predicate P.

3.2 Possibly, God Exists – Proving Corollary C

Axiom **A3** and **T1** instantiated by *God-like* together imply corollary **C**. This is rendered as follows, where *God-like* is represented by g.

Macro 11 $ax_3(V)$ is defined as

$$\mathsf{world}(V) \rightarrow \mathsf{pos}(V,\dot{\mathsf{g}}).$$

Macro 12 $coro(V)$ is defined as

$$\mathsf{world}(V) \rightarrow \exists W\,(\mathsf{r}(V,W) \wedge \exists X\,(\mathsf{e}(W,X) \wedge \mathsf{g}(W,X))).$$

Macro 13 $pre\_coro(V)$ is defined as $thm_1(V,\mathsf{g}) \wedge ax_3(V)$.

This formula is valid: $pre\_coro(\mathsf{v}) \rightarrow coro(\mathsf{v})$.

Notice that, differently from the proofs reported in [7, Fig. 2], **C**, represented here by *coro*, can be proven independently from the definition of *God-like*, **D1**, which is represented here by the Macros $def_1^{\rightarrow}$ and $def_1^{\rightarrow\neg}$ defined below.

3.3 Essence – Proving Theorem T2

With macros $def_1^{\rightarrow}$ and $def_1^{\rightarrow\neg}$, defined now, we represent the left-to-right direction of **D1**. Actually, only this direction of **D1** is required for the proving the further theorems.

Macro 14 $def_1^{\rightarrow}(V,X,P)$ is defined as

$$\mathsf{g}(V,X) \rightarrow (\mathsf{pos}(V,P') \rightarrow P(V,X)),$$

where $P' := \dot{P}$.

Macro 15 $def_1^{\rightarrow\neg}(V,X,P)$ is defined as

$$\mathsf{g}(V,X) \rightarrow (\mathsf{pos}(V,P') \rightarrow \neg P(V,X)),$$

where $P' := \dot{\neg} P$.

The following macro *val_ess* renders the definiens of the *essence of* relationship between a predicate and an individual in **D2**. It is originally a formula with predicate quantification, but without application of a predicate to a predicate. The macro *val_ess* exposes the universally quantified predicate as parameter Q, permitting to use it instantiated with some specific predicate.

Macro 16 *val_ess*(V, P, X, Q) is defined as

$$\begin{array}{lll} P(V,X) & & \wedge \\ (Q(V,X) & \rightarrow & \\ \forall W\,(\mathsf{r}(V,W) & \rightarrow & \\ \quad \forall Y\,(\mathsf{e}(W,Y) \rightarrow (P(W,Y) \rightarrow Q(W,Y))))). & & \end{array}$$

The universally quantified version of *val_ess* can be be expressed by prefixing a predicate quantifier upon Q. Eliminating this second-order quantifier shows another view on *essence*.

Input: $\forall q\, \mathit{val\_ess}(\mathsf{v}, \mathsf{p}, \mathsf{x}, q)$.
Result of elimination:

$$\begin{array}{ll} \mathsf{p}(\mathsf{v},\mathsf{x}) & \wedge \\ \forall y \forall z\,(\mathsf{e}(y,z) \wedge \mathsf{p}(y,z) \wedge \mathsf{r}(\mathsf{v},y) \rightarrow y = \mathsf{v}) & \wedge \\ \forall y \forall z\,(\mathsf{e}(y,z) \wedge \mathsf{p}(y,z) \wedge \mathsf{r}(\mathsf{v},y) \rightarrow z = \mathsf{x}). & \end{array}$$

We convert the elimination result "manually" to a more clear form and prove equivalence by referencing to the "last result" via a macro.

Macro 17 *last_result* is defined as F,
where last_ppl_result(F).

This formula is valid: $\mathsf{p}(\mathsf{v},\mathsf{x}) \wedge \forall w\,(\mathsf{r}(\mathsf{v},w) \rightarrow \forall y\,(\mathsf{e}(w,y) \rightarrow (\mathsf{p}(w,y) \rightarrow w = \mathsf{v} \wedge y = \mathsf{x}))) \leftrightarrow$ *last_result*.

The following definition now renders **D2** as definition of the predicate ess in terms of *val_ess*.

Macro 18 $\mathit{def}_2(V, P)$ is defined as

$$\begin{array}{ll} \mathsf{world}(V) & \rightarrow \\ \forall X\,(\mathsf{ess}(V,P',X) \leftrightarrow \forall Q\, \mathit{val\_ess}(V,P,X,Q)), & \end{array}$$

where $P' := \dot{P}$.

In [31] it is shown that two observations about *essence* mentioned as *NOTE* in Scott's version [25] of Gödel's proof can be derived in this modeling.

The following two macros render the right-to-left direction of axioms **A1** and **A4**. The original axioms involve a universally quantified predicate that appears only in argument role. In the macros, it is represented by the parameter P.

Macro 19 $\mathit{ax}_1^{\leftarrow}(V, P)$ is defined as

$$\mathsf{world}(V) \rightarrow (\neg\mathsf{pos}(V,P') \rightarrow \mathsf{pos}(V,N')),$$

where $N' := \dot{\neg} P$,
$P' := \dot{P}$.

Macro 20 $\mathit{ax}_4(V, P)$ is defined as

$$\mathsf{world}(V) \rightarrow (\mathsf{pos}(V,P') \rightarrow \forall W\,(\mathsf{r}(V,W) \rightarrow \mathsf{pos}(W,P'))),$$

where $P' := \dot{P}$.

Theorem **T2** is rendered by the following macro with ess unfolded, which permits expansion into a universal second-order formula without occurrence of a predicate in argument position.

Macro 21 $proto\_thm_2(V, X)$ is defined as

$$\mathsf{world}(V) \rightarrow (\mathsf{e}(V, X) \rightarrow (\mathsf{g}(V, X) \rightarrow \forall Q\, val\_ess(V, \mathsf{g}, X, Q))).$$

Macro 22 $pre\_proto\_thm_2(V, X, Q)$ is defined as

$$\begin{array}{ll} ax_1^{\leftarrow}(V, Q) & \wedge \\ \forall W\,(\mathsf{r}(V, W) \rightarrow \forall X\,(\mathsf{e}(W, X) \rightarrow def_1^{\rightarrow}(W, X, Q))) & \wedge \\ def_1^{\rightarrow\neg}(V, X, Q) & \wedge \\ ax_4(V, Q). & \end{array}$$

This formula is valid: $\forall q\, \exists \dot{q} \exists \dot{\neg} q\, pre\_proto\_thm_2(\mathsf{v}, \mathsf{x}, q) \rightarrow proto\_thm_2(\mathsf{v}, \mathsf{x})$.

In this implication on the left side the constants $\dot{q}$ and $\dot{\neg} q$, which represent predicates q and $\lambda vx.\neg q(v, x)$ in argument positions, are existentially quantified.

3.4 Necessarily, God Exists – Proving Theorem T3

The definiens of *necessary existence*, which is defined in Definition **D3**, is rendered here by the following macro *val_ne*, expressed in terms of *val_ess*, the representation of the definiens of *essence*, to avoid the occurrence of a predicate representative in argument position.

Macro 23 $val\_ne(V, X)$ is defined as

$$\begin{array}{ll} \forall P\,(\forall Q\, val\_ess(V, P, X, Q) & \rightarrow \\ \quad \forall W\,(\mathsf{r}(V, W) \rightarrow \exists Y\,(\mathsf{e}(W, Y) \wedge P(W, Y)))). & \end{array}$$

Eliminating the quantified predicates shows another view on necessary existence.
Input: $val\_ne(\mathsf{v}, \mathsf{x})$.
Result of elimination:

$$\forall y\,(\mathsf{r}(\mathsf{v}, y) \rightarrow y = \mathsf{v}) \wedge \forall y\,(\mathsf{r}(\mathsf{v}, y) \rightarrow \mathsf{e}(y, \mathsf{x})).$$

The elimination result can be brought into a more clear form.
This formula is valid: $\forall w\,(\mathsf{r}(\mathsf{v}, w) \rightarrow w = \mathsf{v} \wedge \mathsf{e}(w, \mathsf{x})) \leftrightarrow last\_result$.

In analogy to the definition of the predicate ess in Macro 18 we define the predicate ne in terms of *val_ne*.

Macro 24 $def_3(V, X)$ is defined as

$$\mathsf{world}(V) \rightarrow (\mathsf{e}(V, X) \rightarrow (\mathsf{ne}(V, X) \leftrightarrow val\_ne(V, X))).$$

The following formula renders a fragment of the definition of necessary existence on a "shallow" level, that is, in terms of just the predicates ess and ne, without referring to their definientia *val_ess* and *val_ne*.

Macro 25 $def_3^{\rightarrow}(V, X, P)$ is defined as

$$\begin{array}{l} \mathsf{world}(V) \rightarrow \\ (\mathsf{e}(V, X) \rightarrow \\ \ (\mathsf{ne}(V, X) \rightarrow \\ \ \ (\mathsf{ess}(V, P', X) \rightarrow \\ \ \ \ \forall W\,(\mathsf{r}(V, W) \rightarrow \exists Y\,(\mathsf{e}(W, Y) \wedge P(W, Y)))))), \end{array}$$

where $P' := \dot{P}$.

Correctness of $def_3^{\rightarrow}$ can be established by showing that it follows from the definitions of ess and ne.

This formula is valid: $def_2(\mathsf{v},\mathsf{p}) \wedge def_3(\mathsf{v},\mathsf{x}) \rightarrow def_3^{\rightarrow}(\mathsf{v},\mathsf{x},\mathsf{p})$.

The following macro renders **T2**, in contrast to Macro 21 now expressed in terms of the predicate ess instead of its definiens *val_ess*.

Macro 26 $thm_2(V,X)$ is defined as

$$\mathsf{world}(V) \rightarrow (\mathsf{e}(V,X) \rightarrow (\mathsf{g}(V,X) \rightarrow \mathsf{ess}(V,\dot{\mathsf{g}},X))).$$

Axiom **A5** ($\mathsf{pos}(\mathsf{ne})$) is represented as follows.

Macro 27 $ax_5(V)$ is defined as

$$\mathsf{world}(V) \rightarrow \mathsf{pos}(V,\dot{\mathsf{ne}}).$$

Scott's version [25] shows theorem **T3** via the lemma **L**, rendered as follows.

Macro 28 $lemma_2(V)$ is defined as

$$\begin{array}{ll} \mathsf{world}(V) & \rightarrow \\ (\exists X\,(\mathsf{e}(V,X) \wedge \mathsf{g}(V,X)) & \rightarrow \\ \forall W\,(\mathsf{r}(V,W) \rightarrow \exists Y\,(\mathsf{e}(W,Y) \wedge \mathsf{g}(W,Y)))). & \end{array}$$

Macro 29 $pre\_lemma_2(V,X)$ is defined as

$$\begin{array}{ll} ax_5(V) & \wedge \\ def_1^{\rightarrow}(V,X,\mathsf{ne}) & \wedge \\ def_3^{\rightarrow}(V,X,\mathsf{g}) & \wedge \\ thm_2(V,X). & \end{array}$$

This formula is valid: $\forall v\,(\forall x\, pre\_lemma_2(v,x) \rightarrow lemma_2(v))$.

The following formula states theorem **T3**, the overall result to show.

Macro 30 $thm_3(V)$ is defined as

$$\mathsf{world}(V) \rightarrow \forall W\,(\mathsf{r}(V,W) \rightarrow \exists Y\,(\mathsf{e}(W,Y) \wedge \mathsf{g}(W,Y))).$$

Macro 31 $pre\_thm_3(V)$ is defined as $r\_world_1 \wedge \forall v\, lemma_2(v) \wedge coro(V)$.

Macro 32 *euclidean* is defined as $\forall x \forall y \forall z\,(\mathsf{r}(x,y) \wedge \mathsf{r}(x,z) \rightarrow \mathsf{r}(z,y))$.

Macro 33 *symmetric* is defined as $\forall x \forall y\,(\mathsf{r}(x,y) \rightarrow \mathsf{r}(y,x))$.

This formula is valid: $symmetric \vee euclidean \rightarrow (pre\_thm_3(\mathsf{v}) \rightarrow thm_3(\mathsf{v}))$.

As observed in [7], **T3** can not be just proven in the modal logic **S5**, but also in **KB**, whose accessibility relation is just constrained to be symmetric. We have shown here in a single statement that the proof is possible for a Euclidean as well as a symmetric accessibility relation by presupposing the disjunction of both properties. Precondition pre_thm_3 includes *coro* instantiated with just the current world and $lemma_2$ with a universal quantifier upon the world parameter. In fact, as shown now, using $lemma_2$ there just instantiated with the current world would not be sufficient to derive thm_3.

This formula is not valid: $symmetric \vee euclidean \rightarrow (r\_world_1 \wedge lemma_2(\mathsf{v}) \wedge coro(\mathsf{v}) \rightarrow thm_3(\mathsf{v}))$.

In [31] further aspects of Gödel's proof are modeled, in particular modal collapse and monotheism.

4 On Weakening the Frame Condition for Theorem T3

In the proof of thm_3 from pre_thm_3 we used the additional frame condition $euclidean \vee symmetric$. The observation that the weaker **KB** instead of **S5** suffices to prove **T3** was an important finding of [7]. Hence, the question arises whether the precondition on the accessibility relation can be weakened further.

In general, the *weakest sufficient condition* [19,11,28] of a formula G on a set Q of predicates within a formula F can be expressed as the second-order formula $\forall p_1 \ldots \forall p_n\,(F \rightarrow G)$, where $p_1, \ldots, p_n$ are all predicates that occur free in $F \rightarrow G$ and are not members of Q. This formula denotes the weakest (with respect to entailment) formula H in which only predicates in Q occur free such that $H \rightarrow (F \rightarrow G)$ is valid. Second-order quantifier elimination can be applied to this formula to "compute" a weakest sufficient condition, that is, converting it to a first-order formula, which, of course, is inherently not possible in all cases.

For **T3**, the weakest sufficient condition on the accessibility relation r and the domain membership relation and e is the second-order formula

$$\forall g \forall v\,(pre\_thm_3(v) \rightarrow thm_3(v)).$$

Unfortunately, elimination of the second-order quantifier upon g fails for this formula (at least with the current version of *PIE*). But elimination succeeds for a simplified variant of the problem, which considers just propositional modal logic and combines two instances of Lemma $lemma_2$ with an unfolding of **C**. The way in which this simplification was obtained is outlined in [31].

Macro 34 $lemma_2\_simp(V)$ is defined as

$$\mathsf{g}(V) \rightarrow \forall W\,(\mathsf{r}(V, W) \rightarrow \mathsf{g}(W)).$$

Macro 35 $pre\_thm_3\_simp\_inst(V)$ is defined as

$$lemma_2\_simp(V) \quad \wedge$$
$$\exists W\,(\mathsf{r}(V, W) \wedge \mathsf{g}(W) \wedge lemma_2\_simp(W)).$$

Macro 36 $thm_3\_simp(V)$ is defined as

$$\forall W\,(\mathsf{r}(V, W) \rightarrow \mathsf{g}(W)).$$

This formula is valid: $euclidean \vee symmetric \rightarrow (pre\_thm_3\_simp\_inst(\mathsf{v}) \rightarrow thm_3\_simp(\mathsf{v}))$.

Input: $\forall g \forall v\,(pre\_thm_3\_simp\_inst(v) \rightarrow thm_3\_simp(v))$.
Result of elimination:

$$\forall x \forall y \forall z\,(\mathsf{r}(x, y) \wedge \mathsf{r}(x, z) \rightarrow \mathsf{r}(y, x) \vee \mathsf{r}(y, z) \vee x = y \vee y = z).$$

We write the resulting first-order formula in a slightly different form, give it a name, verify equivalence to the original form and show some of its properties.

Macro 37 *frame_cond_simp* is defined as

$$\forall x \forall y \forall z\,(\mathsf{r}(x, y) \wedge \mathsf{r}(x, z) \wedge y \neq x \wedge y \neq z \rightarrow \mathsf{r}(y, x) \vee \mathsf{r}(y, z)).$$

This formula is valid: $frame\_cond\_simp \leftrightarrow last\_result$.

Macro 38 *reflexive* is defined as $\forall x\, \mathsf{r}(x, x)$.

This formula is valid: $\mathit{reflexive} \rightarrow (\mathit{symmetric} \vee \mathit{euclidean} \leftrightarrow \mathit{frame\_cond\_simp})$.
This formula is valid: $\mathit{symmetric} \vee \mathit{euclidean} \rightarrow \mathit{frame\_cond\_simp}$.
This formula is not valid: $\mathit{frame\_cond\_simp} \rightarrow \mathit{symmetric} \vee \mathit{euclidean}$.

Thus, the obtained frame condition *frame_cond_simp* is under the assumption of reflexivity equivalent to *symmetric* $\vee$ *euclidean*, and without that assumption strictly weaker. The following statement shows that this weaker frame condition also works for our original problem, proving **T3**.

This formula is valid: $\mathit{frame\_cond\_simp} \rightarrow (\mathit{pre\_thm}_3(\mathsf{v}) \rightarrow \mathit{thm}_3(\mathsf{v}))$.

Hence, via the detour through elimination applied to a simplified problem, we have found a strictly weaker frame condition for **T3** than *symmetric* $\vee$ *euclidean*, but, since elimination has just been performed on the second-order formula representing the simplified problem, we do not know whether it is *the weakest* one.

5 Conclusion

We reconstructed Gödel's ontological proof in an environment that integrates automated first-order theorem proving, second-order quantifier elimination, a formula macro mechanism and LaTeX-based formula pretty-printing, supplementing a number of previous works that render Gödel's proof in other automated theorem proving environments. Particular observations of interest for the study of Gödel's proof that became apparent through our modeling include the following:

1. The presentation of the derivation of theorem **T1** exhibits the few actually used instantiations of axioms **A1** and **A2**. The derivation is via a lemma, which might be useful as internal interface in the proof because it can be justified in alternate ways.
2. Corollary **C** can be shown independently from the actual definition of *God-like* (**D1**) just on the basis of the assumption that **T1** applies to *God-like*.[4]
3. In the whole proof, definition **D1** is only used in the left-to-right direction.[5]
4. Second-order quantifier elimination yields first-order representations of *essence* (definition **D2**) and *necessary existence* (definition **D3**).
5. Lemma **L** can be derived independently from the definiens of *essence*. Here the predicate ess appears in the respective expanded formula passed to the reasoner, but not its definiens.
6. For the derivation of theorem **T3** an accessibility relationship is sufficient that, unless reflexivity is assumed, is strictly weaker than the disjunction of the Euclidean property and symmetry.

If non-experts in automated reasoning are addressed, the syntactical presentation of Gödel's argument is of particular importance [8]. We approached this

[4] This is also apparent in [5, Fig. 4, line 20].
[5] This applies if **A3** is given as in Scott's version, but not if it is derived from further general properties of *positive*, as in Gödel's original version and in [5, Fig. 4, line 19].

requirement by means of formula macro definitions with the representation of input formulas by Prolog terms and LaTeX pretty-printing for output formulas.

Most automated formalizations of metaphysical arguments, e.g., [12,24,7,8,5], seem closely tied to a particular system or combination of systems. Of course, processing a *PIE* document similarly depends on the *PIE* system with specific embedded reasoners. However, a system-independent view on the formalization is at least obtainable: The underlying target logic of the macro expansion is just the well-known classical first-order logic extended with predicate quantification. Reasoning tasks are only performed on the expanded formulas. The *PIE* system can output these explicitly (see, e.g., [31]), providing a low-level, but system-independent logical representation of the complete formalization. As a further beneficial aspect, such an explicit low-level formalization may prevent the unnoticed interaction with features of involved special logics.

A limitation of our approach might be that there is no automated support for the passage from the low to the high level, i.e., folding into formula macros. If trust in proofs is an issue, steps in the overall workflow for which no proof representations are produced may be objectionable. This concerns macro expansion, formula normalization (see, however, [21]), pre- and postprocessing of formulas, and in particular second-order quantifier elimination, for which the creation of proofs seems an unexplored terrain. A practical makeshift is comparison with the few other elimination systems [2, Sect. 4].

In principle it should be possible to integrate second-order quantifier elimination as used here also into automated reasoning environments based on other paradigms, in particular the heterogeneous environments that involve forms of higher-order reasoning and are applied in [7,8,6,5].

Concerning second-order quantifier elimination, an issue that might be worth further investigation is the generalization of the method applied here ad-hoc to weaken the precondition on the accessibility relation: We started from an elimination problem that could not be solved (at least with the current implementation of *PIE*), moved to a simpler problem and then verified that the solution of the simpler problem is also applicable to the original problem, where it does not represent the originally desired unique *weakest* sufficient condition, but nevertheless a condition that is weaker than the condition known before.

Acknowledgments. The author thanks Christoph Benzmüller and anonymous reviewers for helpful remarks. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 457292495.

References

1. Ackermann, W.: Untersuchungen über das Eliminationsproblem der mathematischen Logik. Math. Annn. **110**, 390–413 (1935). https://doi.org/10.1007/BF01448035
2. Alassaf, R., Schmidt, R.A.: DLS-Forgetter: An implementation of the DLS forgetting calculus for first-order logic. In: Calvanese, D., Iocchi, L. (eds.) GCAI

2019. EPiC Series in Computing, vol. 65, pp. 127–138. EasyChair (2019). https://doi.org/10.29007/hvz6
3. van Benthem, J.: Modal Logic and Classical Logic. Bibliopolis (1983)
4. van Benthem, J.: Modal Logic for Open Minds. CSLI Publications (2010)
5. Benzmüller, C.: A (simplified) supreme being necessarily exists, says the computer: Computationally explored variants of Gödel's ontological argument. In: Calvanese, D., Erdem, E., Thielscher, M. (eds.) KR 2020. pp. 779–789. IJCAI organization (2020). https://doi.org/10.24963/kr.2020/80
6. Benzmüller, C., Weber, L., Woltzenlogel Paleo, B.: Computer-assisted analysis of the Anderson-Hájek ontological controversy. Logica Universalis **11**(1), 139–151 (2017). https://doi.org/10.1007/s11787-017-0160-9
7. Benzmüller, C., Woltzenlogel Paleo, B.: Automating Gödel's ontological proof of god's existence with higher-order automated theorem provers. In: Schaub, T., Friedrich, G., O'Sullivan, B. (eds.) ECAI 2014. FAIA, vol. 263, pp. 93–98. IOS Press (2014). https://doi.org/10.3233/978-1-61499-419-0-93
8. 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. pp. 936–942. AAAI Press (2016), https://www.ijcai.org/Proceedings/16/Papers/137.pdf
9. Dahn, I., Wernhard, C.: First order proof problems extracted from an article in the Mizar mathematical library. In: FTP'97. pp. 58–62. RISC-Linz Report Series No. 97–50, Joh. Kepler Univ., Linz (1997), https://www.logic.at/ftp97/papers/dahn.pdf
10. Doherty, P., Łukaszewicz, W., Szałas, A.: Computing circumscription revisited: A reduction algorithm. JAR **18**(3), 297–338 (1997). https://doi.org/10.1023/A:1005722130532
11. Doherty, P., Łukaszewicz, W., Szałas, A.: Computing strongest necessary and weakest sufficient conditions of first-order formulas. In: Nebel, B. (ed.) IJCAI-01. pp. 145–151. Morgan Kaufmann (2001), https://www.ijcai.org/Proceedings/01/IJCAI-2001-b.pdf#page=133
12. Fitelson, B., Zalta, E.N.: Steps toward a computational metaphysics. J. Philos. Log. **36**(2), 227–247 (2007). https://doi.org/10.1007/s10992-006-9038-7
13. Fitting, M.: Types, Tableaus, and Gödel's God. Springer (2002). https://doi.org/10.1007/978-94-010-0411-4
14. Gödel, K.: Ontologischer Beweis – notes in Kurt Gödel's hand (1970), transcriptions published in [27, pp. 144–145] and also in [26, pp. 256–257]
15. Kanckos, A., Lethen, T.: The development of Gödel's ontological proof. Review of Symbolic Logic pp. 1–19 (2019). https://doi.org/10.1017/s1755020319000479
16. Kanckos, A., Woltzenlogel Paleo, B.: Variants of Gödel's ontological proof in a natural deduction calculus. Studia Logica **105**, 553–586 (2017). https://doi.org/10.1007/s11225-016-9700-1
17. Kirchner, D., Benzmüller, C., Zalta, E.N.: Computer science and metaphysics: A cross-fertilization. Open Philosophy **2**, 230–251 (2019). https://doi.org/10.1515/opphil-2019-0015
18. Knuth, D.E.: Literate programming. Comput. J. **27**(2), 97–111 (1984)
19. Lin, F.: On strongest necessary and weakest sufficient conditions. Artificial Intelligence **128**, 143–159 (2001). https://doi.org/10.1016/S0004-3702(01)00070-4
20. McCune, W.: Prover9 and Mace4 (2005–2010), http://www.cs.unm.edu/~mccune/prover9

21. de Nivelle, H.: Extraction of proofs from the clausal normal form transformation. In: Bradfield, J. (ed.) CSL 2002. LNCS, vol. 2471, pp. 584–598. Springer (2002). https://doi.org/10.1007/3-540-45793-3_39
22. Ohlbach, H.J.: SCAN – elimination of predicate quantifiers: System description. In: McRobbie, M.A., Slaney, J.K. (eds.) CADE-13. LNCS (LNAI), vol. 1104, pp. 161–165. Springer (1996). https://doi.org/10.1007/3-540-61511-3_77
23. Ohlbach, H.J., Engel, T., Schmidt, R.A., Gabbay, D.M.: SCAN, http://www.mettel-prover.org/scan/index.html
24. Oppenheimer, P.E., Zalta, E.N.: A computationally-discovered simplification of the ontological argument. Australasian J. Philos. **89**(2), 333–349 (2011). https://doi.org/10.1080/00048401003674482
25. Scott, D.: Gödel's ontological proof – notes in Dana Scott's hand (1970), transcriptions published in [27, pp. 145–146] and also in [26, pp. 257–258]
26. Sobel, J.H.: Gödel's ontological proof. In: Thomson, J.J. (ed.) On Being and Saying: Essays for Richard Cartwright. MIT Press, Cambridge, MA (1987)
27. Sobel, J.H.: Logic and Theism: Arguments For and Against Beliefs in God. Cambridge University Press, Cambridge (2004). https://doi.org/10.1017/CBO9780511497988
28. Wernhard, C.: Projection and scope-determined circumscription. Journal of Symbolic Computation **47**, 1089–1108 (2012). https://doi.org/10.1016/j.jsc.2011.12.034
29. Wernhard, C.: The PIE system for proving, interpolating and eliminating. In: Fontaine, P., Schulz, S., Urban, J. (eds.) PAAR 2016. CEUR Workshop Proceedings, vol. 1635, pp. 125–138. CEUR-WS.org (2016), http://ceur-ws.org/Vol-1635/paper-11.pdf
30. Wernhard, C.: Facets of the PIE environment for proving, interpolating and eliminating on the basis of first-order logic. In: Hofstedt, P., et al. (eds.) DECLARE 2019, Revised Selected Papers. LNCS (LNAI), vol. 12057, pp. 160–177. Springer (2020). https://doi.org/10.1007/978-3-030-46714-2_11
31. Wernhard, C.: Applying second-order quantifier elimination in inspecting Gödel's ontological proof (extended version). Tech. rep. (2021), https://arxiv.org/abs/2110.11108