Types, Tableaus and Gödel's God in Isabelle/HOL

David Fuenmayor[1] and Christoph Benzmüller[2]

[1]Freie Universität Berlin, Germany
[2]University of Luxemburg, Luxemburg & Freie Universität Berlin, Germany

April 20, 2017

Abstract

A computer-formalisation of the most essential parts of Fitting's textbook *Types, Tableaus and Gödel's Go*d in Isabelle/HOL is presented. In particular, Fitting'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

We present a study in Computational Metaphysics: a computer-formalisation and verification of Fitting's emendation of the ontological argument (for the existence of God) as presented in his well-known textbook *Types, Tableaus and Gödel's God* [8]. Fitting's argument is an emendation of Kurt Gödel's modern variant [9] (resp. Dana Scott's variant [10]) of the ontological argument.

The motivation is to avoid the modal collapse [11, 12], 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 (see e.g. [2, 1, **?**, **?**] 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 [9] have been shown inconsistent [3, 7] while Scott's version has been verified [5]. Further experiments, contributing amongst others to the clarification of a related debate between Hajek and Anderson, are presented and discussed in [6]. The enabling technique that has been employed in all of these experiments has been shallow semantical embeddings of (extensional) higher-order modal logics in classical higher-order logic (see [6, 4] and the references therein).

Fitting's emendation also intends to avoid the modal collapse. In contrast to the above emendations, 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 an shallow embedding of this more advanced logic. The experiments presented below confirm that Fitting's argument as presented in [8] 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.

2 Embedding of Intensional Higher-Order Modal Logic

The following shallow embedding of Intensional Higher-Order Modal Logic (IHOML) in Isabelle/HOL is inspired by the work of [6]. We expand this approach to allow for intensional types and actualist quantifiers as employed in Fitting's textbook ([8]).

2.1 Declarations

typedecl i — Type for possible worlds
type-synonym $io = (i \Rightarrow bool)$ — Type for formulas whose truth-value is world-dependent
typedecl e $(\mathbf{0})$ — Type for individuals

Aliases for common unary predicate types:

type-synonym $ie = (i \Rightarrow \mathbf{0})$ $(\uparrow\mathbf{0})$
type-synonym $se = (\mathbf{0} \Rightarrow bool)$ $(\langle\mathbf{0}\rangle)$

type-synonym $ise = \ (\mathbf{0}{\Rightarrow}io) \quad (\uparrow\langle\mathbf{0}\rangle)$
type-synonym $sie = \ (\uparrow\mathbf{0}{\Rightarrow}bool) \quad (\langle\uparrow\mathbf{0}\rangle)$
type-synonym $isie = \ (\uparrow\mathbf{0}{\Rightarrow}io) \quad (\uparrow\langle\uparrow\mathbf{0}\rangle)$
type-synonym $sise = \ (\uparrow\langle\mathbf{0}\rangle{\Rightarrow}bool) \quad (\langle\uparrow\langle\mathbf{0}\rangle\rangle)$
type-synonym $isise = \ (\uparrow\langle\mathbf{0}\rangle{\Rightarrow}io) \quad (\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle)$
type-synonym $sisise = \ (\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle{\Rightarrow}bool) \ (\langle\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle\rangle)$
type-synonym $isisise = \ (\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle{\Rightarrow}io) \ (\uparrow\langle\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle\rangle)$
type-synonym $sse = \ \langle\mathbf{0}\rangle{\Rightarrow}bool \quad (\langle\langle\mathbf{0}\rangle\rangle)$
type-synonym $isse = \ \langle\mathbf{0}\rangle{\Rightarrow}io \quad (\uparrow\langle\langle\mathbf{0}\rangle\rangle)$

Aliases for common binary relation types:

type-synonym $see = \ (\mathbf{0}{\Rightarrow}\mathbf{0}{\Rightarrow}bool) \quad (\langle\mathbf{0},\mathbf{0}\rangle)$
type-synonym $isee = \ (\mathbf{0}{\Rightarrow}\mathbf{0}{\Rightarrow}io) \quad (\uparrow\langle\mathbf{0},\mathbf{0}\rangle)$
type-synonym $sieie = \ (\uparrow\mathbf{0}{\Rightarrow}\uparrow\mathbf{0}{\Rightarrow}bool) \quad (\langle\uparrow\mathbf{0},\uparrow\mathbf{0}\rangle)$
type-synonym $isieie = \ (\uparrow\mathbf{0}{\Rightarrow}\uparrow\mathbf{0}{\Rightarrow}io) \quad (\uparrow\langle\uparrow\mathbf{0},\uparrow\mathbf{0}\rangle)$
type-synonym $ssese = \ (\langle\mathbf{0}\rangle{\Rightarrow}\langle\mathbf{0}\rangle{\Rightarrow}bool) \quad (\langle\langle\mathbf{0}\rangle,\langle\mathbf{0}\rangle\rangle)$
type-synonym $issese = \ (\langle\mathbf{0}\rangle{\Rightarrow}\langle\mathbf{0}\rangle{\Rightarrow}io) \quad (\uparrow\langle\langle\mathbf{0}\rangle,\langle\mathbf{0}\rangle\rangle)$
type-synonym $ssee = \ (\langle\mathbf{0}\rangle{\Rightarrow}\mathbf{0}{\Rightarrow}bool) \quad (\langle\langle\mathbf{0}\rangle,\mathbf{0}\rangle)$
type-synonym $issee = \ (\langle\mathbf{0}\rangle{\Rightarrow}\mathbf{0}{\Rightarrow}io) \quad (\uparrow\langle\langle\mathbf{0}\rangle,\mathbf{0}\rangle)$
type-synonym $isisee = \ (\uparrow\langle\mathbf{0}\rangle{\Rightarrow}\mathbf{0}{\Rightarrow}io) \quad (\uparrow\langle\uparrow\langle\mathbf{0}\rangle,\mathbf{0}\rangle)$
type-synonym $isiseise = \ (\uparrow\langle\mathbf{0}\rangle{\Rightarrow}\uparrow\langle\mathbf{0}\rangle{\Rightarrow}io) \quad (\uparrow\langle\uparrow\langle\mathbf{0}\rangle,\uparrow\langle\mathbf{0}\rangle\rangle)$
type-synonym $isiseisise = \ (\uparrow\langle\mathbf{0}\rangle{\Rightarrow}\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle{\Rightarrow}io) \ (\uparrow\langle\uparrow\langle\mathbf{0}\rangle,\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle\rangle)$

consts $aRel{::}i{\Rightarrow}i{\Rightarrow}bool$ (**infixr** r 70) — Accessibility relation

2.2 Definition of Logical Operators

abbreviation $mnot \ :: io{\Rightarrow}io \ (\neg\text{-}[52]53)$
  where $\neg\varphi \equiv \lambda w.\ \neg(\varphi\ w)$
abbreviation $mand \ :: io{\Rightarrow}io{\Rightarrow}io$ (**infixr**$\wedge 51$)
  where $\varphi\wedge\psi \equiv \lambda w.\ (\varphi\ w)\wedge(\psi\ w)$
abbreviation $mor \ :: io{\Rightarrow}io{\Rightarrow}io$ (**infixr**$\vee 50$)
  where $\varphi\vee\psi \equiv \lambda w.\ (\varphi\ w)\vee(\psi\ w)$
abbreviation $mimp \ :: io{\Rightarrow}io{\Rightarrow}io$ (**infixr**$\rightarrow 49$)
  where $\varphi\rightarrow\psi \equiv \lambda w.\ (\varphi\ w)\longrightarrow(\psi\ w)$
abbreviation $mequ \ :: io{\Rightarrow}io{\Rightarrow}io$ (**infixr**$\leftrightarrow 48$)
  where $\varphi\leftrightarrow\psi \equiv \lambda w.\ (\varphi\ w)\longleftrightarrow(\psi\ w)$
abbreviation $xor{::}\ bool{\Rightarrow}bool{\Rightarrow}bool$ (**infixr**$\oplus 50$)
  where $\varphi\oplus\psi \equiv \ (\varphi\vee\psi)\ \wedge\ \neg(\varphi\wedge\psi)$
abbreviation $mxor \ :: io{\Rightarrow}io{\Rightarrow}io$ (**infixr**$\bigoplus 50$)
  where $\varphi\bigoplus\psi \equiv \lambda w.\ (\varphi\ w)\oplus(\psi\ w)$

2.3 Definition of Posibilist Quantifiers

abbreviation $mforall \ :: ('t{\Rightarrow}io){\Rightarrow}io \ (\forall)$
  where $\forall\,\Phi \equiv \lambda w.\forall\, x.\ (\Phi\ x\ w)$
abbreviation $mexists \ :: ('t{\Rightarrow}io){\Rightarrow}io \ (\exists)$
  where $\exists\,\Phi \equiv \lambda w.\exists\, x.\ (\Phi\ x\ w)$

abbreviation *mforallB* :: $('t{\Rightarrow}io){\Rightarrow}io$ (**binder**$\forall$ [*8*]*9*) — Binder notation
where $\forall\, x.\ \varphi(x) \equiv \forall\, \varphi$
abbreviation *mexistsB* :: $('t{\Rightarrow}io){\Rightarrow}io$ (**binder**$\exists$ [*8*]*9*)
where $\exists\, x.\ \varphi(x) \equiv \exists\, \varphi$

2.4 Definition of Actualist Quantifiers

The following predicate is used to model actualist quantifiers by restricting domains of quantification. Note that since this is a meta-logical concept we never use it in our object language.

consts *Exists*::$\uparrow\langle\mathbf{0}\rangle$ (*existsAt*)

Note that no polymorphic types are needed in the definitions since actualist quantification only makes sense for individuals.

abbreviation *mforallAct* :: $\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle$ ($\forall^E$)
where $\forall^E\Phi \equiv \lambda w.\forall\, x.\ (existsAt\ x\ w){\longrightarrow}(\Phi\ x\ w)$
abbreviation *mexistsAct* :: $\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle$ ($\exists^E$)
where $\exists^E\Phi \equiv \lambda w.\exists\, x.\ (existsAt\ x\ w) \wedge (\Phi\ x\ w)$

abbreviation *mforallActB* :: $\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle$ (**binder**$\forall^E$[*8*]*9*) — Binder notation
where $\forall^E x.\ \varphi(x) \equiv \forall^E\varphi$
abbreviation *mexistsActB* :: $\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle$ (**binder**$\exists^E$[*8*]*9*)
where $\exists^E x.\ \varphi(x) \equiv \exists^E\varphi$

2.5 Definition of Modal Operators

abbreviation *mbox* :: $io{\Rightarrow}io$ ($\Box$-[*52*]*53*)
where $\Box\varphi \equiv \lambda w.\forall\, v.\ (w\ r\ v){\longrightarrow}(\varphi\ v)$
abbreviation *mdia* :: $io{\Rightarrow}io$ ($\Diamond$-[*52*]*53*)
where $\Diamond\varphi \equiv \lambda w.\exists\, v.\ (w\ r\ v){\wedge}(\varphi\ v)$

2.6 Definition of the *extension-of* Operator

In contrast to the approach taken in Fitting's book (p. 88), the $\downarrow$ operator is embedded as a binary operator applying to (world-dependent) atomic formulas whose first argument is a 'relativized' term (preceded by $\downarrow$). Depending on the types involved we need to define this operator differently to ensure type correctness.

(a) Predicate φ takes an (intensional) individual concept as argument:

abbreviation *mextIndiv*::$\uparrow\langle\mathbf{0}\rangle{\Rightarrow}\uparrow\mathbf{0}{\Rightarrow}io$ (**infix** $\downharpoonleft$ *60*)
where $\varphi\ {\downharpoonleft}c \equiv \lambda w.\ \varphi\ (c\ w)\ w$

(b) Predicate φ takes an intensional predicate as argument:

abbreviation *mextPredArg*::$(('t{\Rightarrow}io){\Rightarrow}io){\Rightarrow}('t{\Rightarrow}io){\Rightarrow}io$ (**infix** $\downarrow$ *60*)
where $\varphi\ {\downarrow}P \equiv \lambda w.\ \varphi\ (\lambda x\ u.\ P\ x\ w)\ w$

(c) Predicate φ takes an extensional predicate as argument:

abbreviation $extPredArg::(('t{\Rightarrow}bool){\Rightarrow}io){\Rightarrow}('t{\Rightarrow}io){\Rightarrow}io$ (**infix** $\downarrow$ *60*)
where $\varphi \downarrow P \equiv \lambda w.\ \varphi\ (\lambda x.\ P\ x\ w)\ w$

(d) Predicate φ takes an extensional predicate as first argument:

abbreviation $extPredArg1::(('t{\Rightarrow}bool){\Rightarrow}'b{\Rightarrow}io){\Rightarrow}('t{\Rightarrow}io){\Rightarrow}'b{\Rightarrow}io$ (**infix** $\downarrow_1$ *60*)
where $\varphi \downarrow_1 P \equiv \lambda z.\ \lambda w.\ \varphi\ (\lambda x.\ P\ x\ w)\ z\ w$

2.7 Definition of Equality

abbreviation meq :: $'t{\Rightarrow}'t{\Rightarrow}io$ (**infix**$\approx$*60*) — normal equality (for all types)
where $x \approx y \equiv \lambda w.\ x = y$
abbreviation $meqC$ :: $\uparrow\langle\uparrow\mathbf{0},\uparrow\mathbf{0}\rangle$ (**infixr**$\approx^C$*52*) — eq. for individual concepts
where $x \approx^C y \equiv \lambda w.\ \forall v.\ (x\ v) = (y\ v)$
abbreviation $meqL$ :: $\uparrow\langle\mathbf{0},\mathbf{0}\rangle$ (**infixr**$\approx^L$*52*) — Leibniz eq. for individuals
where $x \approx^L y \equiv \forall\varphi.\ \varphi(x) {\rightarrow} \varphi(y)$

2.8 Miscellaneous

abbreviation $negpred$ :: $\langle\mathbf{0}\rangle{\Rightarrow}\langle\mathbf{0}\rangle$ ($\rightharpoondown$-[*52*]*53*)
where $\rightharpoondown\Phi \equiv \lambda x.\ \neg(\Phi\ x)$
abbreviation $mnegpred$ :: $\uparrow\langle\mathbf{0}\rangle{\Rightarrow}\uparrow\langle\mathbf{0}\rangle$ ($\rightharpoondown$-[*52*]*53*)
where $\rightharpoondown\Phi \equiv \lambda x.\lambda w.\ \neg(\Phi\ x\ w)$
abbreviation $mandpred$ :: $\uparrow\langle\mathbf{0}\rangle{\Rightarrow}\uparrow\langle\mathbf{0}\rangle{\Rightarrow}\uparrow\langle\mathbf{0}\rangle$ (**infix &** *53*)
where $\Phi\ \&\ \varphi \equiv \lambda x.\lambda w.\ (\Phi\ x\ w) \wedge (\varphi\ x\ w)$

2.9 Meta-logical Predicates

abbreviation $valid$ :: $io{\Rightarrow}bool$ ($\lfloor\text{-}\rfloor$ [*8*]) **where** $\lfloor\psi\rfloor \equiv \forall w.(\psi\ w)$
abbreviation $satisfiable$ :: $io{\Rightarrow}bool$ ($\lfloor\text{-}\rfloor^{sat}$ [*8*]) **where** $\lfloor\psi\rfloor^{sat} \equiv \exists w.(\psi\ w)$
abbreviation $countersat$ :: $io{\Rightarrow}bool$ ($\lfloor\text{-}\rfloor^{csat}$ [*8*]) **where** $\lfloor\psi\rfloor^{csat} \equiv \exists w.\neg(\psi\ w)$
abbreviation $invalid$ :: $io{\Rightarrow}bool$ ($\lfloor\text{-}\rfloor^{inv}$ [*8*]) **where** $\lfloor\psi\rfloor^{inv} \equiv \forall w.\neg(\psi\ w)$

2.10 Verifying the Embedding

Verifying K Principle and Necessitation:

lemma *K*: $\lfloor(\Box(\varphi \rightarrow \psi)) \rightarrow (\Box\varphi \rightarrow \Box\psi)\rfloor$ **by** *simp* — K Schema
lemma *NEC*: $\lfloor\varphi\rfloor \Longrightarrow \lfloor\Box\varphi\rfloor$ **by** *simp* — Necessitation

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

lemma $\lfloor(\forall x.\Box(\varphi\ x)) \rightarrow \Box(\forall x.(\varphi\ x))\rfloor$ **by** *simp*
lemma $\lfloor\Box(\forall x.(\varphi\ x)) \rightarrow (\forall x.\Box(\varphi\ x))\rfloor$ **by** *simp*

(Converse) Barcan Formulas not satisfied for actualist quantifiers:

lemma $\lfloor(\forall^E x.\Box(\varphi\ x)) \rightarrow \Box(\forall^E x.(\varphi\ x))\rfloor$ **nitpick oops** — countersatisfiable
lemma $\lfloor\Box(\forall^E x.(\varphi\ x)) \rightarrow (\forall^E x.\Box(\varphi\ x))\rfloor$ **nitpick oops** — countersatisfiable

Well known relations between meta-logical notions:

lemma $\lfloor\varphi\rfloor \longleftrightarrow \neg\lfloor\varphi\rfloor^{csat}$ **by** *simp*
lemma $\lfloor\varphi\rfloor^{sat} \longleftrightarrow \neg\lfloor\varphi\rfloor^{inv}$ **by** *simp*

Contingent truth does not allow for necessitation:

lemma $\lfloor\Diamond\varphi\rfloor \longrightarrow \lfloor\Box\varphi\rfloor$ **nitpick oops** — countersatisfiable
lemma $\lfloor\Box\varphi\rfloor^{sat} \longrightarrow \lfloor\Box\varphi\rfloor$ **nitpick oops** — countersatisfiable

Modal Collapse is countersatisfiable:

lemma $\lfloor\varphi \rightarrow \Box\varphi\rfloor$ **nitpick oops** — countersatisfiable

2.11 Useful Definitions for Axiomatization of Further Logics

The best known logics (*K4, K5, KB, K45, KB5, D, D4, D5, D45, ...*) are obtained through axiomatization of combinations of the following:

abbreviation M
where $M \equiv \forall\varphi.\ \Box\varphi \rightarrow \varphi$
abbreviation B
where $B \equiv \forall\varphi.\ \varphi \rightarrow \Box\Diamond\varphi$
abbreviation D
where $D \equiv \forall\varphi.\ \Box\varphi \rightarrow \Diamond\varphi$
abbreviation IV
where $IV \equiv \forall\varphi.\ \Box\varphi \rightarrow \Box\Box\varphi$
abbreviation V
where $V \equiv \forall\varphi.\ \Diamond\varphi \rightarrow \Box\Diamond\varphi$

Because the embedding is of a semantic nature, it is more efficient to instead make use of the well-known *Sahlqvist correspondence*, which links axioms to constraints on a model's accessibility relation: axioms M, B, D, IV, V impose reflexivity, symmetry, seriality, transitivity and euclideanness respectively.

lemma *reflexive aRel* $\Longrightarrow \lfloor M\rfloor$ **by** *blast* — aka T
lemma *symmetric aRel* $\Longrightarrow \lfloor B\rfloor$ **by** *blast*
lemma *serial aRel* $\Longrightarrow \lfloor D\rfloor$ **by** *blast*
lemma *preorder aRel* $\Longrightarrow \lfloor M\rfloor \wedge \lfloor IV\rfloor$ **by** *blast* — S4 - reflexive + transitive
lemma *equivalence aRel* $\Longrightarrow \lfloor M\rfloor \wedge \lfloor V\rfloor$ **by** *blast* — S5 - preorder + symmetric

lemma *reflexive aRel* $\wedge$ *euclidean aRel* $\Longrightarrow \lfloor M\rfloor \wedge \lfloor V\rfloor$ **by** *blast* — S5

Using these definitions, we can derive axioms for the most common modal logics. 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 Book Examples

In this section we verify that our embedded logic works as intended by proving the examples provided in the book. In many cases, for good mesure, we consider further theorems derived from the original ones. We were able to confirm that all results (proves or counterexamples) agree with our expectations.

3.1 Modal Logic - Syntax and Semantics (Chapter 7)

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

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

lemma $\lfloor((\lambda\alpha.\ \varphi\ \alpha)\ \ (\tau{::}\uparrow\mathbf{0})) \leftrightarrow (\varphi\ \ \tau)\rfloor$ **by** *simp*
lemma $\lfloor((\lambda\alpha.\ \varphi\ \alpha)\ \ (\tau{::}\mathbf{0})) \leftrightarrow (\varphi\ \ \tau)\rfloor$ **by** *simp*
lemma $\lfloor((\lambda\alpha.\ \Box\varphi\ \alpha)\ (\tau{::}\uparrow\mathbf{0})) \leftrightarrow (\Box\varphi\ \tau)\rfloor$ **by** *simp*
lemma $\lfloor((\lambda\alpha.\ \Box\varphi\ \alpha)\ (\tau{::}\mathbf{0})) \leftrightarrow (\Box\varphi\ \tau)\rfloor$ **by** *simp*

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

lemma $\lfloor((\lambda\alpha.\ \varphi\ \alpha)\ \rfloor(\tau{::}\uparrow\mathbf{0})) \leftrightarrow (\varphi\ \rfloor\tau)\rfloor$ **by** *simp*

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

lemma $\lfloor((\lambda\alpha.\ \Box\varphi\ \alpha)\ \rfloor(\tau{::}\uparrow\mathbf{0})) \leftrightarrow (\Box\varphi\ \rfloor\tau)\rfloor$ **nitpick oops** — countersatisfiable
lemma $\lfloor((\lambda\alpha.\ \Diamond\varphi\ \alpha)\ \rfloor(\tau{::}\uparrow\mathbf{0})) \leftrightarrow (\Diamond\varphi\ \rfloor\tau)\rfloor$ **nitpick oops** — countersatisfiable

Example 7.13, p. 96:

lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ \ (P{::}\uparrow\langle\mathbf{0}\rangle) \rightarrow \Diamond((\lambda X.\ \exists\, X)\ \ P)\rfloor$ **by** *simp*
lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ \downarrow(P{::}\uparrow\langle\mathbf{0}\rangle) \rightarrow \Diamond((\lambda X.\ \exists\, X)\ \downarrow P)\rfloor$
nitpick[*card 't=1*, *card i=2*] **oops** — nitpick finds same counterexample as book

with other types for P:

lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ \ (P{::}\uparrow\langle\uparrow\mathbf{0}\rangle) \rightarrow \Diamond((\lambda X.\ \exists\, X)\ \ P)\rfloor$ **by** *simp*
lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ \downarrow(P{::}\uparrow\langle\uparrow\mathbf{0}\rangle) \rightarrow \Diamond((\lambda X.\ \exists\, X)\ \downarrow P)\rfloor$
nitpick[*card 't=1*, *card i=2*] **oops** — countersatisfiable
lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ \ (P{::}\uparrow\langle\langle\mathbf{0}\rangle\rangle) \rightarrow \Diamond((\lambda X.\ \exists\, X)\ \ P)\rfloor$ **by** *simp*
lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ \downarrow(P{::}\uparrow\langle\langle\mathbf{0}\rangle\rangle) \rightarrow \Diamond((\lambda X.\ \exists\, X)\ \downarrow P)\rfloor$
nitpick[*card 't=1*, *card i=2*] **oops** — countersatisfiable
lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ \ (P{::}\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle) \rightarrow \Diamond((\lambda X.\ \exists\, X)\ \ P)\rfloor$ **by** *simp*
lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ \downarrow(P{::}\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle) \rightarrow \Diamond((\lambda X.\ \exists\, X)\ \downarrow P)\rfloor$
nitpick[*card 't=1*, *card i=2*] **oops** — countersatisfiable

Example 7.14, p. 98:

lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ \downarrow(P{::}\uparrow\langle\mathbf{0}\rangle) \rightarrow (\lambda X.\ \exists\, X)\ \downarrow P\rfloor$ **by** *simp*
lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ \ (P{::}\uparrow\langle\mathbf{0}\rangle) \rightarrow (\lambda X.\ \exists\, X)\ \ P\rfloor$
nitpick[*card 't=1*, *card i=2*] **oops** — countersatisfiable

with other types for P:

lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ \downarrow(P{::}\uparrow\langle\uparrow\mathbf{0}\rangle) \rightarrow (\lambda X.\ \exists\, X)\ \downarrow P\rfloor$ **by** *simp*

lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ \ (P{::}\uparrow\langle\uparrow\mathbf{0}\rangle) \rightarrow (\lambda X.\ \exists\, X)\ \ P\rfloor$
  nitpick[*card 't=1, card i=2*] **oops** — countersatisfiable
lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ {\downarrow}(P{::}\uparrow\langle\langle\mathbf{0}\rangle\rangle) \rightarrow (\lambda X.\ \exists\, X)\ {\downarrow}P\rfloor$ **by** *simp*
lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ \ (P{::}\uparrow\langle\langle\mathbf{0}\rangle\rangle) \rightarrow (\lambda X.\ \exists\, X)\ \ P\rfloor$
  nitpick[*card 't=1, card i=2*] **oops** — countersatisfiable
lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ {\downarrow}(P{::}\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle){\rightarrow}\ (\lambda X.\ \exists\, X)\ {\downarrow}P\rfloor$ **by** *simp*
lemma $\lfloor(\lambda X.\ \Diamond\exists\, X)\ \ (P{::}\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle){\rightarrow}\ (\lambda X.\ \exists\, X)\ \ P\rfloor$
  nitpick[*card 't=1, card i=2*] **oops** — countersatisfiable

Example 7.15, p. 99:

lemma $\lfloor\Box(P\ \ (c{::}\uparrow\mathbf{0})) \rightarrow (\exists\, x{::}\uparrow\mathbf{0}.\ \Box(P\ x))\rfloor$ **by** *auto*

with other types for P:

lemma $\lfloor\Box(P\ \ (c{::}\mathbf{0})) \rightarrow (\exists\, x{::}\mathbf{0}.\ \Box(P\ x))\rfloor$ **by** *auto*
lemma $\lfloor\Box(P\ \ (c{::}\langle\mathbf{0}\rangle)) \rightarrow (\exists\, x{::}\langle\mathbf{0}\rangle.\ \Box(P\ x))\rfloor$ **by** *auto*

Example 7.16, p. 100:

lemma $\lfloor\Box(P\ {\downharpoonleft}(c{::}\uparrow\mathbf{0})) \rightarrow (\exists\, x{::}\mathbf{0}.\ \Box(P\ x))\rfloor$
  nitpick[*card 't=2, card i=2*] **oops** — counterexample with two worlds found

Example 7.17, p. 101:

lemma $\lfloor\forall\, Z{::}\uparrow\mathbf{0}.\ (\lambda x{::}\mathbf{0}.\ \ \Box((\lambda y{::}\mathbf{0}.\ \ x \approx y)\ {\downharpoonleft}Z))\ {\downharpoonleft}Z\rfloor$
  nitpick[*card 't=2, card i=2*] **oops** — countersatisfiable
lemma $\lfloor\forall\, z{::}\mathbf{0}.\ \ (\lambda x{::}\mathbf{0}.\ \ \Box((\lambda y{::}\mathbf{0}.\ \ x \approx y)\ \ z))\ z\rfloor$ **by** *simp*
lemma $\lfloor\forall\, Z{::}\uparrow\mathbf{0}.\ (\lambda X{::}\uparrow\mathbf{0}.\ \Box((\lambda\, Y{::}\uparrow\mathbf{0}.\ X \approx\ Y)\ \ \ Z))\ Z\rfloor$ **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:

lemma $\lfloor\Diamond\exists\,(P{::}\uparrow\langle\mathbf{0}\rangle) \rightarrow (\exists\, X{::}\uparrow\mathbf{0}.\ \Diamond(P\ {\downharpoonleft}X))\rfloor$ **by** *auto*
lemma $\lfloor\Diamond\exists\,(P{::}\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle) \rightarrow (\exists\, X{::}\uparrow\langle\mathbf{0}\rangle.\ \Diamond(P\ {\downarrow}X))\rfloor$
  nitpick[*card 't=1, card i=2*] **oops** — countersatisfiable

Exercise 7.4:

lemma $\lfloor\Diamond(\exists\, x{::}\mathbf{0}.\ (\lambda\, Y.\ \ Y\ x)\ {\downarrow}(P{::}\uparrow\langle\mathbf{0}\rangle)) \rightarrow (\exists\, x.\ (\lambda\, Y.\ \Diamond(\,Y\ x))\ {\downarrow}P)\rfloor$
  nitpick[*card 't=1, card i=2*] **oops** — 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 $\lfloor((\lambda X.\ \Box(X\ {\downharpoonleft}(p{::}\uparrow\mathbf{0})))\ {\downarrow}(\lambda x.\ \Diamond(\lambda z.\ z \approx x)\ {\downharpoonleft}p))\rfloor$

by *auto* — using normal equality
lemma $\lfloor((\lambda X.\ \Box(X \downharpoonleft(p{::}\uparrow\mathbf{0})))\ \downarrow(\lambda x.\ \Diamond(\lambda z.\ z \approx^{L} x)\ \downharpoonleft p))\rfloor$
by *auto* — using Leibniz equality
lemma $\lfloor((\lambda X.\ \Box(X\ \ (p{::}\uparrow\mathbf{0})))\ \downarrow(\lambda x.\ \Diamond(\lambda z.\ z \approx^{C} x)\ p))\rfloor$
by *simp* — using equality as defined for individual concepts

3.2.2 Extensionality (Subsection 1.2)

In the book, extensionality is assumed (globally) for extensional terms. Extensionality is however already implicit in Isabelle/HOL as we can see:

lemma *EXT*: $\forall\,\alpha{::}\langle\mathbf{0}\rangle.\ \forall\,\beta{::}\langle\mathbf{0}\rangle.\ (\forall\,\gamma{::}\mathbf{0}.\ (\alpha\ \gamma \longleftrightarrow \beta\ \gamma)) \longrightarrow (\alpha = \beta)$ **by** *auto*
lemma *EXT-set*: $\forall\,\alpha{::}\langle\langle\mathbf{0}\rangle\rangle.\ \forall\,\beta{::}\langle\langle\mathbf{0}\rangle\rangle.\ (\forall\,\gamma{::}\langle\mathbf{0}\rangle.\ (\alpha\ \gamma \longleftrightarrow \beta\ \gamma)) \longrightarrow (\alpha = \beta)$
by *auto*

Extensionality for intensional terms is also already implicit in the HOL embedding:

lemma *EXT-int*: $\lfloor(\lambda x.\ ((\lambda y.\ x{\approx}y)\ \downharpoonleft(\alpha{::}\uparrow\mathbf{0}\ )))\ \ \downharpoonleft(\beta{::}\uparrow\mathbf{0})\rfloor \longrightarrow \alpha = \beta$ **by** *auto*
lemma *EXT-int-pred*: $\lfloor(\lambda x.\ ((\lambda y.\ x{\approx}y)\ \downarrow(\alpha{::}\uparrow\langle\mathbf{0}\rangle)))\ \downarrow(\beta{::}\uparrow\langle\mathbf{0}\rangle)\rfloor \longrightarrow \alpha = \beta$
using *ext* **by** *metis*

3.2.3 D*e Re* and *De Dicto* (Subsection 2)

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

lemma $\lfloor\forall\,\alpha.\ ((\lambda\beta.\ \Box(\alpha\ \beta))\ (\tau{::}\mathbf{0}))\ \ \ \leftrightarrow \Box((\lambda\beta.\ (\alpha\ \beta))\ \tau)\rfloor$ **by** *simp*
lemma $\lfloor\forall\,\alpha.\ ((\lambda\beta.\ \Box(\alpha\ \beta))\ (\tau{::}\uparrow\mathbf{0}))\ \ \leftrightarrow \Box((\lambda\beta.\ (\alpha\ \beta))\ \tau)\rfloor$ **by** *simp*
lemma $\lfloor\forall\,\alpha.\ ((\lambda\beta.\ \Box(\alpha\ \beta))\ (\tau{::}\langle\mathbf{0}\rangle))\ \ \leftrightarrow \Box((\lambda\beta.\ (\alpha\ \beta))\ \tau)\rfloor$ **by** *simp*
lemma $\lfloor\forall\,\alpha.\ ((\lambda\beta.\ \Box(\alpha\ \beta))\ (\tau{::}\uparrow\langle\mathbf{0}\rangle)) \leftrightarrow \Box((\lambda\beta.\ (\alpha\ \beta))\ \tau)\rfloor$ **by** *simp*

De re is not equivalent to *de dicto* for relativized (intensional) terms:

lemma $\lfloor\forall\,\alpha.\ ((\lambda\beta.\ \Box(\alpha\ \beta))\ \downharpoonleft(\tau{::}\uparrow\mathbf{0})) \leftrightarrow \Box(\ (\lambda\beta.\ (\alpha\ \beta))\ \downharpoonleft\tau)\rfloor$
nitpick[*card 't=2*, *card i=2*] **oops** — countersatisfiable
lemma $\lfloor\forall\,\alpha.\ ((\lambda\beta.\ \Box(\alpha\ \beta))\ \downarrow(\tau{::}\uparrow\langle\mathbf{0}\rangle)) \leftrightarrow \Box(\ (\lambda\beta.\ (\alpha\ \beta))\ \downarrow\tau)\rfloor$
nitpick[*card 't=1*, *card i=2*] **oops** — countersatisfiable

Proposition 9.6 - Equivalences between *de dicto* and *de re*:

abbreviation *deDictoEquDeRe*$::\uparrow\langle\uparrow\mathbf{0}\rangle$
where *deDictoEquDeRe* $\tau \equiv \forall\,\alpha.\ ((\lambda\beta.\ \Box(\alpha\ \beta))\ \downharpoonleft\tau) \leftrightarrow \Box((\lambda\beta.\ (\alpha\ \beta))\ \downharpoonleft\tau)$
abbreviation *deDictoImplDeRe*$::\uparrow\langle\uparrow\mathbf{0}\rangle$
where *deDictoImplDeRe* $\tau \equiv \forall\,\alpha.\ \Box((\lambda\beta.\ (\alpha\ \beta))\ \downharpoonleft\tau) \rightarrow ((\lambda\beta.\ \Box(\alpha\ \beta))\ \downharpoonleft\tau)$
abbreviation *deReImplDeDicto*$::\uparrow\langle\uparrow\mathbf{0}\rangle$
where *deReImplDeDicto* $\tau \equiv \forall\,\alpha.\ ((\lambda\beta.\ \Box(\alpha\ \beta))\ \downharpoonleft\tau) \rightarrow \Box((\lambda\beta.\ (\alpha\ \beta))\ \downharpoonleft\tau)$

abbreviation *deDictoEquDeRe-pred*$::('t{\Rightarrow}io){\Rightarrow}io$
where *deDictoEquDeRe-pred* $\tau \equiv \forall\,\alpha.\ ((\lambda\beta.\ \Box(\alpha\ \beta))\ \downarrow\tau) \leftrightarrow \Box((\lambda\beta.\ (\alpha\ \beta))\ \downarrow\tau)$
abbreviation *deDictoImplDeRe-pred*$::('t{\Rightarrow}io){\Rightarrow}io$
where *deDictoImplDeRe-pred* $\tau \equiv \forall\,\alpha.\ \Box((\lambda\beta.\ (\alpha\ \beta))\ \downarrow\tau) \rightarrow ((\lambda\beta.\ \Box(\alpha\ \beta))\ \downarrow\tau)$

abbreviation *deReImplDeDicto-pred*::$('t \Rightarrow io) \Rightarrow io$
where *deReImplDeDicto-pred* $\tau \equiv \forall\, \alpha.\ ((\lambda\beta.\ \Box(\alpha\ \beta))\ {\downarrow}\tau) \rightarrow \Box((\lambda\beta.\ (\alpha\ \beta))\ {\downarrow}\tau)$

3.2.4 Rigidity (Subsection 3)

Rigidity for intensional individuals:

abbreviation *rigidIndiv*::$\uparrow\langle\uparrow\mathbf{0}\rangle$ **where**
rigidIndiv $\tau \equiv (\lambda\beta.\ \Box((\lambda z.\ \beta \approx z)\ \lfloor\tau))\ \lfloor\tau$

Rigidity for intensional predicates:

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

Proposition 9.8 - We can prove it using local consequence (global consequence follows directly).

lemma $\lfloor$*rigidIndiv* $(\tau{::}\uparrow\mathbf{0}) \rightarrow$ *deReImplDeDicto* $\tau\rfloor$ **by** *simp*
lemma $\lfloor$*deReImplDeDicto* $(\tau{::}\uparrow\mathbf{0}) \rightarrow$ *rigidIndiv* $\tau\rfloor$ **by** *auto*
lemma $\lfloor$*rigidPred* $(\tau{::}\uparrow\langle\mathbf{0}\rangle) \rightarrow$ *deReImplDeDicto-pred* $\tau\rfloor$ **by** *simp*
lemma $\lfloor$*deReImplDeDicto-pred* $(\tau{::}\uparrow\langle\mathbf{0}\rangle) \rightarrow$ *rigidPred* $\tau\rfloor$ **by** *auto*

3.2.5 Stability Conditions (Subsection 4)

axiomatization where
S5: *equivalence aRel* — We use the Sahlqvist correspondence for improved performance

Definition 9.10 - Stability:

abbreviation *stabilityA*::$('t \Rightarrow io) \Rightarrow io$ **where** *stabilityA* $\tau \equiv \forall\, \alpha.\ (\tau\ \alpha) \rightarrow \Box(\tau\ \alpha)$
abbreviation *stabilityB*::$('t \Rightarrow io) \Rightarrow io$ **where** *stabilityB* $\tau \equiv \forall\, \alpha.\ \Diamond(\tau\ \alpha) \rightarrow (\tau\ \alpha)$

Proposition 9.10 - Note it is valid only for global consequence.

lemma $\lfloor$*stabilityA* $(\tau{::}\uparrow\langle\mathbf{0}\rangle)\rfloor \longrightarrow \lfloor$*stabilityB* $\tau\rfloor$ **using** *S5* **by** *blast*
lemma $\lfloor$*stabilityA* $(\tau{::}\uparrow\langle\mathbf{0}\rangle) \rightarrow$ *stabilityB* $\tau\rfloor$
nitpick[*card 't=1*, *card i=2*] **oops** — countersatisfiable for local consequence

lemma $\lfloor$*stabilityB* $(\tau{::}\uparrow\langle\mathbf{0}\rangle)\rfloor \longrightarrow \lfloor$*stabilityA* $\tau\rfloor$ **using** *S5* **by** *blast*
lemma $\lfloor$*stabilityB* $(\tau{::}\uparrow\langle\mathbf{0}\rangle) \rightarrow$ *stabilityA* $\tau\rfloor$
nitpick[*card 't=1*, *card i=2*] **oops** — countersatisfiable for local consequence

Theorem 9.11 - Note that we can prove even local consequence.

theorem $\lfloor$*rigidPred* $(\tau{::}\uparrow\langle\mathbf{0}\rangle) \leftrightarrow$ (*stabilityA* $\tau \wedge$ *stabilityB* τ)$\rfloor$ **by** *meson*
theorem $\lfloor$*rigidPred* $(\tau{::}\uparrow\langle\uparrow\mathbf{0}\rangle) \leftrightarrow$ (*stabilityA* $\tau \wedge$ *stabilityB* τ)$\rfloor$ **by** *meson*
theorem $\lfloor$*rigidPred* $(\tau{::}\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle) \leftrightarrow$ (*stabilityA* $\tau \wedge$ *stabilityB* τ)$\rfloor$ **by** *meson*

4 Gödel's Argument, Formally (Chapter 11)

"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." [8] p. 138.

4.1 Part I - God's Existence is Possible

We divide Gödel's Argument as presented in the book 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 may encompass some criticism of the notion of property entailment used by Gödel throughout the argument.

4.1.1 General Definitions

abbreviation $\mathit{existencePredicate}{::}\uparrow\langle\mathbf{0}\rangle$ $(E!)$
where $E!\ x \ \equiv \lambda w.\ (\exists^E y.\ y \approx x)\ w$ — existence predicate in the object-language

lemma $E!\ x\ w \longleftrightarrow \mathit{existsAt}\ x\ w$
by *simp* — safety check: correctly matches its meta-logical counterpart

consts $\mathit{positiveProperty}{::}\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle$ $(\mathcal{P})$ — Positiveness/Perfection

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

abbreviation $\mathit{God}{::}\uparrow\langle\mathbf{0}\rangle$ (G) **where** $G \equiv (\lambda x.\ \forall Y.\ \mathcal{P}\ Y \rightarrow\ Y\ x)$
abbreviation $\mathit{God\text{-}star}{::}\uparrow\langle\mathbf{0}\rangle$ $(G*)$ **where** $G* \equiv (\lambda x.\ \forall Y.\ \mathcal{P}\ Y \leftrightarrow\ Y\ x)$

Definitions needed to formalise *A3*:

abbreviation $\mathit{appliesToPositiveProps}{::}\uparrow\langle\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle\rangle$ (pos) **where**
$\mathit{pos}\ Z \equiv\ \forall X.\ Z\ X \rightarrow \mathcal{P}\ X$
abbreviation $\mathit{intersectionOf}{::}\uparrow\langle\uparrow\langle\mathbf{0}\rangle,\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle\rangle$ $(\mathit{intersec})$ **where**
$\mathit{intersec}\ X\ Z \equiv\ \Box(\forall x.(X\ x \leftrightarrow (\forall Y.\ (Z\ Y) \rightarrow (Y\ x))))$ — quantifier is possibilist

abbreviation $\mathit{Entailment}{::}\uparrow\langle\uparrow\langle\mathbf{0}\rangle,\uparrow\langle\mathbf{0}\rangle\rangle$ (**infix** $\Rrightarrow$ *60*) **where**
$X \Rrightarrow\ Y \equiv\ \Box(\forall^E z.\ X\ z \rightarrow\ Y\ z)$

4.1.2 Axioms

axiomatization where
A1a: $\lfloor\forall X.\ \mathcal{P}\ (\rightharpoondown X) \rightarrow \neg(\mathcal{P}\ X)\ \rfloor$ **and** — Axiom 11.3A
A1b: $\lfloor\forall X.\ \neg(\mathcal{P}\ X) \rightarrow \mathcal{P}\ (\rightharpoondown X)\rfloor$ **and** — Axiom 11.3B
A2: $\lfloor\forall X\ Y.\ (\mathcal{P}\ X \wedge (X \Rrightarrow Y)) \rightarrow \mathcal{P}\ Y\rfloor$ **and** — Axiom 11.5
A3: $\lfloor\forall Z\ X.\ (pos\ Z \wedge intersec\ X\ Z) \rightarrow \mathcal{P}\ X\rfloor$ — Axiom 11.10

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

lemma $\lfloor D\rfloor$ **using** *A1a A1b A2* **by** *blast* — axioms already imply D axiom
lemma $\lfloor D\rfloor$ **using** *A1a A3* **by** *metis*

4.1.3 Theorems

lemma $\lfloor\exists X.\ \mathcal{P}\ X\rfloor$ **using** *A1b* **by** *auto*
lemma $\lfloor\exists X.\ \mathcal{P}\ X \wedge\ \Diamond\exists^E\ X\rfloor$ **using** *A1a A1b A2* **by** *metis*

Being self-identical is a positive property:

lemma $\lfloor(\exists X.\ \mathcal{P}\ X \wedge\ \Diamond\exists^E\ X) \rightarrow \mathcal{P}\ (\lambda x\ w.\ x = x)\rfloor$ **using** *A2* **by** *fastforce*

Proposition 11.6

lemma $\lfloor(\exists X.\ \mathcal{P}\ X) \rightarrow \mathcal{P}\ (\lambda x\ w.\ x = x)\rfloor$ **using** *A2* **by** *fastforce*

lemma $\lfloor\mathcal{P}\ (\lambda x\ w.\ x = x)\rfloor$ **using** *A1b A2* **by** *blast*
lemma $\lfloor\mathcal{P}\ (\lambda x\ w.\ x = x)\rfloor$ **using** *A3* **by** *metis*

Being non-self-identical is a negative property:

lemma $\lfloor(\exists X.\ \mathcal{P}\ X\ \wedge \Diamond\exists^E\ X) \rightarrow\ \mathcal{P}\ (\rightharpoondown (\lambda x\ w.\ \neg x = x))\rfloor$
using *A2* **by** *fastforce*

lemma $\lfloor(\exists X.\ \mathcal{P}\ X) \rightarrow\ \mathcal{P}\ (\rightharpoondown (\lambda x\ w.\ \neg x = x))\rfloor$ **using** *A2* **by** *fastforce*
lemma $\lfloor(\exists X.\ \mathcal{P}\ X) \rightarrow\ \mathcal{P}\ (\rightharpoondown (\lambda x\ w.\ \neg x = x))\rfloor$ **using** *A3* **by** *metis*

Proposition 11.7

lemma $\lfloor(\exists X.\ \mathcal{P}\ X) \rightarrow \neg\mathcal{P}\ ((\lambda x\ w.\ \neg x = x))\rfloor$ **using** *A1a A2* **by** *blast*
lemma $\lfloor\neg\mathcal{P}\ (\lambda x\ w.\ \neg x = x)\rfloor$ **using** *A1a A2* **by** *blast*

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

theorem *T1*: $\lfloor\forall X.\ \mathcal{P}\ X \rightarrow \Diamond\exists^E\ X\rfloor$ **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*: $\lfloor\forall x.\ G\ x \leftrightarrow G{*}\ x\rfloor$ **using** *A1b* **by** *force*

Proposition 11.15 - Possibilist existence of *God*∗ directly implies *A1b*:

lemma $\lfloor\exists\ G{*} \rightarrow (\forall X.\ \neg(\mathcal{P}\ X) \rightarrow \mathcal{P}\ (\rightharpoondown X))\rfloor$ **by** *meson*

Proposition 11.16 - *A3* implies $P(G)$ (local consequence):

lemma *A3implT2-local*: $\lfloor(\forall\, Z\, X.\ (pos\ Z \wedge intersec\ X\ Z) \rightarrow \mathcal{P}\ X) \rightarrow \mathcal{P}\ G\rfloor$
proof −
  {
  fix *w*
  have *1*: *pos* $\mathcal{P}$ *w* **by** *simp*
  have *2*: *intersec G* $\mathcal{P}$ *w* **by** *simp*
  {
    assume $(\forall\, Z\, X.\ (pos\ Z \wedge intersec\ X\ Z) \rightarrow \mathcal{P}\ X)\ w$
    hence $(\forall\, X.\ ((pos\ \mathcal{P}) \wedge (intersec\ X\ \mathcal{P})) \rightarrow \mathcal{P}\ X)\ w$ **by** (*rule allE*)
    hence $(((pos\ \mathcal{P}) \wedge (intersec\ G\ \mathcal{P})) \rightarrow \mathcal{P}\ G)\ w$ **by** (*rule allE*)
    hence *3*: $((pos\ \mathcal{P} \wedge intersec\ G\ \mathcal{P})\ w) \longrightarrow \mathcal{P}\ G\ w$ **by** *simp*
    hence *4*: $((pos\ \mathcal{P}) \wedge (intersec\ G\ \mathcal{P}))\ w$ **using** *1 2* **by** *simp*
    from *3 4* **have** $\mathcal{P}\ G\ w$ **by** (*rule mp*)
  }
  hence $(\forall\, Z\, X.\ (pos\ Z \wedge intersec\ X\ Z) \rightarrow \mathcal{P}\ X)\ w \ \longrightarrow \mathcal{P}\ G\ w$ **by** (*rule impI*)
  }
  thus *?thesis* **by** (*rule allI*)
qed

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

lemma *A3implT2-global*: $\lfloor\forall\, Z\, X.\ (pos\ Z \wedge intersec\ X\ Z) \rightarrow \mathcal{P}\ X\rfloor \longrightarrow \lfloor\mathcal{P}\ G\rfloor$
  using *A3implT2-local* **by** *smt*

God is a positive property. Note that this theorem can be axiomatized directly (as noted by Dana Scott). We will do so for the second part.

theorem *T2*: $\lfloor\mathcal{P}\ G\rfloor$ **using** *A3implT2-global A3* **by** *simp*

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

theorem *T3*: $\lfloor\Diamond\exists^{E}\ G\rfloor$ **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, among others, an essentialist premise and the S5 axioms. A more detailed analysis of these rather philosophical issues is foreseen as follow-up work.

4.2.1 General Definitions

abbreviation *existencePredicate*::$\uparrow\langle\mathbf{0}\rangle$ $(E!)$ **where**
  $E!\ x\ \equiv (\lambda w.\ (\exists^{E} y.\ y{\approx}x)\ w)$
consts *positiveProperty*::$\uparrow\langle\uparrow\langle\mathbf{0}\rangle\rangle$ $(\mathcal{P})$

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

abbreviation *God-star*::$\uparrow\langle\mathbf{0}\rangle$ ($G*$) **where**
$G* \equiv (\lambda x.\ \forall\, Y.\ \mathcal{P}\ Y \leftrightarrow\ Y\ x)$
abbreviation *Entailment*::$\uparrow\langle\uparrow\langle\mathbf{0}\rangle,\uparrow\langle\mathbf{0}\rangle\rangle$ (**infix** $\Rrightarrow$ *60*) **where**
$X \Rrightarrow\ Y \equiv\ \Box(\forall^{E} z.\ X\ z \rightarrow\ Y\ z)$

4.2.2 Axioms 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:$\lfloor\forall\, X.\ \mathcal{P}\ (\rightharpoondown X) \rightarrow \neg(\mathcal{P}\ X)\ \rfloor$ **and** — Axiom 11.3A
A1b:$\lfloor\forall\, X.\ \neg(\mathcal{P}\ X) \rightarrow \mathcal{P}\ (\rightharpoondown X)\rfloor$ **and** — Axiom 11.3B
A2: $\lfloor\forall\, X\ Y.\ (\mathcal{P}\ X \wedge (X \Rrightarrow\ Y)) \rightarrow \mathcal{P}\ Y\rfloor$ **and** — Axiom 11.5
T2: $\lfloor\mathcal{P}\ G\rfloor$ — Proposition 11.16

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

4.2.3 Useful Results from Part I

lemma *GodDefsAreEquivalent*: $\lfloor\forall\, x.\ G\ x \leftrightarrow\ G*\ x\rfloor$ **using** *A1b* **by** *fastforce*

theorem *T1*: $\lfloor\forall\, X.\ \mathcal{P}\ X \rightarrow \Diamond\exists^{E}\ X\rfloor$
using *A1a A2* **by** *blast* — Positive properties are possibly instantiated
theorem *T3*: $\lfloor\Diamond\exists^{E}\ G\rfloor$ **using** *T1 T2* **by** *simp* — God exists possibly

4.2.4 Axioms for Part II

$\mathcal{P}$ satisfies so-called stability conditions (p. 124). This means it designates rigidly (an essentialist assumption).

axiomatization where
A4a: $\lfloor\forall\, X.\ \mathcal{P}\ X \rightarrow \Box(\mathcal{P}\ X)\rfloor$ — Axiom 11.11
lemma *A4b*: $\lfloor\forall\, X.\ \neg(\mathcal{P}\ X) \rightarrow \Box\neg(\mathcal{P}\ X)\rfloor$ **using** *A1a A1b A4a* **by** *blast*

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

4.2.5 Theorems

abbreviation *essenceOf*::$\uparrow\langle\uparrow\langle\mathbf{0}\rangle,\mathbf{0}\rangle$ ($\mathcal{E}$) **where**
$\mathcal{E}\ Y\ x \equiv (Y\ x) \wedge (\forall\, Z.\ Z\ x \rightarrow\ Y \Rrightarrow\ Z)$
abbreviation *beingIdenticalTo*::$\mathbf{0}\Rightarrow\uparrow\langle\mathbf{0}\rangle$ (*id*) **where**
$id\ x\ \equiv (\lambda y.\ y \approx x)$ — note that *id* is a rigid predicate

Theorem 11.20 - Informal Proposition 5

theorem *GodIsEssential*: $\lfloor\forall\, x.\ G\ x \rightarrow (\mathcal{E}\ G\ x)\rfloor$ **using** *A1b A4a* **by** *metis*

Theorem 11.21

theorem $\lfloor\forall\, x.\ G*\ x \rightarrow (\mathcal{E}\ G*\ x)\rfloor$ **using** *A4a* **by** *meson*

Theorem 11.22 - Something can have only one essence:

theorem $\lfloor \forall X\ Y\ z.\ (\mathcal{E}\ X\ z \wedge \mathcal{E}\ Y\ z) \rightarrow (X \Rrightarrow Y)\rfloor$ **by** *meson*

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

theorem *EssencesCharacterizeCompletely*: $\lfloor \forall X\ y.\ \mathcal{E}\ X\ y \rightarrow (X \Rrightarrow (id\ y))\rfloor$
proof (*rule ccontr*)
  assume $\neg\ \lfloor \forall X\ y.\ \mathcal{E}\ X\ y \rightarrow (X \Rrightarrow (id\ y))\rfloor$
  hence $\exists w.\ \neg((\ \forall X\ y.\ \mathcal{E}\ X\ y \rightarrow X \Rrightarrow id\ y)\ w)$ **by** *simp*
  then obtain w **where** $\neg((\ \forall X\ y.\ \mathcal{E}\ X\ y \rightarrow X \Rrightarrow id\ y)\ w)$ ..
  hence $(\exists X\ y.\ \mathcal{E}\ X\ y \wedge \neg(X \Rrightarrow id\ y))\ w$ **by** *simp*
  hence $\exists X\ y.\ \mathcal{E}\ X\ y\ w \wedge (\neg(X \Rrightarrow id\ y))\ w$ **by** *simp*
  then obtain P **where** $\exists y.\ \mathcal{E}\ P\ y\ w \wedge (\neg(P \Rrightarrow id\ y))\ w$ ..
  then obtain a **where** *1*: $\mathcal{E}\ P\ a\ w \wedge (\neg(P \Rrightarrow id\ a))\ w$ ..
  hence *2*: $\mathcal{E}\ P\ a\ w$ **by** (*rule conjunct1*)
  from *1* **have** $(\neg(P \Rrightarrow id\ a))\ w$ **by** (*rule conjunct2*)
  hence $\exists x.\ \exists z.\ w\ r\ x \wedge\ existsAt\ z\ x \wedge P\ z\ x \wedge \neg(a = z)$ **by** *blast*
  then obtain $w1$ **where** $\exists z.\ w\ r\ w1 \wedge\ existsAt\ z\ w1 \wedge P\ z\ w1 \wedge \neg(a = z)$ ..
  then obtain b **where** *3*: $w\ r\ w1 \wedge\ existsAt\ b\ w1 \wedge P\ b\ w1 \wedge \neg(a = b)$ ..
  hence $w\ r\ w1$ **by** *simp*
  from *3* **have** $existsAt\ b\ w1$ **by** *simp*
  from *3* **have** $P\ b\ w1$ **by** *simp*
  from *3* **have** *4*: $\neg(a = b)$ **by** *simp*
  from *2* **have** $P\ a\ w$ **by** *simp*
  from *2* **have** $\forall Y.\ Y\ a\ w \longrightarrow ((P \Rrightarrow Y)\ w)$ **by** *auto*
  hence $(\neg(id\ b))\ a\ w \longrightarrow (P \Rrightarrow (\neg(id\ b)))\ w$ **by** (*rule allE*)
  hence $\neg(\neg(id\ b))\ a\ w \vee ((P \Rrightarrow (\neg(id\ b)))\ w)$ **by** *blast*
  then show *False* **proof**
    assume $\neg(\neg(id\ b))\ a\ w$
    hence $a = b$ **by** *simp*
    thus *False* **using** *4* **by** *auto*
    next
    assume $((P \Rrightarrow (\neg(id\ b)))\ w)$
    hence $\forall x.\ \forall z.\ (w\ r\ x \wedge existsAt\ z\ x \wedge P\ z\ x) \longrightarrow (\neg(id\ b))\ z\ x$ **by** *blast*
    hence $\forall z.\ (w\ r\ w1 \wedge existsAt\ z\ w1 \wedge P\ z\ w1) \longrightarrow (\neg(id\ b))\ z\ w1$
      by (*rule allE*)
    hence $(w\ r\ w1 \wedge existsAt\ b\ w1 \wedge P\ b\ w1) \longrightarrow (\neg(id\ b))\ b\ w1$ **by** (*rule allE*)
    hence $\neg(w\ r\ w1 \wedge existsAt\ b\ w1 \wedge P\ b\ w1) \vee (\neg(id\ b))\ b\ w1$ **by** *simp*
    hence $(\neg(id\ b))\ b\ w$ **using** *3* **by** *simp*
    hence $\neg(b{=}b)$ **by** *simp*
    thus *False* **by** *simp*
  qed
qed

Definition 11.24 - Necessary Existence (Informal Definition 6):

abbreviation *necessaryExistencePred*::$\uparrow\langle\mathbf{0}\rangle$ (*NE*)
  where $NE\ x\ \equiv (\lambda w.\ (\forall Y.\ \ \mathcal{E}\ Y\ x \rightarrow \Box\exists^{E}\ Y)\ w)$

Axiom 11.25 (Informal Axiom 5)

axiomatization where

A5: $\lfloor \mathcal{P}\ NE \rfloor$

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*: $\lfloor \exists\ G \rightarrow\ \Box\exists^{E}\ G \rfloor$
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 $\forall Y.\ (\mathcal{E}\ Y\ g\ w) \longrightarrow (\Box\exists^{E}\ Y)\ w$ **by** *simp*
    hence *2*: $(\mathcal{E}\ G\ g\ w) \longrightarrow (\Box\exists^{E}\ G)\ w$ **by** (*rule allE*)
    have $(\forall x.\ G\ x \rightarrow (\mathcal{E}\ G\ x))\ w$ **using** *GodIsEssential*
      by (*rule allE*) — GodIsEssential follows from Axioms 11.11 and 11.3B
    hence $(G\ g \rightarrow (\mathcal{E}\ G\ g))\ w$ **by** (*rule allE*)
    hence $G\ g\ w \longrightarrow \mathcal{E}\ G\ g\ w$ **by** *simp*
    from *this 1* **have** *3*: $\mathcal{E}\ G\ g\ w$ **by** (*rule mp*)
    from *2 3* **have** $(\Box\exists^{E}\ G)\ w$ **by** (*rule mp*)
  }
  hence $(\exists x.\ G\ x\ w) \longrightarrow (\Box\exists^{E}\ G)\ w$ **by** (*rule impI*)
  hence $((\exists x.\ G\ x) \rightarrow\ \Box\exists^{E}\ G)\ w$ **by** *simp*
}
thus *?thesis* **by** (*rule allI*)
qed

Modal Collapse is countersatisfiable until we introduce S5 axioms:

lemma $\lfloor \forall \Phi.(\Phi \rightarrow (\Box\ \Phi)) \rfloor$ **nitpick oops**

Axiomatizing semantic frame conditions for different modal logics. All axioms together imply an S5 logic:

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 ($\lfloor \forall \Phi.(\Phi \rightarrow (\Box\ \Phi)) \rfloor$) is actually valid (see proof below)

Some useful rules:

lemma *modal-distr*: $\lfloor \Box(\varphi \rightarrow \psi) \rfloor \Longrightarrow \lfloor (\Diamond\varphi \rightarrow \Diamond\psi) \rfloor$ **by** *blast*
lemma *modal-trans*: $(\lfloor \varphi \rightarrow \psi \rfloor \wedge \lfloor \psi \rightarrow \chi \rfloor) \Longrightarrow \lfloor \varphi \rightarrow \chi \rfloor$ **by** *simp*

Theorem 11.27 - Informal Proposition 8

theorem *possExistenceImpliesNecEx*: $\lfloor\Diamond\exists\ G \rightarrow \Box\exists^E\ G\rfloor$ — local consequence
proof –
  have $\lfloor\exists\ G \rightarrow \Box\exists^E\ G\rfloor$ **using** *GodExistenceImpliesNecExistence*
    by *simp* — follows from Axioms 11.11, 11.25 and 11.3B
  hence $\lfloor\Box(\exists\ G \rightarrow \Box\exists^E\ G)\rfloor$ **using** *NEC* **by** *simp*
  hence *1*: $\lfloor\Diamond\exists\ G \rightarrow \Diamond\Box\exists^E\ G\rfloor$ **by** (*rule modal-distr*)
  have *2*: $\lfloor\Diamond\Box\exists^E\ G \rightarrow \Box\exists^E\ G\rfloor$ **using** *symm tran* **by** *metis*
  from *1 2* **have** $\lfloor\Diamond\exists\ G \rightarrow \Diamond\Box\exists^E\ G\rfloor \wedge \lfloor\Diamond\Box\exists^E\ G \rightarrow \Box\exists^E\ G\rfloor$ **by** *simp*
  thus *?thesis* **by** (*rule modal-trans*)
qed

lemma *T4*: $\lfloor\Diamond\exists\ G\rfloor \longrightarrow \lfloor\Box\exists^E\ G\rfloor$ **using** *possExistenceImpliesNecEx*
    by *simp* — global consequence

Corollary 11.28 - Necessary (actualist) existence of God (for both definitions):

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

4.2.6 Monotheism

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

theorem *Monotheism-LeibnizEq*: $\lfloor\forall x.\ G\ x \rightarrow (\forall y.\ G\ y \rightarrow (x\ \approx^L\ y))\rfloor$
  using *GodDefsAreEquivalent* **by** *simp*

Monotheism for normal models is trickier. We need to consider some previous results (p. 162):

lemma *GodExistenceIsValid*: $\lfloor\exists^E\ G\rfloor$ **using** *GodNecExists refl*
  by *auto* — Note that we hadn't needed frame reflexivity until now

Proposition 11.29

theorem *Monotheism-normalModel*: $\lfloor\exists x.\forall y.\ G\ y \leftrightarrow x \approx y\rfloor$
proof –
{
  fix w
  have $\lfloor\exists^E\ G\rfloor$ **using** *GodExistenceIsValid* **by** *simp* — follows from corollary 11.28

  hence $(\exists^E\ G)\ w$ **by** (*rule allE*)
  then obtain g **where** *1*: *existsAt* $g\ w \wedge G\ g\ w$ **..**
  hence *2*: $\mathcal{E}\ G\ g\ w$ **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 *3*: $\mathcal{E}\ G\ y\ w$ **using** *GodIsEssential* **by** *blast*

have $(\mathcal{E}\ G\ y \rightarrow (G \Rrightarrow id\ y))\ w$ **using** *EssencesCharacterizeCompletely*
by *simp* — follows from theorem 11.23
hence $\mathcal{E}\ G\ y\ w \longrightarrow ((G \Rrightarrow id\ y)\ w)$ **by** *simp*
from *this 3* **have** $(G \Rrightarrow id\ y)\ w$ **by** (*rule mp*)
hence $(\Box(\forall^E z.\ G\ z \rightarrow z \approx y))\ w$ **by** *simp*
hence $\forall x.\ w\ r\ x \longrightarrow ((\forall z.\ (existsAt\ z\ x \wedge G\ z\ x) \longrightarrow z = y))$ **by** *auto*
hence $w\ r\ w \longrightarrow ((\forall z.\ (existsAt\ z\ w \wedge G\ z\ w) \longrightarrow z = y))$ **by** (*rule allE*)
hence $\forall z.\ (w\ r\ w \wedge existsAt\ z\ w \wedge G\ z\ w) \longrightarrow z = y$ **by** *auto*
hence *4*: $(w\ r\ w \wedge existsAt\ g\ w \wedge G\ g\ w) \longrightarrow g = y$ **by** (*rule allE*)
have $w\ r\ w$ **using** *refl*
by *simp* — note that we rely explicitly on frame reflexivity (Axiom M)
hence $w\ r\ w \wedge (existsAt\ g\ w \wedge G\ g\ w)$ **using** *1* **by** (*rule conjI*)
from *4 this* **have** $g = y$ **by** (*rule mp*)
thus $(g \approx y)\ w$ **by** *simp*
next
assume $(g \approx y)\ w$
from *this 2* **have** $\mathcal{E}\ G\ y\ w$ **by** *simp*
thus $G\ y\ w$ **by** (*rule conjunct1*)
qed
}
hence $\forall y.\ G\ y\ w \longleftrightarrow (g \approx y)\ w$ **by** (*rule allI*)
hence $\exists x.\ (\forall y.\ G\ y\ w \longleftrightarrow (x \approx y)\ w)$ **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*: $\lfloor \forall x.\ G\ x \rightarrow E!\ x \rfloor$
using *GodExistenceIsValid Monotheism-normalModel* **by** *metis*

4.2.7 Positive Properties are Necessarily Instantiated

lemma *PosPropertiesNecExist*: $\lfloor \forall Y.\ \mathcal{P}\ Y \rightarrow \Box\exists^E\ Y \rfloor$ **using** *GodNecExists A4a*
by *meson* — Proposition 11.31: follows from corollary 11.28 and axiom A4a

4.2.8 Objections and Criticism

lemma *useful*: $(\forall x.\ \varphi\ x \longrightarrow \psi) \Longrightarrow ((\exists x.\ \varphi\ x) \longrightarrow \psi)$ **by** *simp*

After introducing the S5 axioms Modal Collapse becomes valid (pp. 163-4):

lemma *ModalCollapse*: $\lfloor \forall \Phi.(\Phi \rightarrow (\Box\ \Phi)) \rfloor$
proof –
{
fix w
{
fix Q
have $(\forall x.\ G\ x \rightarrow (\mathcal{E}\ G\ x))\ w$ **using** *GodIsEssential*
by (*rule allE*) — follows from Axioms 11.11 and 11.3B

hence $\forall x.\ G\ x\ w \longrightarrow \mathcal{E}\ G\ x\ w$ **by** *simp*
hence $\forall x.\ G\ x\ w \longrightarrow (\forall Z.\ Z\ x \rightarrow \Box(\forall^E z.\ G\ z \rightarrow Z\ z))\ w$ **by** *force*
hence $\forall x.\ G\ x\ w \longrightarrow ((\lambda y.\ Q)\ x \rightarrow \Box(\forall^E z.\ G\ z \rightarrow (\lambda y.\ Q)\ z))\ w$ **by** *force*
hence $\forall x.\ G\ x\ w \longrightarrow (Q \rightarrow \Box(\forall^E z.\ G\ z \rightarrow Q))\ w$ **by** *simp*
hence *1*: $(\exists x.\ G\ x\ w) \longrightarrow ((Q \rightarrow \Box(\forall^E z.\ G\ z \rightarrow Q))\ w)$ **by** (*rule useful*)
have $\exists x.\ G\ x\ w$ **using** *GodExistenceIsValid* **by** *auto*
from *1 this* **have** $(Q \rightarrow \Box(\forall^E z.\ G\ z \rightarrow Q))\ w$ **by** (*rule mp*)
hence $(Q \rightarrow \Box((\exists^E z.\ G\ z) \rightarrow Q))\ w$ **using** *useful* **by** *blast*
hence $(Q \rightarrow (\Box(\exists^E z.\ G\ z) \rightarrow \Box Q))\ w$ **by** *simp*
hence $(Q \rightarrow \Box Q)\ w$ **using** *GodNecExists* **by** *simp*
}
hence $(\forall \Phi.\ \Phi \rightarrow \Box\ \Phi)\ w$ **by** (*rule allI*)
}
thus *?thesis* **by** (*rule allI*)
qed

5 Fitting's Solution

In this section we tackle Fitting's solution to the objections raised in his previous 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 commited 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 chosen were those better suiting argument's validity.

5.1 Implicit Extensionality Assumptions

Since Isabelle/HOL is extensional, extensionality principles are valid directly out of the box:

lemma *EXT*: $\forall \alpha{::}\langle \mathbf{0}\rangle.\ \forall \beta{::}\langle \mathbf{0}\rangle.\ (\forall \gamma{::}\mathbf{0}.\ (\alpha\ \gamma \longleftrightarrow \beta\ \gamma)) \longrightarrow (\alpha = \beta)$ **by** *auto*
lemma *EXT-set*: $\forall \alpha{::}\langle\langle \mathbf{0}\rangle\rangle.\ \forall \beta{::}\langle\langle \mathbf{0}\rangle\rangle.\ (\forall \gamma{::}\langle \mathbf{0}\rangle.\ (\alpha\ \gamma \longleftrightarrow \beta\ \gamma)) \longrightarrow (\alpha = \beta)$ **by** *auto*
lemma *EXT-intensional*: $\lfloor(\lambda x.\ ((\lambda y.\ x{\approx}y)\ \downharpoonleft(\alpha{::}\uparrow\mathbf{0}\ )))\ \ \downharpoonleft(\beta{::}\uparrow\mathbf{0})\ \rfloor \longrightarrow \alpha = \beta$ **by** *auto*
lemma *EXT-int-pred*: $\lfloor(\lambda x.\ ((\lambda y.\ x{\approx}y)\ \downarrow(\alpha{::}\uparrow\langle\mathbf{0}\rangle)))\ \downarrow(\beta{::}\uparrow\langle\mathbf{0}\rangle)\rfloor \longrightarrow \alpha = \beta$ **using** *ext* **by** *metis*

5.2 General Definitions

The following technical definitions are needed only for type correctness. They are used to convert extensional objects into rigid intensional ones.

abbreviation *trivialExpansion*::*bool*⇒*io* ((|-|)) **where** $(|\varphi|) \equiv \lambda w.\ \varphi$
abbreviation *existencePredicate*::$\uparrow\langle\mathbf{0}\rangle$ ($E!$) **where**
$E!\ x \ \equiv (\lambda w.\ (\exists^E y.\ y{\approx}x)\ w)$

consts *positiveProperty*::$\uparrow\langle\langle\mathbf{0}\rangle\rangle$ ($\mathcal{P}$)
abbreviation *God*::$\uparrow\langle\mathbf{0}\rangle$ (G) **where** $G \equiv (\lambda x.\ \forall Y.\ \mathcal{P}\ Y \rightarrow (|Y\ x|))$
abbreviation *God-star*::$\uparrow\langle\mathbf{0}\rangle$ ($G*$) **where** $G* \equiv (\lambda x.\ \forall Y.\ \mathcal{P}\ Y \leftrightarrow (|Y\ x|))$

abbreviation *Entailment*::$\uparrow\langle\langle\mathbf{0}\rangle,\langle\mathbf{0}\rangle\rangle$ (**infix** $\Rrightarrow$ *60*) **where**
$X \Rrightarrow Y \equiv \Box(\forall^E z.\ (|X\ z|) \rightarrow (|Y\ z|))$

5.3 Part I - God's Existence is Possible

axiomatization where
A1a:$\lfloor\forall X.\ \mathcal{P}\ (\neg X) \rightarrow \neg(\mathcal{P}\ X)\ \rfloor$ **and** — Axiom 11.3A
A1b:$\lfloor\forall X.\ \neg(\mathcal{P}\ X) \rightarrow \mathcal{P}\ (\neg X)\rfloor$ **and** — Axiom 11.3B
A2: $\lfloor\forall X\ Y.\ (\mathcal{P}\ X \wedge (X \Rrightarrow Y)) \rightarrow \mathcal{P}\ Y\rfloor$ **and** — Axiom 11.5
T2: $\lfloor\mathcal{P} \downarrow G\rfloor$ — Proposition 11.16 (modified)

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

lemma *GodDefsAreEquivalent*: $\lfloor\forall x.\ G\ x \leftrightarrow G*\ x\rfloor$ **using** *A1b* **by** *fastforce*

T1 (Positive properties are possibly instantiated) can be formalised in two different ways:

theorem *T1a*: $\lfloor\forall X{::}\langle\mathbf{0}\rangle.\ \mathcal{P}\ X \rightarrow \Diamond(\exists^E z.\ (|X\ z|))\rfloor$
using *A1a A2* **by** *blast* — this is the one used in the book
theorem *T1b*: $\lfloor\forall X{::}\uparrow\langle\mathbf{0}\rangle.\ \mathcal{P} \downarrow X \rightarrow \Diamond(\exists^E z.\ X\ z)\rfloor$
nitpick oops — this one is also possible but not valid so we won't use it

Some interesting (non-) equivalences:

lemma $\lfloor\Box\exists^E\ (Q{::}\uparrow\langle\mathbf{0}\rangle) \leftrightarrow \Box(\exists^E \downarrow Q)\rfloor$ **by** *simp*
lemma $\lfloor\Box\exists^E\ (Q{::}\uparrow\langle\mathbf{0}\rangle) \leftrightarrow ((\lambda X.\ \Box\exists^E\ X)\ Q)\rfloor$ **by** *simp*
lemma $\lfloor\Box\exists^E\ (Q{::}\uparrow\langle\mathbf{0}\rangle) \leftrightarrow ((\lambda X.\ \Box\exists^E \downarrow X)\ Q)\rfloor$ **by** *simp*
lemma $\lfloor\Box\exists^E\ (Q{::}\uparrow\langle\mathbf{0}\rangle) \leftrightarrow ((\lambda X.\ \Box\exists^E\ X) \downarrow Q)\rfloor$ **nitpick oops** — not equivalent!

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

theorem *T3-deRe*: $\lfloor(\lambda X.\ \Diamond\exists^E\ X) \downarrow G\rfloor$ **using** *T1a T2* **by** *simp*
theorem *T3-deDicto*: $\lfloor\Diamond\exists^E \downarrow G\rfloor$ **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 (unless *T1b* were valid but it isn't)

lemma assumes *T1b*: $\lfloor\forall X.\ \mathcal{P}\downarrow X \rightarrow \Diamond(\exists^E z.\ X\ z)\rfloor$
shows *T3-deDicto*: $\lfloor\Diamond\exists^E \downarrow G\rfloor$ **using** *assms T2* **by** *simp*

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

In this variant $\mathcal{P}$ also designates rigidly.

axiomatization where
A4a: $\lfloor\forall X.\ \mathcal{P}\ X \rightarrow \Box(\mathcal{P}\ X)\rfloor$ — Axiom 11.11
lemma *A4b*: $\lfloor\forall X.\ \neg(\mathcal{P}\ X) \rightarrow \Box\neg(\mathcal{P}\ X)\rfloor$ **using** *A1a A1b A4a* **by** *blast*

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

abbreviation *essenceOf*::$\uparrow\langle\langle\mathbf{0}\rangle,\mathbf{0}\rangle$ ($\mathcal{E}$) **where**
$\mathcal{E}\ Y\ x \equiv (\!|Y\ x|\!) \wedge (\forall Z{::}\langle\mathbf{0}\rangle.\ (\!|Z\ x|\!) \rightarrow Y \Rrightarrow Z)$

Theorem 11.20 - Informal Proposition 5

theorem *GodIsEssential*: $\lfloor\forall x.\ G\ x \rightarrow ((\mathcal{E}\downarrow_1 G)\ x)\rfloor$ **using** *A1b* **by** *metis*

Theorem 11.21

theorem *God-starIsEssential*: $\lfloor\forall x.\ G{*}\ x \rightarrow ((\mathcal{E}\downarrow_1 G{*})\ x)\rfloor$ **by** *meson*

abbreviation *necExistencePred*:: $\uparrow\langle\mathbf{0}\rangle$ (*NE*) **where**
$NE\ x \equiv \lambda w.\ (\forall Y.\ \mathcal{E}\ Y\ x \rightarrow \Box(\exists^E z.\ (\!|Y\ z|\!)))\ w$

Informal Axiom 5

axiomatization where
A5: $\lfloor\mathcal{P}\downarrow NE\rfloor$

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

Reminder: We use the down-arrow notation because it is more explicit. See (non-) equivalences above.

lemma $\lfloor\exists\ G \leftrightarrow \exists\downarrow G\rfloor$ **by** *simp*
lemma $\lfloor\exists^E\ G \leftrightarrow \exists^E\downarrow G\rfloor$ **by** *simp*
lemma $\lfloor\Box\exists^E\ G \leftrightarrow \Box\exists^E\downarrow G\rfloor$ **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:

theorem *GodExImpliesNecEx-v1*: $\lfloor\exists\downarrow G \rightarrow \Box\exists^E\downarrow G\rfloor$
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*
  hence $\forall\, Y.\ (\mathcal{E}\ \ Y\ g\ w) \longrightarrow (\Box(\exists^E z.\ (\!|Y\ z|\!))) \ w$ **by** *simp*
  hence $(\mathcal{E}\ (\lambda x.\ G\ x\ w)\ g\ w) \longrightarrow (\Box(\exists^E z.\ (\!|(\lambda x.\ G\ x\ w)\ z|\!)))\ w$ **by** (*rule allE*)
  hence *2*: $((\mathcal{E}\ {\downarrow_1}G)\ g\ w) \longrightarrow (\Box(\exists^E\ G))\ w$ **using** *A4b* **by** *meson*
  have $(\forall\, x.\ G\ x \rightarrow ((\mathcal{E}\ {\downarrow_1}G)\ x))\ w$ **using** *GodIsEssential* **by** (*rule allE*)
  hence $(G\ g \rightarrow ((\mathcal{E}\ {\downarrow_1}G)\ g))\ w$ **by** (*rule allE*)
  hence $G\ g\ w \longrightarrow (\mathcal{E}\ {\downarrow_1}G)\ g\ w$ **by** *simp*
  from *this 1* **have** *3*: $(\mathcal{E}\ {\downarrow_1}G)\ g\ w$ **by** (*rule mp*)
  from *2 3* **have** $(\Box\exists^E\ G)\ w$ **by** (*rule mp*)
}
hence $(\exists\, x.\ G\ x\ w) \longrightarrow (\Box\exists^E\ G)\ w$ **by** (*rule impI*)
hence $((\exists\, x.\ G\ x) \rightarrow\ \Box\exists^E\ G)\ w$ **by** *simp*
}
thus *?thesis* **by** (*rule allI*)
qed

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

theorem *GodExImpliesNecEx-v2*: $\lfloor \exists\ {\downarrow}G \rightarrow ((\lambda X.\ \Box\exists^E\ X)\ {\downarrow}G)\rfloor$
  using *A4a GodExImpliesNecEx-v1* **by** *metis*

Compared to Goedel's argument, the following theorems can be proven in K logic (note that S5 no longer needed):

Theorem 11.27 - Informal Proposition 8

theorem *possExImpliesNecEx-v1*: $\lfloor \Diamond\exists\ {\downarrow}G \rightarrow \Box\exists^E\ {\downarrow}G\rfloor$
  using *GodExImpliesNecEx-v1 T3-deRe* **by** *metis*
theorem *possExImpliesNecEx-v2*: $\lfloor(\lambda X.\ \Diamond\exists^E\ X)\ {\downarrow}G \rightarrow ((\lambda X.\ \Box\exists^E\ X)\ {\downarrow}G)\rfloor$
  using *GodExImpliesNecEx-v2* **by** *blast*

Corollaries:

lemma *T4-v1*: $\lfloor\Diamond\exists\ {\downarrow}G\rfloor \longrightarrow \lfloor\Box\exists^E\ {\downarrow}G\rfloor$
  using *possExImpliesNecEx-v1* **by** *simp*
lemma *T4-v2*: $\lfloor(\lambda X.\ \Diamond\exists^E\ X)\ {\downarrow}G\rfloor \longrightarrow \lfloor(\lambda X.\ \Box\exists^E\ X)\ {\downarrow}G\rfloor$
  using *possExImpliesNecEx-v2* **by** *simp*

5.5 Conclusion - Necessary Existence of God

Version I - *de dicto* reading:

lemma *GodNecExists-v1*: $\lfloor\Box\exists^E\ {\downarrow}G\rfloor$
  using *GodExImpliesNecEx-v1 T3-deRe* **by** *fastforce* — Corollary 11.28
lemma *God-starNecExists-v1*: $\lfloor\Box\exists^E\ {\downarrow}G{*}\rfloor$
  using *GodNecExists-v1 GodDefsAreEquivalent* **by** *simp*
lemma $\lfloor\Box(\lambda X.\ \exists^E\ X)\ {\downarrow}G{*}\rfloor$
  using *God-starNecExists-v1* **by** *simp* — *de dicto* shown here explicitly

Version II - *de re* reading:

lemma *GodNecExists-v2*: $\lfloor(\lambda X.\ \Box\exists^E\ X)\ {\downarrow}G\rfloor$
using *T3-deRe T4-v2* **by** *blast*
lemma *God-starNecExists-v2*: $\lfloor(\lambda X.\ \Box\exists^E\ X)\ {\downarrow}G{*}\rfloor$
using *GodNecExists-v2 GodDefsAreEquivalent* **by** *simp*

5.6 Modal Collapse

Modal Collapse is countersatisfiable even in *S5*. Note that countermodels with a cardinality of one for the domain of ground-level objects are found by Nitpick (the countermodel shown in the book has cardinality of two).

lemma $\lfloor\forall\,\Phi.(\Phi \rightarrow (\Box\ \Phi))\rfloor$
nitpick[*card 't=1*, *card i=2*] **oops** — countermodel found in *K*

axiomatization where
S5: *equivalence aRel* — assume accesibility relation is an equivalence

lemma $\lfloor\forall\,\Phi.(\Phi \rightarrow (\Box\ \Phi))\rfloor$
nitpick[*card 't=1*, *card i=2*] **oops** — countermodel found in *S5*

References

[1] A. Anderson and M. Gettings. Gödel ontological proof revisited. In *Gödel'96: Logical Foundations of Mathematics, Computer Science, and Physics: Lecture Notes in Logic 6*, pages 167–172. Springer, 1996.

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

[3] C. Benzmüller. Automating quantified conditional logics in HOL. In F. Rossi, editor, *23rd International Joint Conference on Artificial Intelligence (IJCAI-13)*, pages 746–753, Beijing, China, 2013.

[4] C. Benzmüller. Universal reasoning, rational argumentation and human-machine interaction. *arXiv, http://arxiv.org/abs/1703.09620*, 2017.

[5] C. Benzmüller and B. W. Paleo. Automating Gödel's ontological proof of God's existence with higher-order automated theorem provers. In T. Schaub, G. Friedrich, and B. O'Sullivan, editors, *ECAI 2014*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 93 – 98. IOS Press, 2014.

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

[7] C. Benzmüller and B. Woltzenlogel Paleo. An object-logic explanation for the inconsistency in gödel's ontological theory (extended abstract). In M. Helmert and F. Wotawa, editors, *KI 2016: Advances in Artificial Intelligence, Proceedings*, LNCS, Berlin, Germany, 2016. Springer.

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

[9] K. Gödel. *Appx.A: Notes in Kurt Gödel's Hand*, pages 144–145. In [12], 2004.

[10] D. Scott. *Appx.B: Notes in Dana Scott's Hand*, pages 145–146. In [12], 2004.

[11] J. Sobel. Gödel's ontological proof. In *On Being and Saying. Essays for Richard Cartwright*, pages 241–261. MIT Press, 1987.

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