Abstract
A computer-formalisation of the essential parts of Fitting’s textbook *Types, Tableaus and Gödel’s God* in Isabelle/HOL is presented. In particular, Fitting’s (and Anderson’s) variant of the ontological argument is verified and confirmed. This variant avoids the modal collapse, which has been criticised as an undesirable side-effect of Kurt Gödel’s (and Dana Scott’s) versions of the ontological argument. Fitting’s work is employing an intensional higher-order modal logic, which we shallowly embed here in classical higher-order logic. We then utilize the embedded logic for the formalisation of Fitting’s argument.

Contents

1 Introduction

2 Embedding of Intensional Higher-Order Modal Logic
   2.1 Type Declarations
   2.2 Definitions
       2.2.1 Logical Operators as Truth-Sets
       2.2.2 Possibilist Quantification
       2.2.3 Actualist Quantification
       2.2.4 Modal Operators
       2.2.5 Extension-of Operator
       2.2.6 Equality
       2.2.7 Meta-logical Predicates
   2.3 Verifying the Embedding
   2.4 Useful Definitions for Axiomatization of Further Logics

August 16, 2018
1 Introduction

We present a study on Computational Metaphysics: a computer-formalisation and verification of Fitting’s variant of the ontological argument (for the existence of God) as presented in his textbook *Types, Tableaus and Gödel’s God* [12]. Fitting’s argument is an emendation of Kurt Gödel’s modern variant [15] (resp. Dana Scott’s variant [17]) of the ontological argument.

The motivation is to avoid the modal collapse [18, 19], which has been criticised as an undesirable side-effect of the axioms of Gödel resp. Scott. The modal collapse essentially states that there are no contingent truths and that everything is determined. Several authors (e.g. [2, 1, 16, 10]) have proposed emendations of the argument with the aim of maintaining the essential result (the necessary existence of God) while at the same time avoiding the modal collapse. Related work has formalised several of these variants on the computer and verified or falsified them. For example, Gödel’s axioms [15] have been shown inconsistent [8, 9] while Scott’s version has been verified [5]. Further experiments, contributing amongst others to the clarification of a related debate between Hájek and Anderson, are presented and discussed in [6]. The enabling technique in all of these experiments has been shallow semantical embeddings of (extensional) higher-order modal logics in classical higher-order logic (see [6, 3] and the references therein).

Fitting’s emendation also intends to avoid the modal collapse. However, in contrast to the above variants, Fitting’s solution is based on the use of an intensional as opposed to an extensional higher-order modal logic. For our work this imposed the additional challenge to provide a shallow embedding of this more advanced logic. The experiments presented below confirm that Fitting’s argument as presented in his textbook [12] is valid and that it avoids the modal collapse as intended.

The work presented here originates from the *Computational Metaphysics* lecture course held at FU Berlin in Summer 2016 [7].
2 Embedding of Intensional Higher-Order Modal Logic

The object logic being embedded, intensional higher-order modal logic (IHOML), is a modification of the intentional logic developed by Montague and Gallin [14]. IHOML is introduced by Fitting in the second part of his textbook [12] in order to formalise his emendation of Gödel’s ontological argument. We offer here a shallow embedding of this logic in Isabelle/HOL, which has been inspired by previous work on the semantical embedding of multimodal logics with quantification [6]. We expand this approach to allow for actualist quantifiers, intensional types and their related operations.

2.1 Type Declarations

Since IHOML and Isabelle/HOL are both typed languages, we introduce a type-mapping between them. We follow as closely as possible the syntax given by Fitting (see p. 86). According to this syntax, if \( \tau \) is an extensional type, \( \uparrow \tau \) is the corresponding intensional type. For instance, a set of (red) objects has the extensional type \( \langle \! \langle 0 \rangle \! \rangle \), whereas the concept ‘red’ has intensional type \( \uparrow \langle \! \langle 0 \rangle \! \rangle \). In what follows, terms having extensional (intensional) types will be called extensional (intensional) terms.

**type-decl** i — type for possible worlds
**type-synonym** io = \((i \Rightarrow \text{bool})\) — formulas with world-dependent truth-value
**type-decl** e \((0)\) — individuals

Aliases for common unary predicate types:

**type-synonym** ie = \((i \Rightarrow \langle \! \langle 0 \rangle \! \rangle)\)
**type-synonym** ise = \((i \Rightarrow \langle \! \langle \uparrow \langle \! \langle 0 \rangle \! \rangle \rangle \rangle)\)
**type-synonym** isie = \((\uparrow \langle \! \langle 0 \rangle \! \rangle \Rightarrow \langle \! \langle \uparrow \langle \! \langle 0 \rangle \! \rangle \rangle \rangle)\)
**type-synonym** sisie = \((\uparrow \langle \! \langle \uparrow \langle \! \langle 0 \rangle \! \rangle \rangle \rangle \Rightarrow \langle \! \langle \uparrow \langle \! \langle \uparrow \langle \! \langle 0 \rangle \! \rangle \rangle \rangle \rangle \rangle)\)

**type-synonym** sse = \(\langle \! \langle \! \langle 0 \rangle \! \rangle \! \rangle \Rightarrow \langle \! \langle \! \langle \! \langle 0 \rangle \! \rangle \! \rangle \! \rangle \rangle\)
**type-synonym** issese = \((\langle \! \langle \! \langle 0 \rangle \! \rangle \! \rangle \Rightarrow \langle \! \langle \! \langle \! \langle \! \langle 0 \rangle \! \rangle \! \rangle \! \rangle \! \rangle)\)

Aliases for common binary relation types:

**type-synonym** see = \((0 \Rightarrow 0 \Rightarrow \text{bool})\)
**type-synonym** ise = \((0 \Rightarrow 0 \Rightarrow \langle \! \langle 0 \rangle \! \rangle)\)
**type-synonym** sieie = \((\uparrow 0 \Rightarrow \uparrow 0 \Rightarrow \text{bool})\)
**type-synonym** isieie = \((\uparrow 0 \Rightarrow \uparrow 0 \Rightarrow \langle \! \langle 0 \rangle \! \rangle)\)
**type-synonym** ssese = \((\langle \! \langle 0 \rangle \! \rangle \Rightarrow \langle \! \langle 0 \rangle \! \rangle \Rightarrow \text{bool})\)
**type-synonym** issese = \((\langle \! \langle 0 \rangle \! \rangle \Rightarrow \langle \! \langle 0 \rangle \! \rangle \Rightarrow \langle \! \langle 0 \rangle \! \rangle)\)
**type-synonym** isisser = \((\langle \! \langle 0 \rangle \! \rangle \Rightarrow \langle \! \langle 0 \rangle \! \rangle \Rightarrow \langle \! \langle 0 \rangle \! \rangle)\)
2.2 Definitions

2.2.1 Logical Operators as Truth-Sets

abbreviation mnot :: io⇒io (¬-[52,53])
   where ¬ϕ ≡ λx. ¬(ϕ x)
abbreviation mnegpred :: (0)⇒(0) (¬-[52,53])
   where ¬ϕ ≡ λx. λw. ¬(ϕ x w)
abbreviation manganese :: ℓ⇒io⇒io (infixr∧51)
   where ϕ∧ψ ≡ λx. (ϕ x)∧(ψ x)
abbreviation mor :: io⇒io⇒io (infixr∨50)
   where ϕ∨ψ ≡ λx. (ϕ x)∨(ψ x)
abbreviation mimp :: io⇒io⇒io (infixr→49)
   where ϕ→ψ ≡ λx. (ϕ x)→(ψ x)
abbreviation mequ :: io⇒io⇒io (infixr↔48)
   where ϕ↔ψ ≡ λx. (ϕ x)↔(ψ x)
abbreviation xor :: bool⇒bool⇒bool (infixr⊕50)
   where ϕ⊕ψ ≡ ((ϕψ) ∧ ¬(ϕ∧ψ))
abbreviation mxor :: io⇒io⇒io (infixr⊕50)
   where ϕ⊕ψ ≡ λx. (ϕ x)⊕(ψ x)

2.2.2 Possibilist Quantification

abbreviation mforall :: 't⇒io⇒io (∀)
   where ∀ ϕ ≡ λx. (ϕ x)
abbreviation mxexists :: 't⇒io⇒io (∃)
   where ∃ ϕ ≡ λx. (ϕ x)
abbreviation mforallB :: 't⇒io⇒io (binder∀[8]9)  — Binder notation
   where ∀ x. ϕ x ≡ ∀ ϕ
abbreviation mxexistsB :: 't⇒io⇒io (binder∃[8]9)
   where ∃ x. ϕ x ≡ ∃ ϕ

2.2.3 Actualist Quantification

The following predicate is used to model actualist quantifiers by restricting
the domain of quantification at every possible world. This standard tech-
nique has been referred to as existence relativization ([13], p. 106), high-
lighting the fact that this predicate can be seen as a kind of meta-logical
‘existence predicate’ telling us which individuals actually exist at a given
world. This meta-logical concept does not appear in our object language.
consts Exists::↑⟨0⟩ (existsAt)

abbreviation mforallAct :: ↑⟨↑⟨0⟩⟩ (∀E)
where ∀E Φ ≡ λ w. ∀x. (existsAt x w) ⟷ (Φ x w)

abbreviation mexistsAct :: ↑⟨↑⟨0⟩⟩ (∃E)
where ∃E Φ ≡ λ w. ∃x. (existsAt x w) ∧ (Φ x w)

abbreviation mforallActB :: ↑⟨↑⟨0⟩⟩ (binder∀E[8][9]) — binder notation
where ∀E x. ϕ(x) ≡ ∀E ϕ

abbreviation mexistsActB :: ↑⟨↑⟨0⟩⟩ (binder∃E[8][9])
where ∃E x. ϕ(x) ≡ ∃E ϕ

2.2.4 Modal Operators

consts aRel::i⇒i⇒bool (infixr r 70) — accessibility relation r

abbreviation mbox :: io⇒io (□-52)[53]
where □ϕ ≡ λ w. ∀v. (w r v) ⟷ (ϕ v)

abbreviation mdia :: io⇒io (♦-52)[53]
where ♦ϕ ≡ λ w. ∃v. (w r v) ∧ (ϕ v)

2.2.5 Extension-of Operator

According to Fitting’s semantics ([12], pp. 92-4) ↓ is an unary operator applying only to intensional terms. A term of the form ↓α designates the extension of the intensional object designated by α, at some given world. For instance, suppose we take possible worlds as persons, we can therefore think of the concept ‘red’ as a function that maps each person to the set of objects that person classifies as red (its extension). We can further state, the intensional term r of type ↑⟨0⟩ designates the concept ‘red’. As can be seen, intensional terms in ITHOML designate functions on possible worlds and they always do it rigidly. We will sometimes refer to an intensional object explicitly as ‘rigid’, implying that its (rigidly) designated function has the same extension in all possible worlds.

Terms of the form ↓α are called relativized (extensional) terms; they are always derived from intensional terms and their type is extensional (in the color example ↓r would be of type ⟨0⟩). Relativized terms may vary their denotation from world to world of a model, because the extension of an intensional term can change from world to world, i.e. they are non-rigid.

To recap: an intensional term denotes the same function in all worlds (i.e. it’s rigid), whereas a relativized term denotes a (possibly) different extension (an object or a set) at every world (i.e. it’s non-rigid). To find out the denotation of a relativized term, a world must be given. Relativized terms are the only non-rigid terms.
For our Isabelle/HOL embedding, we had to follow a slightly different approach; we model ↓ as a predicate applying to formulas of the form \( \Phi(\downarrow\alpha_1 \ldots \alpha_n) \) (for our treatment we only need to consider cases involving one or two arguments, the first one being a relativized term). For instance, the formula

\[ Q(\downarrow a_1)^w \]

(evaluated at world \( w \)) is modelled as \( [(Q,a_1)^w \) (or \( (Q \downarrow a_1)^w \) using infix notation), which gets further translated into \( Q(a_1(w))^w \).

Depending on the particular types involved, we have to define ↓ differently to ensure type correctness (see a-d below). Nevertheless, the essence of the Extension-of operator remains the same: a term \( \alpha \) preceded by ↓ behaves as a non-rigid term, whose denotation at a given possible world corresponds to the extension of the original intensional term \( \alpha \) at that world.

(a) Predicate \( \varphi \) takes as argument a relativized term derived from an (intensional) individual of type \( \uparrow \mathbf{0} \):

**abbreviation** \text{extIndivArg}::\( \uparrow (0) \Rightarrow \uparrow \mathbf{0} \Rightarrow \mathbf{io} \) (infix \( \mid 60 \))

where \( \varphi | c \equiv \lambda w. \varphi (c \ w) \)

(b) A variant of (a) for terms derived from predicates (types of form \( \uparrow(t) \)):

**abbreviation** \text{extPredArg}::\( (\uparrow t \Rightarrow \mathbf{bool}) \Rightarrow \mathbf{io} \Rightarrow (\uparrow t \Rightarrow \mathbf{io}) \Rightarrow \mathbf{io} \) (infix \( \downarrow 60 \))

where \( \varphi \downarrow P \equiv \lambda w. \varphi (\lambda x. P \ x \ w) \ w \)

(c) A variant of (b) with a second argument (the first one being relativized):

**abbreviation** \text{extPredArg1}::\( (\uparrow t \Rightarrow \mathbf{bool}) \Rightarrow \mathbf{b} \Rightarrow \mathbf{io} \Rightarrow (\uparrow t \Rightarrow \mathbf{io}) \Rightarrow \mathbf{b} \Rightarrow \mathbf{io} \) (infix \( \downarrow_1 60 \))

where \( \varphi \downarrow_1 P \equiv \lambda z. \lambda w. \varphi (\lambda x. P \ x \ w) \ z \ w \)

In what follows, the ‘[ - ]’ parentheses are an operator used to convert extensional objects into ‘rigid’ intensional ones:

**abbreviation** \text{trivialConversion}::\( \mathbf{bool} \Rightarrow \mathbf{io} \) (infix \( \) \) where \( \{ \varphi \} \equiv (\lambda w. \varphi) \)

(d) A variant of (b) where \( \varphi \) takes ‘rigid’ intensional terms as argument:

**abbreviation** \text{mextPredArg}::\( (\uparrow \mathbf{io}) \Rightarrow \mathbf{io} \Rightarrow (\uparrow \mathbf{io}) \Rightarrow \mathbf{io} \) (infix \( \downarrow 60 \))

where \( \varphi \downarrow P \equiv \lambda w. \varphi (\lambda x. \{P \ w\}) \ w \)

2.2.6 Equality

**abbreviation** \text{meq} :: \( t \Rightarrow t \Rightarrow \mathbf{io} \) (infix\( \approx 60 \)) — normal equality (for all types)

where \( x \approx y \equiv \lambda w. x = y \)

**abbreviation** \text{meqC} :: \( \uparrow(\uparrow 0,\uparrow 0) \) (infix\( \approx C 52 \)) — eq. for individual concepts

where \( x \approx^C y \equiv \lambda v. \forall v. (x \ v) = (y \ v) \)

**abbreviation** \text{meqL} :: \( \uparrow(0,0) \) (infix\( \approx L 52 \)) — Leibniz eq. for individuals

where \( x \approx^L y \equiv \forall \varphi. \varphi(x) \Rightarrow \varphi(y) \)

2.2.7 Meta-logical Predicates

**abbreviation** \text{valid} :: \( \mathbf{io} \Rightarrow \mathbf{bool} \) (infix\( [8] \)) where \( [\varphi] \equiv \forall w.(\varphi \ w) \)

**abbreviation** \text{satisfiable} :: \( \mathbf{io} \Rightarrow \mathbf{bool} \) (infix\( \sat 8 \)) where \( [\varphi]^\sat \equiv \exists w.(\varphi \ w) \)

**abbreviation** \text{countersat} :: \( \mathbf{io} \Rightarrow \mathbf{bool} \) (infix\( \csat 8 \)) where \( [\varphi]^\csat \equiv \exists w.\neg(\varphi \ w) \)

**abbreviation** \text{invalid} :: \( \mathbf{io} \Rightarrow \mathbf{bool} \) (infix\( \inv 8 \)) where \( [\varphi]^\inv \equiv \forall w.\neg(\varphi \ w) \)
2.3 Verifying the Embedding

The above definitions introduce modal logic $K$ with possibilist and actualist quantifiers, as evidenced by the following tests:

Verifying $K$ Principle and Necessitation:

- **Lemma**: $K$: \[\lfloor (\Box (\varphi \rightarrow \psi)) \rightarrow (\Box \varphi \rightarrow \Box \psi) \rfloor \text{ by simp} \quad - K \text{ schema} \]
- **Lemma**: NEC: $\lfloor \varphi \rfloor \rightarrow \lfloor [\Box \varphi] \rfloor \text{ by simp} \quad - \text{necessitation} \]

Local consequence implies global consequence (we will use this lemma often):

- **Lemma**: localImpGlobalCons: $\lfloor \varphi \rightarrow \xi \rfloor \rightarrow \lfloor \varphi \rfloor \rightarrow \lfloor \xi \rfloor \text{ by simp} \]

But global consequence does not imply local consequence:

- **Lemma**: $\lfloor [\varphi] \rightarrow [\xi] \rfloor \rightarrow [\varphi \rightarrow \xi] \text{ nitpick oops} - \text{countersatisfiable} \]

Barcan and Converse Barcan Formulas are satisfied for standard (possibilist) quantifiers:

- **Lemma**: $\lfloor (\forall x. (\Box x \cdot (\varphi x))) \rightarrow (\Box (\forall x. (\varphi x))) \rfloor \text{ by simp} \]
- **Lemma**: $\lfloor (\Box (\forall x. (\varphi x))) \rightarrow (\forall x. (\Box (\varphi x))) \rfloor \text{ by simp} \]

(Converse) Barcan Formulas not satisfied for actualist quantifiers:

- **Lemma**: $\lfloor (\forall x. (\Box (\varphi x))) \rightarrow (\Box (\forall x. (\varphi x))) \rfloor \text{ nitpick oops} - \text{countersatisfiable} \]
- **Lemma**: $\lfloor (\forall x. (\Box (\varphi x))) \rightarrow (\forall x. (\Box (\varphi x))) \rfloor \text{ nitpick oops} - \text{countersatisfiable} \]

Above we have made use of (counter-)model finder Nitpick [11] for the first time. For all the conjectured lemmas above, Nitpick has found a counter-model, i.e. a model satisfying all the axioms which falsifies the given formula. This means, the formulas are not valid.

Well known relations between meta-logical notions:

- **Lemma**: $\lfloor \varphi \rfloor \leftrightarrow \neg [\varphi]^{csat} \text{ by simp} \]
- **Lemma**: $[\varphi]^{sat} \leftrightarrow \neg [\varphi]^{inv} \text{ by simp} \]

Contingent truth does not allow for necessitation:

- **Lemma**: $\lfloor \Diamond \varphi \rfloor \rightarrow [\Box \varphi] \text{ nitpick oops} - \text{countersatisfiable} \]
- **Lemma**: $[\Box \varphi]^{sat} \rightarrow [\Box \varphi] \text{ nitpick oops} - \text{countersatisfiable} \]

*Modal collapse* is countersatisfiable:

- **Lemma**: $\lfloor \varphi \rightarrow \Box \varphi \rfloor \text{ nitpick oops} - \text{countersatisfiable} \]

8
2.4 Useful Definitions for Axiomatization of Further Logics

The best known normal logics (K4, K5, KB, K45, KB5, D, D4, D5, D45, ... ) can be obtained by combinations of the following axioms:

- **abbreviation M**
  
  where \( M \equiv \forall \varphi. \Box \varphi \to \varphi \)

- **abbreviation B**
  
  where \( B \equiv \forall \varphi. \varphi \to \Box \Diamond \varphi \)

- **abbreviation D**
  
  where \( D \equiv \forall \varphi. \Box \varphi \to \Diamond \varphi \)

- **abbreviation IV**
  
  where \( IV \equiv \forall \varphi. \Box \varphi \to \Box \Box \varphi \)

- **abbreviation V**
  
  where \( V \equiv \forall \varphi. \Diamond \varphi \to \Box \Diamond \varphi \)

Instead of postulating (combinations of) the above axioms we instead make use of the well-known Sahlqvist correspondence, which links axioms to constraints on a model’s accessibility relation (e.g. reflexive, symmetric, etc.; the definitions of which are not shown here). We show that reflexivity, symmetry, seriality, transitivity and euclideanness imply axioms M, B, D, IV, V respectively.

- **lemma reflexive aRel** \( \Rightarrow [M] \) by blast — aka T
- **lemma symmetric aRel** \( \Rightarrow [B] \) by blast
- **lemma serial aRel** \( \Rightarrow [D] \) by blast
- **lemma transitive aRel** \( \Rightarrow [IV] \) by blast
- **lemma euclidean aRel** \( \Rightarrow [V] \) by blast

- **lemma preorder aRel** \( \Rightarrow [M] \wedge [IV] \) by blast — S4: reflexive + transitive
- **lemma equivalence aRel** \( \Rightarrow [M] \wedge [V] \) by blast — S5: preorder + symmetric
- **lemma reflexive aRel \& euclidean aRel** \( \Rightarrow [M] \wedge [V] \) by blast — S5

Using these definitions, we can derive axioms for the most common modal logics (see also [4]). Thereby we are free to use either the semantic constraints or the related Sahlqvist axioms. Here we provide both versions. In what follows we use the semantic constraints (for improved performance).
3 Textbook Examples

In this section we provide further evidence that our embedded logic works as intended by proving the examples discussed in the book. In many cases, we consider further theorems which we derived from the original ones. We were able to confirm that all results (proofs or counterexamples) agree with Fitting’s claims.

3.1 Modal Logic - Syntax and Semantics (Chapter 7)

Reminder: We call a term relativized if it is of the form $\downarrow\alpha$ (i.e. an intensional term preceded by the extension-of operator), otherwise it is non-relativized. Relativized terms are non-rigid and non-relativized terms are rigid.

3.1.1 Considerations Regarding $\beta\eta$-redex (p. 94)

$\beta\eta$-redex is valid for non-relativized (intensional or extensional) terms:

- **Lemma**: $[\lambda\alpha. \varphi \alpha \ (\varphi \tau)] \leftrightarrow (\varphi \tau)$ by simp
- **Lemma**: $[\lambda\alpha. \varphi \alpha \ (\tau::0)] \leftrightarrow (\varphi \tau)$ by simp
- **Lemma**: $[\lambda\alpha. \Box\varphi \alpha \ (\tau::0)] \leftrightarrow (\Box\varphi \tau)]$ by simp
- **Lemma**: $[\lambda\alpha. \Box\varphi \alpha \ (\tau::0)] \leftrightarrow (\Box\varphi \tau)]$ by simp

$\beta\eta$-redex is valid for relativized terms as long as no modal operators occur inside the predicate abstract:

- **Lemma**: $[\lambda\alpha. \varphi \alpha \ (\tau::0)] \leftrightarrow (\varphi \tau)]$ by simp

$\beta\eta$-redex is non-valid for relativized terms when modal operators are present:

- **Lemma**: $[\lambda\alpha. \Box\varphi \alpha \ (\tau::0)] \leftrightarrow (\Box\varphi \tau)]$ nitpick oops — countersatisfiable
- **Lemma**: $[\lambda\alpha. \Diamond\varphi \alpha \ (\tau::0)] \leftrightarrow (\Diamond\varphi \tau)]$ nitpick oops — countersatisfiable

Example 7.13, p. 96:

- **Lemma**: $[(\lambda X. \Diamond\exists X) \ (P::\uparrow(0)) \rightarrow \Diamond((\lambda X. \exists X) \ P)]$ by simp
- **Lemma**: $[(\lambda X. \Diamond\exists X) \downarrow(P::\uparrow(0)) \rightarrow \Diamond((\lambda X. \exists X) \downarrow P)]$
  - nitpick[card 't=1, card i=2] oops — nitpick finds same counterexample as book with other types for P:
    - **Lemma**: $[(\lambda X. \Diamond\exists X) \ (P::\uparrow(0)) \rightarrow \Diamond((\lambda X. \exists X) \ P)]$ by simp
    - **Lemma**: $[(\lambda X. \Diamond\exists X) \downarrow(P::\uparrow(0)) \rightarrow \Diamond((\lambda X. \exists X) \downarrow P)]$
      - nitpick[card 't=1, card i=2] oops — countersatisfiable
    - **Lemma**: $[(\lambda X. \Diamond\exists X) \ (P::\uparrow(0)) \rightarrow \Diamond((\lambda X. \exists X) \ P)]$ by simp
    - **Lemma**: $[(\lambda X. \Diamond\exists X) \downarrow(P::\uparrow(0)) \rightarrow \Diamond((\lambda X. \exists X) \downarrow P)]$
      - nitpick[card 't=1, card i=2] oops — countersatisfiable
Example 7.14, p. 98:

\[\text{lemma } [(\lambda X. \exists X) \downarrow (P \uparrow 0)] \rightarrow (\lambda X. \exists X) \downarrow P] \text{ by simp}\]

\[\text{lemma } [(\lambda X. \exists X) \downarrow (P \uparrow 0)] \rightarrow (\lambda X. \exists X) \downarrow P]\]

\[\text{nitpick } [\text{card 't=1, card i=2}] \text{ oops} — \text{countersatisfiable} \]

with other types for \(P\):

\[\text{lemma } [(\lambda X. \exists X) \downarrow (P \uparrow 0)] \rightarrow (\lambda X. \exists X) \downarrow P] \text{ by simp}\]

\[\text{lemma } [(\lambda X. \exists X) \downarrow (P \uparrow 0)] \rightarrow (\lambda X. \exists X) \downarrow P]\]

\[\text{nitpick } [\text{card 't=1, card i=2}] \text{ oops} — \text{countersatisfiable} \]

\[\text{lemma } [(\lambda X. \exists X) \downarrow (P \uparrow 0)] \rightarrow (\lambda X. \exists X) \downarrow P] \text{ by simp}\]

\[\text{lemma } [(\lambda X. \exists X) \downarrow (P \uparrow 0)] \rightarrow (\lambda X. \exists X) \downarrow P]\]

\[\text{nitpick } [\text{card 't=1, card i=2}] \text{ oops} — \text{countersatisfiable} \]

Example 7.15, p. 99:

\[\text{lemma } [\square (P \uparrow 0)) \rightarrow (\exists x : \uparrow 0. \square (P x))] \text{ by auto}\]

with other types for \(P\):

\[\text{lemma } [\square (P \uparrow 0)) \rightarrow (\exists x : \uparrow 0. \square (P x))] \text{ by auto}\]

\[\text{lemma } [\square (P \uparrow 0)) \rightarrow (\exists x : \uparrow 0. \square (P x))] \text{ by auto}\]

Example 7.16, p. 100:

\[\text{lemma } [\square (P \uparrow 0)) \rightarrow (\exists x : \uparrow 0. \square (P x))] \text{ by auto}\]

\[\text{nitpick } [\text{card 't=2, card i=2}] \text{ oops} — \text{counterexample with two worlds found} \]

Example 7.17, p. 101:

\[\text{lemma } [\forall Z : \uparrow 0. (\lambda x : \uparrow 0. \square ((\lambda y : \uparrow 0. x \approx y) \uparrow Z)) \uparrow Z] \]

\[\text{nitpick } [\text{card 't=2, card i=2}] \text{ oops} — \text{countersatisfiable} \]

\[\text{lemma } [\forall Z : \uparrow 0. (\lambda x : \uparrow 0. \square ((\lambda y : \uparrow 0. x \approx y) \uparrow Z)) \uparrow Z] \text{ by simp}\]

\[\text{lemma } [\forall Z : \uparrow 0. (\lambda x : \uparrow 0. \square ((\lambda y : \uparrow 0. x \approx y) \uparrow Z)) \uparrow Z] \text{ by simp}\]

3.1.2 Exercises (p. 101)

For Exercises 7.1 and 7.2 see variations on Examples 7.13 and 7.14 above.

Exercise 7.3:

\[\text{lemma } [\diamond \exists (P \uparrow 0)) \rightarrow (\exists X : \uparrow 0. \diamond (P \uparrow X))] \text{ by auto}\]

Exercise 7.4:

\[\text{lemma } [\diamond \exists x : \uparrow 0. (\lambda Y. Y x) \downarrow (P \uparrow 0))] \rightarrow (\exists x. (\lambda Y. \diamond (Y x) \downarrow P))] \]

\[\text{nitpick } [\text{card 't=1, card i=2}] \text{ oops} — \text{countersatisfiable} \]

For Exercise 7.5 see Example 7.17 above.
3.2 Miscellaneous Matters (Chapter 9)

3.2.1 Equality Axioms (Subsection 1.1)

Example 9.1:

lemma \[ ((\lambda X. \square(X \cdot (p::\tau))) \downarrow (\lambda x. \Diamond (\lambda z. z \approx x) \cdot p)) \]
by auto — using normal equality
lemma \[ ((\lambda X. \square(X \cdot (p::\tau))) \downarrow (\lambda x. \Diamond (\lambda z. z \approx x) \cdot p)) \]
by auto — using Leibniz equality
lemma \[ ((\lambda X. \square(X \cdot (p::\tau))) \downarrow (\lambda x. \Diamond (\lambda z. z \approx x) \cdot p)) \]
by simp — using equality as defined for individual concepts

3.2.2 Extensionality (Subsection 1.2)

In Fitting’s book (p. 118), extensionality is assumed (globally) for extensional terms. While Fitting introduces the following extensionality principles as axioms, they are already implicitly valid in Isabelle/HOL:

lemma EXT: \( \forall x::\{0\}. \forall y::\{0\}. \ (\forall \gamma::0. (\alpha \gamma \leftrightarrow \beta \gamma)) \rightarrow (\alpha = \beta) \) by auto
lemma EXT-set: \( \forall x::\{0\}. \forall y::\{0\}. \ (\forall \gamma::0. (\alpha \gamma \leftrightarrow \beta \gamma)) \rightarrow (\alpha = \beta) \) by auto

3.2.3 De Re and De Dicto (Subsection 2)

De re is equivalent to de dicto for non-relativized (extensional or intensional) terms:

lemma \[ \forall \alpha. \ ((\lambda \beta. \Box(\alpha \beta)) \cdot (\tau::\tau)) \leftrightarrow \Box((\lambda \beta. (\alpha \beta)) \cdot \tau) \] by simp
lemma \[ \forall \alpha. \ ((\lambda \beta. \Box(\alpha \beta)) \cdot (\tau::\tau)) \leftrightarrow \Box((\lambda \beta. (\alpha \beta)) \cdot \tau) \] by simp
lemma \[ \forall \alpha. \ ((\lambda \beta. \Box(\alpha \beta)) \cdot (\tau::\tau)) \leftrightarrow \Box((\lambda \beta. (\alpha \beta)) \cdot \tau) \] by simp
lemma \[ \forall \alpha. \ ((\lambda \beta. \Box(\alpha \beta)) \cdot (\tau::\tau)) \leftrightarrow \Box((\lambda \beta. (\alpha \beta)) \cdot \tau) \] by simp

De re is not equivalent to de dicto for relativized terms:

lemma \[ \forall \alpha. \ ((\lambda \beta. \Box(\alpha \beta)) \cdot (\tau::\tau)) \leftrightarrow \Box((\lambda \beta. (\alpha \beta)) \cdot \tau) \]
nitpick\{card \( \tau=2 \), card \( i=2 \)\} oops — countersatisfiable
lemma \[ \forall \alpha. \ ((\lambda \beta. \Box(\alpha \beta)) \cdot (\tau::\tau)) \leftrightarrow \Box((\lambda \beta. (\alpha \beta)) \cdot \tau) \]
nitpick\{card \( \tau=1 \), card \( i=2 \)\} oops — countersatisfiable

Proposition 9.6 - If we can prove one side of the equivalence, then we can prove the other (p. 120):

abbreviation deDictoImplDeRe\(::\tau\rightarrow\tau\)
where deDictoImplDeRe \(\tau\equiv\forall \alpha. \Box((\lambda \beta. (\alpha \beta)) \cdot \tau) \rightarrow (\lambda \beta. \Box(\alpha \beta)) \cdot \tau)\)
abbreviation deRelImplDeDicto\(::\tau\rightarrow\tau\)
where deRelImplDeDicto \(\tau\equiv\forall \alpha. \Box((\lambda \beta. (\alpha \beta)) \cdot \tau) \rightarrow (\lambda \beta. \Box(\alpha \beta)) \cdot \tau)\)
abbreviation deReEquivDeDicto\(::\tau\rightarrow\tau\)
where deReEquivDeDicto \(\tau\equiv\forall \alpha. \Box((\lambda \beta. (\alpha \beta)) \cdot \tau) \leftrightarrow (\lambda \beta. \Box(\alpha \beta)) \cdot \tau)\)
abbreviation deDictoImplDeRe-pred::′t⇒io⇒io
where deDictoImplDeRe-pred τ ≡ ∀α.(λβ. □((α β) ↓τ)) ↓τ
abbreviation deReImplDeDicto-pred::′t⇒io⇒io
where deReImplDeDicto-pred τ ≡ ∀α.((λβ. □((α β) ↓τ)) ↓τ) → □((λβ. (α β) ↓τ)
abbreviation deReEquDeDicto-pred::′t⇒io⇒io
where deReEquDeDicto-pred τ ≡ ∀α.((λβ. □((α β) ↓τ)) ↓τ) ↔ □((λβ. (α β) ↓τ)

We can prove local consequence:

lemma AimpB: [deReImplDeDicto (τ::↑0) → deDictoImplDeRe τ] by force — for individuals
lemma AimpB-p: [deReImplDeDicto-pred (τ::↑⟨0⟩) → deDictoImplDeRe-pred τ] by force — for predicates

And global consequence follows directly (since local consequence implies global consequence, as shown before):

lemma [deReImplDeDicto (τ::↑0)] → [deDictoImplDeRe τ] using AimpB by (rule localImpGlobalCons) — for individuals
lemma [deReImplDeDicto-pred (τ::↑⟨0⟩)] → [deDictoImplDeRe-pred τ] using AimpB-p by (rule localImpGlobalCons) — for predicates

3.2.4 Rigidity (Subsection 3)

(Local) rigidity for intensional individuals:

abbreviation rigidIndiv::′t⇒io where
rigidIndiv τ ≡ (λβ. □((λz. β ≈ z) ↓τ)) ↓τ

(Local) rigidity for intensional predicates:

abbreviation rigidPred::′t⇒io where
rigidPred τ ≡ (λβ. □((λz. β ≈ z) ↓τ)) ↓τ

Proposition 9.8 - An intensional term is rigid if and only if the de re/de dicto distinction vanishes. Note that we can prove this theorem for local consequence (global consequence follows directly).

lemma [rigidIndiv (τ::↑0)] → [deReEquDeDicto τ] by simp
lemma [deReImplDeDicto (τ::↑0) → rigidIndiv τ] by auto
lemma [rigidPred (τ::↑0)] → [deReEquDeDicto-pred τ] by simp
lemma [deReImplDeDicto-pred (τ::↑⟨0⟩) → rigidPred τ] by auto

3.2.5 Stability Conditions (Subsection 4)

axiomatization where
S5: equivalence aRel — using Sahlqvist correspondence for improved performance

Definition 9.10 - Stability conditions come in pairs:

abbreviation stabilityA::′t⇒io where stabilityA τ ≡ ∀α. (τ α) → □(τ α)
abbreviation stabilityB::′t⇒io where stabilityB τ ≡ ∀α. ◊(τ α) → (τ α)
Proposition 9.10 - In an S5 modal logic both stability conditions are equivalent.

The last proposition holds for global consequence:

\textbf{lemma} \ [\text{stabilityA} (\tau::\uparrow\langle0\rangle)] \rightarrow \ [\text{stabilityB} \ \tau] \ \textbf{using} \ S5 \ \textbf{by} \ \textbf{blast} \\
\textbf{lemma} \ [\text{stabilityB} (\tau::\uparrow\langle0\rangle)] \rightarrow \ [\text{stabilityA} \ \tau] \ \textbf{using} \ S5 \ \textbf{by} \ \textbf{blast}

But it does not hold for local consequence:

\textbf{lemma} \ [\text{stabilityA} (\tau::\uparrow\langle0\rangle) \rightarrow \text{stabilityB} \ \tau] \\
\textbf{nitpick} \ [\text{card} 't=1, \ \text{card} i=2] \ \textbf{oops} — countercatisfiable \\
\textbf{lemma} \ [\text{stabilityB} (\tau::\uparrow\langle0\rangle) \rightarrow \text{stabilityA} \ \tau] \\
\textbf{nitpick} \ [\text{card} 't=1, \ \text{card} i=2] \ \textbf{oops} — countercatisfiable

Theorem 9.11 - A term is rigid if and only if it satisfies the stability conditions. Note that we can prove this theorem for local consequence (global consequence follows directly).

\textbf{theorem} \ [\text{rigidPred} (\tau::\uparrow\langle0\rangle) \leftrightarrow (\text{stabilityA} \ \tau \ \land \ \text{stabilityB} \ \tau)] \ \textbf{by} \ \textbf{meson} \\
\textbf{theorem} \ [\text{rigidPred} (\tau::\uparrow\langle\uparrow0\rangle) \leftrightarrow (\text{stabilityA} \ \tau \ \land \ \text{stabilityB} \ \tau)] \ \textbf{by} \ \textbf{meson} \\
\textbf{theorem} \ [\text{rigidPred} (\tau::\uparrow\langle\uparrow\langle0\rangle\rangle) \leftrightarrow (\text{stabilityA} \ \tau \ \land \ \text{stabilityB} \ \tau)] \ \textbf{by} \ \textbf{meson}
4 Gödel’s Argument, Formally

"Gödel’s particular version of the argument is a direct descendent of that of Leibniz, which in turn derives from one of Descartes. These arguments all have a two-part structure: prove God’s existence is necessary, if possible; and prove God’s existence is possible.” [12], p. 138.

4.1 Part I - God’s Existence is Possible

We separate Gödel’s Argument as presented in Fitting’s textbook (ch. 11) in two parts. For the first one, while Leibniz provides some kind of proof for the compatibility of all perfections, Gödel goes on to prove an analogous result: *(T1)* Every positive property is possibly instantiated, which together with *(T2)* God is a positive property directly implies the conclusion. In order to prove T1, Gödel assumes A2: Any property entailed by a positive property is positive.

We are currently contemplating a follow-up analysis of the philosophical implications of these axioms, which encompasses some criticism of the notion of property entailment used by Gödel throughout the argument.

4.1.1 General Definitions

**abbreviation** existencePredicate::↑⟨(0)⟩ (E)

where E! x ≡ λw. (∃ y z. w ≡ y ≈ z) w — existence predicate in object language

**lemma** E! x w ←→ existsAt x w

by simp — safety check: E! correctly matches its meta-logical counterpart

**consts** positiveProperty::↑⟨↑⟨(0)⟩⟩ (P) — positiveness/perfection

Definitions of God (later shown to be equivalent under axiom A1b):

**abbreviation** God::↑⟨(0)⟩ (G) where G ≡ (λx. ∀ Y. P Y → Y x)

**abbreviation** God-star::↑⟨(0)⟩ (G*) where G* ≡ (λx. ∀ Y. P Y ↔ Y x)

Definitions needed to formalise A3:

**abbreviation** appliesToPositiveProps::↑⟨↑⟨(0)⟩⟩ (pos) where

pos Z ≡ ∀ X. Z X → P X

**abbreviation** intersectionOf::↑⟨↑⟨(0)⟩⟩ (intersec) where

intersec X Z ≡ □(∀ x. (X x ↔ (∀ Y. (Z Y) → (Y x)))) — quantifier is possibilist

**abbreviation** Entailment::↑⟨↑⟨(0)⟩⟩ (infix ⇒ 60) where

X ⇒ Y ≡ □(∀ E z. X z → Y z)
4.1.2 Axioms

axiomatization where

\[ A1a : \forall X. \mathcal{P} (\rightarrow X) \rightarrow \neg(\mathcal{P} X) \] and — axiom 11.3A

\[ A1b : \forall X. \neg(\mathcal{P} X) \rightarrow \mathcal{P} (\rightarrow X) \] and — axiom 11.3B

\[ A2 : \forall X Y. (\mathcal{P} X \wedge (X \Rightarrow Y)) \rightarrow \mathcal{P} Y \] and — axiom 11.5

\[ A3 : \forall Z X. (\text{pos } Z \wedge \text{intersec } X Z) \rightarrow \mathcal{P} X \] — axiom 11.10

lemma True nitpick[satisfy] oops — model found: axioms are consistent

lemma [D] using A1a A1b A2 by blast — axioms already imply D axiom

lemma [D] using A1a A3 by metis

4.1.3 Theorems

lemma \[ \exists X. \mathcal{P} X \] using A1b by auto

lemma \[ \exists X. \mathcal{P} X \wedge \Diamond \exists E X \] using A1a A1b A2 by metis

Being self-identical is a positive property:

lemma \[ \exists X. \mathcal{P} X \wedge \Diamond \exists E X \rightarrow \mathcal{P} (\lambda x w. x = x) \] using A2 by fastforce

Proposition 11.6

lemma \[ \exists X. \mathcal{P} X \rightarrow \mathcal{P} (\lambda x w. x = x) \] using A2 by fastforce

lemma \[ \mathcal{P} (\lambda x w. x = x) \] using A1b A2 by blast

lemma \[ \mathcal{P} (\lambda x w. x = x) \] using A3 by metis

Being non-self-identical is a negative property:

lemma \[ \exists X. \mathcal{P} X \wedge \Diamond \exists E X \rightarrow \mathcal{P} (\rightarrow (\lambda x w. \neg x = x)) \]

using A2 by fastforce

lemma \[ \exists X. \mathcal{P} X \rightarrow \mathcal{P} (\rightarrow (\lambda x w. \neg x = x)) \] using A2 by fastforce

lemma \[ \exists X. \mathcal{P} X \rightarrow \mathcal{P} (\rightarrow (\lambda x w. \neg x = x)) \] using A3 by metis

Proposition 11.7

lemma \[ \exists X. \mathcal{P} X \rightarrow \neg \mathcal{P} ((\lambda x w. \neg x = x)) \] using A1a A2 by blast

lemma \[ \neg \mathcal{P} (\lambda x w. \neg x = x) \] using A1a A2 by blast

Proposition 11.8 (Informal Proposition 1) - Positive properties are possibly instantiated:

theorem T1: \[ \forall X. \mathcal{P} X \rightarrow \Diamond \exists E X \] using A1a A2 by blast

Proposition 11.14 - Both defs (God/God*) are equivalent. For improved performance we may prefer to use one or the other:

lemma GodDefsAreEquivalent: \[ \forall x. \text{G } x \leftrightarrow \text{G* } x \] using A1b by force

Proposition 11.15 - Possibilist existence of God directly implies A1b:

lemma \[ \exists \text{G* } \rightarrow (\forall X. \neg(\mathcal{P} X) \rightarrow \mathcal{P} (\rightarrow X)) \] by meson
Proposition 11.16 - $A3$ implies $P(G)$ (local consequence):

**Lemma** $A3implT2\text{-}local$: $\forall Z X. \ (\text{pos } Z \land \text{intersec } X \ Z) \rightarrow P \ X \rightarrow P \ G$

**Proof**

- Fix $w$
  - Have 1: $\text{pos } P \ w$ by simp
  - Have 2: $\text{intersec } G \ P \ w$ by simp
    - Assume $\forall Z X. \ (\text{pos } Z \land \text{intersec } X \ Z) \rightarrow P \ X$ $w$ by (rule allE)
    - Hence 3: $((\text{pos } P) \land (\text{intersec } G \ P)) \rightarrow P \ G \ w$ by (rule allE)
    - Hence 4: $((\text{pos } P) \land (\text{intersec } G \ P)) \ w$ using 1 2 by simp
      - From 3 4 have $P \ G \ w$ by (rule mp)
    - Hence $\forall Z X. \ (\text{pos } Z \land \text{intersec } X \ Z) \rightarrow P \ X) \ w \rightarrow P \ G \ w$ by (rule impl)
  - Thus ?thesis by (rule allI)

**Qed**

$A3$ implies $P(G)$ (as global consequence):

**Lemma** $A3implT2\text{-}global$: $\forall Z X. \ (\text{pos } Z \land \text{intersec } X \ Z) \rightarrow P \ X \rightarrow [P \ G]$

**Using** $A3implT2\text{-}local$ by (rule localImpGlobalCons)

Being Godlike is a positive property. Note that this theorem can be axiomatized directly, as noted by Dana Scott (see [12], p. 152). We will do so for the second part.

**Theorem** $T2$: $[P \ G]$ using $A3implT2\text{-}global$ $A3$ by simp

Theorem 11.17 (Informal Proposition 3) - Possibly God exists:

**Theorem** $T3$: $\diamond \exists E \ G$ using $T1$ $T2$ by simp

### 4.2 Part II - God’s Existence is Necessary if Possible

We show here that God’s necessary existence follows from its possible existence by adding some additional (potentially controversial) assumptions including an essentialist premise and the $S5$ axioms. Further results like monotheism and the rejection of free will (modal collapse) are also proved.

#### 4.2.1 General Definitions

**Abbreviation** $\text{existencePredicate}:: \uparrow\langle 0 \rangle \ (E!)$ where $E! \ x \equiv (\lambda w. \ (\exists E \ y. \ y \approx x) \ w)$

**Constants** $\text{positiveProperty}:: \uparrow\langle \langle 0 \rangle \rangle \ (P)$
abbreviation God::\(\uparrow\langle 0 \rangle \ (G)\) where \(G \equiv (\lambda x. \forall Y. P \ Y \rightarrow Y \ x)\)

abbreviation God-star::\(\uparrow\langle 0 \rangle \ (G^*)\) where
\(G^* \equiv (\lambda x. \forall Y. P \ Y \leftrightarrow Y \ x)\)

abbreviation Entailment::\(\uparrow\langle\uparrow\langle 0 \rangle, \uparrow\langle 0 \rangle\rangle\) (infix \(\Rightarrow\)) where
\(X \Rightarrow Y \equiv \Box (\forall^E \ z. X \ z \rightarrow Y \ z)\)

4.2.2 Results from Part I

Note that the only use Gödel makes of axiom A3 is to show that being Godlike is a positive property \((T2)\). We follow therefore Scott’s proposal and take \((T2)\) directly as an axiom:

axiomatization where

\(A1a: [\forall X. P (\rightarrow X) \rightarrow \neg(\neg P \ X)] \text{ and} \) — axiom 11.3A
\(A1b: [\forall X. \neg(\neg P \ X) \rightarrow P (\rightarrow X)] \text{ and} \) — axiom 11.3B
\(A2: [\forall X Y. (P \ X \land (X \Rightarrow Y)) \rightarrow P \ Y] \text{ and} \) — axiom 11.5
\(T2: [P \ G] \) — proposition 11.16

lemma True nitpick[satisfy] oops — model found: axioms are consistent

lemma [D] using \(A1a A1b A2\) by blast — axioms already imply D axiom

lemma GodDefsAreEquivalent: \([\forall x. G \ x \leftrightarrow G^* \ x]\) using \(A1b\) by fastforce

theorem T1: \([\forall X. P \ X \rightarrow \Diamond \exists^E \ X]\)
using \(A1a A2\) by blast — positive properties are possibly instantiated

theorem T3: \([\Diamond \exists^E \ G]\) using \(T1 T2\) by simp — God exists possibly

4.2.3 Axioms

\(P\) satisfies the so-called stability conditions (see [12], p. 124), which means it designates rigidly (note that this makes for an essentialist assumption).

axiomatization where

\(A4a: [\forall X. P \ X \rightarrow \Box(\neg P \ X)]\) — axiom 11.11

lemma \(A4b: [\forall X. \neg(\neg P \ X) \rightarrow \Box \neg(\neg P \ X)]\) using \(A1a A1b A4a\) by blast

abbreviation rigidPred::\(\langle t \Rightarrow io\rangle \Rightarrow io\) where
rigidPred \(\tau \equiv (\lambda \beta. \Box((\lambda z. \beta \equiv z) \downarrow \tau)) \downarrow \tau\)

lemma \(\text{[rigidPred} P]\)
using \(A4a A4b\) by blast — \(P\) is therefore rigid

lemma True nitpick[satisfy] oops — model found: so far all axioms A1-4 consistent
4.2.4 Theorems

Remark: Essence is defined here (and in Fitting’s variant) in the version of Scott; Gödel’s original version leads to the inconsistency reported in [8, 9]

abbreviation essenceOf::"∪{0,0} (E) where
\( E \ x \equiv (Y \ x) \land (\forall \ Z. Z \ x \rightarrow Y \Rightarrow Z) \)

abbreviation beingIdenticalTo::0⇒{0} (id) where
\( \text{id} \ x \equiv (\lambda y. y \approx x) \) — note that id is a rigid predicate

Theorem 11.20 - Informal Proposition 5

theorem GodIsEssential: [∀ x. G x → (E G x)] using A1b A4a by meson

Theorem 11.21

theorem [∀ x. G* x → (E G* x)] using A4a by meson

Theorem 11.22 - Something can have only one essence:

theorem [∀ X Y z. (E X z ∧ E Y z) → (X ⇒ Y)] by meson

Theorem 11.23 - An essence is a complete characterization of an individual:

theorem EssencesCharacterizeCompletely: [∀ X y. E X y → (X ⇒ (id y))]

proof (rule ccontr)
assume \( \neg [∀ X y. E X y \rightarrow (X ⇒ (id y))] \)
hence \( ∃ w. \neg (∀ X y. E X y \rightarrow X ⇒ id y) \ w \) by simp
then obtain w where \( \neg (∀ X y. E X y \rightarrow X ⇒ id y) \ w \) ..
hence \( (∃ X y. E X y ∧ \neg (X ⇒ id y)) \ w \) by simp
then obtain P where \( ∃ y. E P y w ∧ (\neg (P ⇒ id y)) \ w \) ..
then obtain a where \( 1: E P a w ∧ (\neg (P ⇒ id a)) \ w \) ..
hence 2: \( E P a w \) by (rule conjunct1)
from 1 have \( (\neg (P ⇒ id a)) \ w \) by (rule conjunct2)
hence \( ∃ x. ∃ z. w r x ∧ \exists i s t a t z x ∧ P \equiv \neg (a = z) \) by blast
then obtain w1 where \( ∃ z. w r w1 ∧ ꔐ z w1 ∧ P \equiv \neg (a = z) \) ..
then obtain b where \( 3: w r w1 ∧ ꔐ z w1 ∧ P b w1 ∧ ꔐ (a = b) \) ..
hence \( w r w1 \) by simp
from 3 have ꔐ z w1 b w1 by simp
from 3 have P b w1 by simp
from 3 have 4: ꔐ (a = b) by simp
from 2 have P a w by simp
from 2 have ∀ Y. Y a w → (((P ⇒ Y) w) by auto
hence \( (\neg (id b)) a w \rightarrow (P ⇒ (\neg (id b))) \ w \) by (rule allE)
hence \( (\neg (id b)) a w \lor ((P ⇒ (\neg (id b))) \ w \) by blast
then show False proof
assume \( (\neg (id b)) a w \)
hence \( a = b \) by simp
thus False using 4 by auto
next
assume \( (P ⇒ (\neg (id b))) \ w \)
hence \( ∃ x. ∃ z. (w r x ∧ ꔐ z x ∧ P \equiv z x) \rightarrow (\neg (id b)) \ z x \) by blast
hence $\forall z. (w r w1 \land \text{existsAt } z w1 \land P z w1) \rightarrow (\neg(id b)) z w1$
  by (rule allE)

hence $(w r w1 \land \text{existsAt } b w1 \land P b w1) \rightarrow (\neg(id b)) b w1$ by (rule allE)

hence $\neg(w r w1 \land \text{existsAt } b w1 \land P b w1) \lor (\neg(id b)) b w1$ by simp

hence $\neg(b=b)$ by simp

thus False by simp

qed

Definition 11.24 - Necessary Existence (Informal Definition 6):

abbreviation necessaryExistencePred::↑⟨0⟩ (NE)
  where NE x ≡ ($\lambda w. (\forall Y. E Y x \rightarrow \square \exists E Y) w$)

Axiom 11.25 (Informal Axiom 5)

axiomatization where
  $A5: [P NE]$

lemma True nitpick[satisfy] oops — model found: so far all axioms consistent

Theorem 11.26 (Informal Proposition 7) - Possibilist existence of God implies necessary actualist existence:

theorem GodExistenceImpliesNecExistence: $[\exists G \rightarrow \square \exists E \ G]$

proof –

  { fix w
    
    assume $\exists x. G x w$
    
    then obtain $g$ where 1: $G g w$ ..
    
    hence $NE g w$ using $A5$ by auto
    — axiom 11.25
    
    hence $\forall Y. (E Y g w) \rightarrow (\square \exists E Y) w$ by simp
    
    hence 2: $(E G g w) \rightarrow (\square \exists G) w$ by (rule allE)
    
    have $(\forall x. G x \rightarrow (E G x)) w$ using GodIsEssential
      by (rule allE) — GodIsEssential follows from Axioms 11.11 and 11.3B
      
      hence $(G g \rightarrow (E G g)) w$ by (rule allE)
    
    hence $G g w \rightarrow E G g w$ by simp
    
    from this 1 have 3: $E G g w$ by (rule mp)
    
    from 2 3 have $(\square \exists E G) w$ by (rule mp)
    
    } hence $(\exists x. G x w) \rightarrow (\square \exists E G) w$ by (rule mp)

  } from this 1 have 3: $E G g w$ by (rule mp)

  } hence $(\exists x. G x) \rightarrow (\square \exists E G) w$ by simp

  thus $\square \text{thesis}$ by (rule allI)

  qed

Modal collapse is countersatisfiable (unless we introduce S5 axioms):

lemma $[\forall \Phi.(\Phi \rightarrow (\square \Phi))]$ nitpick oops
We postulate semantic frame conditions for some modal logics. Taken together, reflexivity, transitivity and symmetry make for an equivalence relation and therefore an \( S5 \) logic (via Sahlqvist correspondence). We prefer to postulate them individually here in order to get more detailed information about their relevance in the proofs presented below.

axiomatization where
refl: reflexive aRel and
tran: transitive aRel and
symm: symmetric aRel

lemma True nitpick[satisfy] oops — model found: axioms still consistent

Using an \( S5 \) logic, modal collapse (\[\forall \Phi . (\Phi \rightarrow (\Box \Phi))\]) is actually valid (see ‘More Objections’ some pages below)

We prove some useful inference rules:

lemma modal-distr: \[\Box(\phi \rightarrow \psi) \implies (\Box \phi \rightarrow \Box \psi)\] by blast

lemma modal-trans: \[(\phi \rightarrow \psi) \land (\psi \rightarrow \chi) \implies (\phi \rightarrow \chi)\] by simp

Theorem 11.27 - Informal Proposition 8. Note that only symmetry and transitivity for the accessibility relation are used.

theorem possExistenceImpliesNecEx: \[\Diamond \exists G \rightarrow \Box \exists E G\] — local consequence

proof
have \[\exists G \rightarrow \Box \exists E G\] using GodExistenceImpliesNecExistence
  by simp — follows from Axioms 11.11, 11.25 and 11.3B
hence \[\Box(\exists G \rightarrow \Box \exists E G)\] using NEC by simp
hence \[\Diamond \exists G \rightarrow \Box \exists E G\] by (rule modal-distr)
have 2: \[\Diamond \Box \exists E G \rightarrow \Box \exists E G\] using symm tran by metis — frame conditions
from 1 2 have \[\Diamond \exists G \rightarrow \Diamond \Box \exists E G \land [\Diamond \Box \exists E G \rightarrow \Box \exists E G]\] by simp
thus ?thesis by (rule modal-trans)
qed

lemma T4: \[\Diamond \exists G \rightarrow [\Box \exists E G]\] using possExistenceImpliesNecEx
  by (rule localImpGlobalCons) — global consequence

Corollary 11.28 - Necessary (actualist) existence of God (for both definitions); reflexivity is still not used:

lemma GodNecExists: \[\Box \exists E G\] using T3 T4 by metis
lemma God-starNecExists: \[\Box \exists E G^*\]
  using GodNecExists GodDefsAreEquivalent by simp

4.2.5 Monotheism

Monotheism for non-normal models (with Leibniz equality) follows directly from God having all and only positive properties:

theorem Monotheism-LeibnizEq: \[\forall x . G x \rightarrow (\forall y . G y \rightarrow (x \approx^L y))\]
  using GodDefsAreEquivalent by simp
Monotheism for normal models is trickier. We need to consider some previous results (p. 162):

**Lemma GodExistenceIsValid**: $\exists E \; G$ using GodNecExists refl by auto — reflexivity is now required by the solver

**Proposition 11.29:**

**Theorem Monotheism-normalModel**: $\exists x. \forall y. \; G y \leftrightarrow x \approx y$

**Proof** —

- fix $w$
- have $\exists E \; G$ using GodExistenceIsValid by simp — follows from corollary 11.28
- hence $\exists E \; G$ $w$ by (rule allE)
- then obtain $g$ where $1$: existsAt $g$ $w$ $\wedge$ $G g$ ..
- hence $E \; G \; g \; w$ using GodIsEssential by blast — follows from ax. 11.11/11.3B

{ fix $y$
- have $G \; y \; w$ $\longleftrightarrow$ $(g \approx y)$ $w$ proof
  - assume $G \; y \; w$
  - hence $E \; G \; y \; w$ using GodIsEssential by blast
  - have $(E \; G \; y \rightarrow (G \Rightarrow id \; y))$ $w$ using EssencesCharacterizeCompletely
    - by simp — follows from theorem 11.23
  - hence $(E \; G \; y \; w)$ $\rightarrow$ $((G \Rightarrow id \; y) \; w)$ by simp
  - from this $3$ have $(G \Rightarrow id \; y) \; w$ by (rule mp)
  - hence $(\Box (\forall z. \; G \; z \rightarrow z \approx y))$ $w$ by simp
  - hence $\forall x. \; w \; r \; x$ $\rightarrow$ $(\forall z. \; (\exists z \; \wedge \; G \; z \; x) \rightarrow z \approx y))$ by auto
  - hence $w \; r \; w$ $\rightarrow$ $(\forall z. \; (\exists z \; \wedge \; G \; z \; w) \rightarrow z \approx y))$ by (rule allE)
  - hence $\forall z. \; (w \; r \; w \; \wedge \; \exists z \; \wedge \; G \; z \; w) \rightarrow z \approx y$ by auto
  - hence $4$: $(w \; r \; w \; \wedge \; \exists z \; \wedge \; G \; z \; w) \rightarrow g \approx y$ $w$ by (rule allE)
  - have $w \; r \; w$ using refl
    - by simp — using frame reflexivity (Axiom M)
  - hence $w \; r \; w \bot$ $(\exists z \; \wedge \; G \; g \; w)$ using $1$ by (rule conjI)
  - from $4$ this have $g \approx y$ $w$ by (rule mp)
  - thus $(g \approx y)$ $w$ by simp

next
- assume $(g \approx y)$ $w$
- from this $2$ have $E \; G \; y \; w$ by simp
- thus $G \; y \; w$ by (rule conjunct1)

qed

{ hence $\forall y. \; G \; y \; w$ $\leftrightarrow$ $(g \approx y)$ $w$ by (rule allI)
- hence $\exists x. \; (\forall y. \; G \; y \; w$ $\leftrightarrow$ $(x \approx y)$ $w$) by (rule exI)
- hence $(\exists x. \; (\forall y. \; G \; y \leftrightarrow (x \approx y)))$ $w$ by simp

thus ?thesis by (rule allI)

qed

**Corollary 11.30:**
lemma GodImpliesExistence: \[\forall x. \ G x \to E! x\]
using GodExistenceIsValid Monotheism-normalModel by metis

4.2.6 Positive Properties are Necessarily Instantiated

lemma PosPropertiesNecExist: \[\forall Y. \ P Y \to \Box \exists E Y\]
using GodNecExists A4a by meson — proposition 11.31: follows from corollary 11.28 and axiom A4a

4.2.7 More Objections

Fitting discusses the objection raised by Sobel [19], who argues that G"odel’s axiom system is too strong: it implies that whatever is the case is so necessarily, i.e. the modal system collapses \(\phi \to \Box \phi\). The modal collapse has been philosophically interpreted as implying the absence of free will.

We start by proving an useful FOL lemma:

lemma useful: \[(\forall x. \ \phi x \to \psi) \Rightarrow (\exists x. \ \phi x \to \psi)\] by simp

In the context of our S5 axioms, the modal collapse becomes valid (pp. 163-4):

lemma ModalCollapse: \[\forall \Phi. \ (\Phi \to (\Box \Phi))\]
proof -
{
fix w
{
fix Q
have \((\forall x. \ G x \to (E G x))\) w using GodIsEssential
by (rule allE) — follows from Axioms 11.11 and 11.3B
hence \(\forall x. \ G x w \to E G x w\) by simp
hence \(\forall x. \ G x w \to (\forall Z. \ Z x \to \Box(\forall E z. \ G z \to Z z))\) w by force
hence \(\forall x. \ G x w \to ((\lambda y. \ Q y) x \to \Box(\forall E z. \ G z \to (\lambda y. \ Q z)))\) w by force
hence \(\forall x. \ G x w \to (Q \to \Box(\forall E z. \ G z \to Q))\) w by simp
hence \(1: (\exists x. \ G x w) \to ((Q \to \Box(\forall E z. \ G z \to Q)) w)\) by (rule useful)
have \(\exists x. \ G x w\) using GodExistenceIsValid by auto
from 1 this have \((Q \to \Box(\forall E z. \ G z \to Q))\) w by (rule imp)
hence \((Q \to \Box(\exists E z. \ G z \to Q))\) w using useful by blast
hence \((Q \to \Box(\exists E z. \ G z \to Q))\) w by simp
hence \((Q \to \Box Q)\) w using GodNecExists by simp
}
hence \(\forall \Phi. \ \Phi \to \Box \Phi\) w by (rule allI)
}
thus ?thesis by (rule allI)
qed
5 Fitting’s Solution

In this section we consider Fitting’s solution to the objections raised in his discussion of Gödel’s Argument pp. 164-9, especially the problem of modal collapse, which has been metaphysically interpreted as implying a rejection of free will. Since we are generally committed to the existence of free will (in a pre-theoretical sense), such a result is philosophically unappealing and rather seen as a problem in the argument’s formalisation.

This part of the book still leaves several details unspecified and the reader is thus compelled to fill in the gaps. As a result, we came across some premises and theorems allowing for different formalisations and therefore leading to disparate implications. Only some of those cases are shown here for illustrative purposes. The options we have chosen here are such that they indeed validate the argument (and we assume that they correspond to Fitting’s intention.

5.1 General Definitions

The following is an existence predicate for our object-language. (We have previously shown it is equivalent to its meta-logical counterpart.)

**abbreviation** existencePredicate::↑⟨0⟩ (E!)
where
E! x ≡ (λw. (∃y. y≈x) w)

Reminder: The ‘(|-|’ parenthesis are used to convert an extensional object into its ‘rigid’ intensional counterpart (e.g. (|-φ|) ≡ λw. φ).

**consts** positiveProperty::↑⟨⟨0⟩⟩ (P)
**abbreviation** God::↑⟨0⟩ (G)
where
G ≡ (λx. ∀Y. P Y → (|Y x|))

**abbreviation** God-star::↑⟨0⟩ (G*)
where
G* ≡ (λx. ∀Y. P Y ↔ (|Y x|))

**abbreviation** Entailment::↑⟨⟨0⟩⟩, ⟨0⟩⟩ (infix ⭕)
where
X ⭕ Y ≡ □(∀E. (|X z|) → (|Y z|))

5.2 Part I - God’s Existence is Possible

**axiomatization** where
A1a: ∀X. P (→X) → ¬(P X) ] and — axiom 11.3A
A1b: ∀X. ¬(P X) → P (→X) ] and — axiom 11.3B
A2: ∀X Y. (P X ∨ (X ⇒ Y)) → P Y ] and — axiom 11.5
T2: [P ↓G] — proposition 11.16 (modified)

**lemma** True nitpick[satisfy] oops — model found: axioms are consistent

**lemma** [D] using A1a A1b A2 by blast — axioms already imply D axiom

**lemma** GodDefsAreEquivalent: [∀x. G x ↔ G* x] using A1b by fastforce
Theorem T1 (Positive properties are possibly instantiated) can be formalised in two different ways:

\[\text{theorem T1a: } [\forall X::(0). \ P X \rightarrow \Diamond (\exists E z. \ \langle X z \rangle)]\]

using A1a A2 by blast — this is the one used in the book

\[\text{theorem T1b: } [\forall X::\langle 0 \rangle. \ P \downarrow X \rightarrow \Diamond (\exists E z. \ X z)]\]

nitpick oops — this one is also possible but not valid so we won’t use it

Some interesting (non-)equivalences:

\[\text{lemma } [\Box \exists E (Q::\langle 0 \rangle) \leftrightarrow \Box \exists E (Q::\downarrow G)] \text{ by simp}\]

\[\text{lemma } [\Box \exists E (Q::\langle 0 \rangle) \leftrightarrow ((\lambda X. \ \exists E X) Q)] \text{ by simp}\]

\[\text{lemma } [\Box \exists E (Q::\langle 0 \rangle) \leftrightarrow ((\lambda X. \ \exists E X) \downarrow Q)] \text{ nitpick oops — not equivalent!}\]

Theorem T3 (God exists possibly) can be formalised in two different ways, using a de re or a de dicto reading.

\[\text{theorem T3-deRe: } [(\lambda X. \ \exists E X) \downarrow G] \text{ using T1a T2 by simp}\]

\[\text{theorem T3-deDicto: } [\exists E \downarrow G] \text{ nitpick oops — countersatisfiable}\]

From the last two theorems, we think T3-deRe should be the version originally implied in the book, since T3-deDicto is not valid (T1b were valid but it isn’t)

\[\text{lemma assumes T1b: } [\forall X. \ P \downarrow X \rightarrow \Diamond (\exists E z. \ X z)]\]

\[\text{shows T3-deDicto: } [\exists E \downarrow G] \text{ using assms T2 by simp}\]

5.3 Part II - God's Existence is Necessary if Possible

In this variant \(P\) also designates rigidly, as shown in the last section.

axiomatization where

\[\text{A4a: } [\forall X. \ P X \rightarrow \Box (P X)] \text{ — axiom 11.11}\]

\[\text{lemma A4b: } [\forall X. \ \neg(P X) \rightarrow \Box \neg(P X)] \text{ using A1a A1b A4a by blast}\]

\[\text{lemma True nitpick[satisfy] oops — model found: so far all axioms consistent}\]

abbreviation \(\text{essenceOf::} \uparrow\langle 0 \rangle.0 \ (E) \text{ where}\)

\(E \ Y x \equiv [Y x] \wedge (\forall Z::(0). [Z x] \rightarrow Y \Rightarrow Z)\)

Theorem 11.20 - Informal Proposition 5

\[\text{theorem GodIsEssential: } [\forall x. \ G x \rightarrow ((E \downarrow G) x)] \text{ using A1b by metis}\]

Theorem 11.21

\[\text{theorem God-starIsEssential: } [\forall x. \ G^* x \rightarrow ((E \downarrow G^*) x)] \text{ by meson}\]

abbreviation \(\text{necExistencePred::} \uparrow\langle 0 \rangle.0 \ (NE) \text{ where}\)

\(NE x \equiv \lambda w. (\forall Y. \ E Y x \rightarrow \Box [\exists E z. \langle Y z \rangle]) w\)
Informal Axiom 5

axiomatization where

\[ A5 : [P \downarrow NE] \]

\[ \text{lemma True nitpick[satisfy]} \text{ oops} \] — model found: so far all axioms consistent

Reminder: We use \[ \downarrow \] instead of \[ G \] because it is more explicit. See (non-)equivalences above.

\[ \text{lemma } [\exists G \leftrightarrow \exists \downarrow G] \text{ by simp} \]

\[ \text{lemma } [\exists E G \leftrightarrow \exists E \downarrow G] \text{ by simp} \]

\[ \text{lemma } [\square \exists E G \leftrightarrow \square \exists E \downarrow G] \text{ by simp} \]

Theorem 11.26 (Informal Proposition 7) - (possibilist) existence of God implies necessary (actualist) existence.

There are two different ways of formalising this theorem. Both of them are proven valid:

First version:

\[ \text{theorem GodExImpliesNecEx-v1: } [\exists \downarrow G \rightarrow \square \exists E \downarrow G] \]

\[ \text{proof} \]

\{ fix \[ w \] \}

\{ assume \[ \exists x. G x w \] \}

\[ \text{then obtain } g \text{ where 1: } G g w \text{ ..} \]

\[ \text{hence } NE g w \text{ using } A5 \text{ by auto} \]

\[ \text{hence } \forall Y. (E \downarrow Y g w) \rightarrow (\square (\exists E z. \langle Y z \rangle)) w \text{ by simp} \]

\[ \text{hence } (E (\lambda x. G x w) g w) \rightarrow (\square (\exists E z. \langle (\lambda x. G x w) z \rangle)) w \text{ by (rule allE)} \]

\[ \text{hence 2: } (E \downarrow_1 G) g w \rightarrow (\square (\exists E G)) w \text{ using } A4b \text{ by meson} \]

\[ \text{have } (\forall x. G x \rightarrow ((E \downarrow_1 G) x)) w \text{ using GodIsEssential by (rule allE)} \]

\[ \text{hence } (G g \rightarrow ((E \downarrow_1 G) g)) w \text{ by (rule allE)} \]

\[ \text{hence } G g w \rightarrow (E \downarrow_1 G) g w \text{ by simp} \]

\[ \text{from this 1 have 3: } (E \downarrow_1 G) g w \text{ by (rule mp)} \]

\[ \text{from 2 3 have } (\exists E G) w \text{ by (rule mp)} \]

\} 

\[ \text{hence } (\exists x. G x w) \rightarrow (\exists E G) w \text{ by (rule impI)} \]

\[ \text{hence } (\exists x. G x) \rightarrow (\exists E G) w \text{ by simp} \]

\} 

\[ \text{thus } ?\text{thesis by (rule allI)} \]

qed

Second version (which can be proven directly by automated tools using the previous version):

\[ \text{theorem GodExImpliesNecEx-v2: } [\exists \downarrow G \rightarrow (\lambda X. \square \exists E X) \downarrow G] \]

\[ \text{using } A4a \text{ GodExImpliesNecEx-v1 by metis} \]

In contrast to Gödel’s argument (as presented by Fitting), the following theorems can be proven in logic \( K \) (the \( S5 \) axioms are no longer needed):
Theorem 11.27 - Informal Proposition 8

**Theorem** possExImpliesNecEx-v1: $[\lozenge \exists \downarrow G \rightarrow \Box \exists \downarrow E \downarrow G]$

**using** GodExImpliesNecEx-v1 T3-deRe by metis

**Theorem** possExImpliesNecEx-v2: $[(\lambda X. \lozenge \exists \downarrow E X) \downarrow G \rightarrow ((\lambda X. \Box \exists \downarrow E X) \downarrow G)]$

**using** GodExImpliesNecEx-v2 by blast

**Corollaries:**

**Lemma** T4-v1: $[\lozenge \exists \downarrow G] \rightarrow [\Box \exists \downarrow G]$

**using** possExImpliesNecEx-v1 by simp

**Lemma** T4-v2: $[(\lambda X. \lozenge \exists \downarrow E X) \downarrow G] \rightarrow [(\lambda X. \Box \exists \downarrow E X) \downarrow G]$

**using** possExImpliesNecEx-v2 by simp

5.4 Conclusion (De Re and De Dicto Reading)

Version I - Necessary Existence of God (de dicto):

**Lemma** GodNecExists-v1: $[\Box \exists \downarrow G]$

**using** GodExImpliesNecEx-v1 T3-deRe by fastforce — corollary 11.28

**Lemma** God-starNecExists-v1: $[\Box \exists \downarrow G^{*}]$

**using** GodNecExists-v1 GodDefsAreEquivalent by simp

**Lemma** $[(\lambda X. \exists \downarrow E X) \downarrow G^{*}]$

**using** God-starNecExists-v1 by simp — de dicto shown here explicitly

Version II - Necessary Existence of God (de re)

**Lemma** GodNecExists-v2: $[(\lambda X. \Box \exists \downarrow E X) \downarrow G]$

**using** T3-deRe T4-v2 by blast

**Lemma** God-starNecExists-v2: $[(\lambda X. \Box \exists \downarrow E X) \downarrow G^{*}]$

**using** GodNecExists-v2 GodDefsAreEquivalent by simp

5.5 Modal Collapse

Modal collapse is countersatisfiable even in $S5$. Note that countermodels with a cardinality of one for the domain of individuals are found by Nitpick (the countermodel shown in the book has cardinality of two).

**Lemma** $[\forall \Phi. (\Phi \rightarrow (\Box \Phi))]$

**Nitpick[card ‘t=1, card i=2]** oops — countermodel found in $K$

axiomatization where

$S5$: equivalence aRel — assume $S5$ logic

**Lemma** $[\forall \Phi. (\Phi \rightarrow (\Box \Phi))]$

**Nitpick[card ‘t=1, card i=2]** oops — countermodel also found in $S5$
6 Anderson’s Alternative

In this final section, we verify Anderson’s emendation of Gödel’s argument, as it is presented in the last part of the textbook by Fitting (pp. 169-171).

6.1 General Definitions

abbreviation \textit{existencePredicate}::\uparrow\langle 0 \rangle (E)
where \text{E}! x \equiv \lambda w. (\exists y E y x) w

consts \textit{positiveProperty}::\uparrow\langle\uparrow\langle 0 \rangle\rangle (P)

abbreviation \textit{God}::\uparrow\langle 0 \rangle (G^A)
where \text{G}^A \equiv \lambda x. \forall Y. (P Y) \leftrightarrow \Box(Y x)

abbreviation \textit{Entailment}::\uparrow\langle\uparrow\langle 0 \rangle, \uparrow\langle 0 \rangle\rangle (infix \Rightarrow 60)
where \text{X} \Rightarrow \text{Y} \equiv \Box(\forall z X z \rightarrow Y z)

6.2 Part I - God’s Existence is Possible

axiomatization where
A1a: [\forall X. P (\rightarrow X) \rightarrow \neg(P X)] \text{ and } — Axiom 11.3A
A2: [\forall X Y. (P X \land (X \Rightarrow Y)) \rightarrow P Y] \text{ and } — Axiom 11.5
T2: [P G^A] — Proposition 11.16

lemma True \text{ nitpick}[\text{satisfy}] \text{ oops} — model found: axioms are consistent

theorem T1: [\forall X. P X \rightarrow \Diamond \exists E X]
using A1a A2 by blast — positive properties are possibly instantiated

theorem T3: [\Diamond \exists E G^A] using T1 T2 by simp — God exists possibly

6.3 Part II - God’s Existence is Necessary if Possible

\text{P} now satisfies only one of the stability conditions. But since the argument uses an S5 logic, the other stability condition is implied. Therefore \text{P} becomes rigid (see p. 124).

axiomatization where
A4a: [\forall X. \text{P} X \rightarrow \Box(P X)] — axiom 11.11

We again postulate our S5 axioms:

axiomatization where
refl: reflexive aRel and
tran: transitive aRel and
symm: symmetric aRel

lemma True \text{ nitpick}[\text{satisfy}] \text{ oops} — model found: so far all axioms consistent

abbreviation \textit{rigidPred}::('t\Rightarrow io)\Rightarrow io
where
\text{rigidPred} \ \tau \equiv (\lambda \beta. \Box((\lambda z. \beta z) \downarrow \tau)) \downarrow \tau

28
lemma A4b: \( \forall X. \neg (P X) \rightarrow \Box \neg (P X) \)
using A4a symm by auto — symmetry is needed (which corresponds to B axiom)

lemma [rigidPred P]
using A4a A4b by blast — P is therefore rigid in a B logic

Essence, Anderson Version (Definition 11.34)

abbreviation essenceOf::\( \uparrow \langle \uparrow \langle 0 \rangle, 0 \rangle \) (E^A)
where
E^A Y x \equiv (\forall Z. \Box (Z x) \leftrightarrow Y \Rightarrow Z)

Necessary Existence, Anderson Version (Definition 11.35)

abbreviation necessaryExistencePred::\( \uparrow \langle 0 \rangle \) (NE^A)
where NE^A x \equiv (\lambda w. (\forall Y. E^A Y x \rightarrow \Box \exists E Y) w)

Theorem 11.36 - If g is God-like, then the property of being God-like is the essence of g.

As shown before, this theorem’s proof could be completely automatized for Gödel’s and Fitting’s variants. For Anderson’s version however, we had to provide Isabelle with some help based on the corresponding natural-language proof given by Anderson (see [2] Theorem 2*, p. 296)

theorem GodIsEssential: \( \forall x. G^A x \rightarrow (E^A G^A x) \)
proof –
{  
  fix w
  
  fix g
  
  assume G^A g w
  
  hence 1: \( \forall Y. (P Y w) \leftrightarrow (\Box (Y g)) w \) by simp
  
  {  
    fix Q
    
    from 1 have 2: \( (P Q w) \leftrightarrow (\Box (Q g)) w \) by (rule allE)
    
    have \( (\Box (Q g)) w \leftrightarrow (G^A \Rightarrow Q) w \) — we need to prove \( \rightarrow \) and \( \leftarrow \)
    
    proof
      assume \( (\Box (Q g)) \) — suppose g is God-like and necessarily has Q
      hence 3: \( (P Q w) \) using 2 by simp — then Q is positive
      
      
      {  
        fix u
        
        have \( (P Q u) \rightarrow (\forall x. G^A x u \rightarrow (\Box (Q x)) u) \)
        
        by auto — using the definition of God-like
        
        have \( (P Q u) \rightarrow (\forall x. G^A x u \rightarrow ((Q x)) u) \)
        
        using refl by auto — and using \( \Box (\varphi x) \rightarrow \varphi x \)
      }
      
      hence \( \forall z. (P Q z) \rightarrow (\forall x. G^A x z \rightarrow Q x z) \) by (rule allI)
      hence \( P Q \rightarrow (\forall x. G^A x \rightarrow Q x) \)
      by auto — if Q is positive, then whatever is God-like has Q
      
      by auto — if Q is positive, then whatever is God-like has Q
      
      29

hence $\Box(P \rightarrow (\forall x. G^A x \rightarrow Q x)]$ by (rule NEC)

hence $[\Box(P Q)) \rightarrow \Box(\forall x. G^A x \rightarrow Q x)]$ using $K$ by auto

hence $((\Box(P Q)) \rightarrow G^A \Rightarrow Q)$ by simp

hence $((\Box(P Q)) \rightarrow G^A \Rightarrow Q) w$ by (rule allE)

hence $4: (\Box(P Q)) w \rightarrow (G^A \Rightarrow Q) w$ by simp

have $[\forall X. P X \rightarrow \Box(P X)]$ by (rule A4a) — using axiom 4

hence $(\forall X. P X \rightarrow (\Box(P X)) w$ by (rule allE)

hence $P Q w \rightarrow (\Box(P Q)) w$ by (rule allE)

hence $P Q w \rightarrow (G^A \Rightarrow Q) w$ using 4 by simp

thus $(G^A \Rightarrow Q) w$ using 3 by (rule mp) —→ direction

next

assume 5: $(G^A \Rightarrow Q) w$ — suppose Q is entailed by being God-like

have $[\forall X. Y. (P X \wedge (X \Rightarrow Y)) \rightarrow P Y] w$ by (rule A2)

hence $(\forall X Y. (P X \wedge (X \Rightarrow Y)) \rightarrow P Y) w$ by (rule allE)

hence $(\forall X Y. (P X w \wedge (X \Rightarrow Y) w) \rightarrow P Y w$ by simp

hence $(\forall Y. (P G^A w \wedge (G^A \Rightarrow Y) w) \rightarrow P Y w$ by (rule allE)

hence 6: $(P G^A w \wedge (G^A \Rightarrow Q) w) \rightarrow P Q w$ by (rule allE)

have $(P G^A] w$ by (rule T2)

hence $P G^A w$ by (rule allE)

hence $(G^A \Rightarrow Q) w$ using 5 by (rule conjI)

from 6 this have $P Q w$ by (rule mp) — Q is positive by A2 and T2

thus $(\Box(Q g)) w$ using 2 by simp

qed

Axiom 11.37 (Anderson’s version of 11.25)

axiomatization where

$A5: [P NE^A]$


lemma True nitpick[satisfy] oops — model found: so far all axioms consistent

Theorem 11.38 - Possibilist existence of God implies necessary actualist existence:

theorem GodExistenceImpliesNecExistence: $[\exists G^A \rightarrow \Box\forall E G^A]$

proof —

{ fix $w$

{
assume ∃x. G\(^A\) x w
then obtain g where 1: G\(^A\) g w

hence NE\(^A\) g w using A5 by blast — axiom 11.25
hence ∀Y. (E\(^A\) Y g w) → (□∃\(^E\) Y) w by simp
hence 2: (E\(^A\) G\(^A\) g w) → (□∃\(^E\) G\(^A\)) w by (rule allE)

have (∀x. G\(^A\) x → (E\(^A\) G\(^A\) x)) w using GodIsEssential
by (rule allE) — axiom 11.11 and 11.3B
hence (∀Y. (∀E Y g w) → (∀E Y g w)) by blast
hence (∃x. G\(^A\) x w) → E\(^A\) G\(^A\) g w by simp

hence (E\(^A\) G\(^A\) g w) → (∀E G\(^A\) g w)

hence (∃x. G\(^A\) x w) → (□∃\(^E\) G\(^A\)) w by (rule mp)

thus ?thesis by (rule allI)
qed

Some useful rules:

lemma modal-distr: [□(ϕ → ψ)] |= [(□ϕ → □ψ)] by blast
lemma modal-trans: [(ϕ → ψ) ∧ [ψ → χ]] |= [ϕ → χ] by simp

Anderson’s version of Theorem 11.27

case possExistenceImpliesNecEx: [□∃ G\(^A\) → □∃\(^E\) G\(^A\)] — local consequence

proof —

have [∃ G\(^A\) → □∃\(^E\) G\(^A\)] using GodExistenceImpliesNecExistence
by simp — follows from Axioms 11.11, 11.25 and 11.3B
hence [□(∃ G\(^A\) → □∃\(^E\) G\(^A\))] using NEC by simp

hence 1: [□∃ G\(^A\) → □□∃\(^E\) G\(^A\)] by (rule modal-distr)
have 2: [□□∃\(^E\) G\(^A\) → □□∃\(^E\) G\(^A\)] using symm tran by metis
from 1 2 have [□∃ G\(^A\) → □□∃\(^E\) G\(^A\)] ∧ [□□∃\(^E\) G\(^A\) → □□∃\(^E\) G\(^A\)] by simp

thus ?thesis by (rule modal-trans)
qed

lemma T4: [□∃ G\(^A\)] → [□∃\(^E\) G\(^A\)] using possExistenceImpliesNecEx
by (rule localImpGlobalCons) — global consequence

Conclusion - Necessary (actualist) existence of God:

lemma GodNecExists: [□∃\(^E\) G\(^A\)] using T3 T4 by metis

6.4 Modal Collapse

Modal collapse is countersatisfiable

lemma [∀Φ.(Φ → (□ Φ))] nitpick oops
We presented a shallow semantical embedding in Isabelle/HOL for an intensional higher-order modal logic (a successor of Montague/Gallin intensional logics) as introduced by M. Fitting in his textbook *Types, Tableaus and Gödel’s God* [12]. We subsequently employed this logic to formalise and verify all results (theorems, examples and exercises) relevant to the discussion of Gödel’s ontological argument in the last part of Fitting’s book. Three different versions of the ontological argument have been considered: the first one by Gödel himself (respectively, Scott), the second one by Fitting and the last one by Anderson.

By employing an interactive theorem-prover like Isabelle, we were not only able to verify Fitting’s results, but also to guarantee consistency. We could prove even stronger versions of many of the theorems and find better counterexamples (i.e. with smaller cardinality) than the ones presented in the book. Another interesting aspect was the possibility to explore the implications of alternative formalisations for definitions and theorems which shed light on interesting philosophical issues concerning entailment, essentialism and free will, which are currently the subject of some follow-up analysis.

The latest developments in *automated theorem proving* allow us to engage in much more experimentation during the formalisation and assessment of arguments than ever before. The potential reduction (of several orders of magnitude) in the time needed for proving or disproving theorems (compared to pen-and-paper proofs), results in almost real-time feedback about the suitability of our speculations. The practical benefits of computer-supported argumentation go beyond mere quantitative (easier, faster and more reliable proofs). The advantages are also qualitative, since it fosters a different approach to argumentation: We can now work iteratively (by ‘trial-and-error’) on an argument by making gradual adjustments to its definitions, axioms and theorems. This allows us to continuously expose and revise the assumptions we indirectly commit ourselves everytime we opt for some particular formalisation.
References


