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

David Fuenmayor, Christoph Benzmüller

March 13, 2017

Contents

1 Embedding of Higher-Order Modal Logic with Intensional Types

1.1 Declarations

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

Aliases for common unary predicate types:

type-synonym $ie = (i \Rightarrow O)$ $(\uparrow O)$
type-synonym $se = (O \Rightarrow bool)$ $(\langle O \rangle)$
type-synonym $ise = (O \Rightarrow io)$ $(\uparrow\langle O \rangle)$
type-synonym $sie = (\uparrow O \Rightarrow bool)$ $(\langle \uparrow O \rangle)$
type-synonym $isie = (\uparrow O \Rightarrow io)$ $(\uparrow\langle \uparrow O \rangle)$
type-synonym $sise = (\uparrow\langle O \rangle \Rightarrow bool)$ $(\langle \uparrow\langle O \rangle \rangle)$
type-synonym $isise = (\uparrow\langle O \rangle \Rightarrow io)$ $(\uparrow\langle \uparrow\langle O \rangle \rangle)$
type-synonym $sisise = (\uparrow\langle \uparrow\langle O \rangle \rangle \Rightarrow bool)$ $(\langle \uparrow\langle \uparrow\langle O \rangle \rangle \rangle)$
type-synonym $isisise = (\uparrow\langle \uparrow\langle O \rangle \rangle \Rightarrow io)$ $(\uparrow\langle \uparrow\langle \uparrow\langle O \rangle \rangle \rangle)$
type-synonym $sse = \langle O \rangle \Rightarrow bool$ $(\langle \langle O \rangle \rangle)$
type-synonym $isse = \langle O \rangle \Rightarrow io$ $(\uparrow\langle \langle O \rangle \rangle)$

Aliases for common binary relation types:

type-synonym $see = (O \Rightarrow O \Rightarrow bool)$ $(\langle O,O \rangle)$
type-synonym $isee = (O \Rightarrow O \Rightarrow io)$ $(\uparrow\langle O,O \rangle)$
type-synonym $sieie = (\uparrow O \Rightarrow \uparrow O \Rightarrow bool)$ $(\langle \uparrow O,\uparrow O \rangle)$
type-synonym $isieie = (\uparrow O \Rightarrow \uparrow O \Rightarrow io)$ $(\uparrow\langle \uparrow O,\uparrow O \rangle)$

type-synonym *ssese* = $(\langle O\rangle \Rightarrow \langle O\rangle \Rightarrow bool)$ $(\langle\langle O\rangle,\langle O\rangle\rangle)$
type-synonym *issese* = $(\langle O\rangle \Rightarrow \langle O\rangle \Rightarrow io)$ $(\uparrow\langle\langle O\rangle,\langle O\rangle\rangle)$
type-synonym *ssee* = $(\langle O\rangle \Rightarrow O \Rightarrow bool)$ $(\langle\langle O\rangle,O\rangle)$
type-synonym *issee* = $(\langle O\rangle \Rightarrow O \Rightarrow io)$ $(\uparrow\langle\langle O\rangle,O\rangle)$
type-synonym *isisee* = $(\uparrow\langle O\rangle \Rightarrow O \Rightarrow io)$ $(\uparrow\langle\uparrow\langle O\rangle,O\rangle)$
type-synonym *isiseise* = $(\uparrow\langle O\rangle \Rightarrow \uparrow\langle O\rangle \Rightarrow io)$ $(\uparrow\langle\uparrow\langle O\rangle,\uparrow\langle O\rangle\rangle)$
type-synonym *isiseisise*= $(\uparrow\langle O\rangle \Rightarrow \uparrow\langle\uparrow\langle O\rangle\rangle \Rightarrow io)$ $(\uparrow\langle\uparrow\langle O\rangle,\uparrow\langle\uparrow\langle O\rangle\rangle\rangle)$

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

1.2 Definition of logical operators in HOL

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 *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 $\varphi \bigoplus \psi \equiv \lambda w.\ (\varphi\ w) \oplus (\psi\ w)$
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 *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**$\forall$ $[8]9$)
where $\forall\, x.\ \varphi(x) \equiv \forall\,\varphi$
abbreviation *mexistsB* $:: ('t \Rightarrow io) \Rightarrow io$ (**binder**$\exists$ $[8]9$)
where $\exists\, x.\ \varphi(x) \equiv \exists\,\varphi$

1.3 Definition of actualist quantifiers

No polymorphic types are used since actualist quantification only makes sense for individuals.

(Meta-logical) existence predicate for restricting domains of quantification:

consts $Exists::\uparrow\langle O\rangle$ $(existsAt)$

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

Binder notation for quantifiers:

abbreviation *mforallActB* :: $\uparrow\langle\uparrow\langle O\rangle\rangle$ (**binder**$\forall^E$[*8*]*9*)
  where $\forall^E x.\ \varphi(x) \equiv \forall^E \varphi$
abbreviation *mexistsActB* :: $\uparrow\langle\uparrow\langle O\rangle\rangle$ (**binder**$\exists^E$[*8*]*9*)
  where $\exists^E x.\ \varphi(x) \equiv \exists^E \varphi$

1.4 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 $\Diamond\varphi \equiv \lambda w.\exists v.\ (w\ r\ v) \wedge (\varphi\ v)$

1.5 Definition of the ExtensionOf operator

Embedding in HOL of (world-dependent) atomic formulas whose first argument is relativized.

ExtensionOf operator is therefore embedded in HOL as a binary operator. Depending on the types involved we need to define this operator differently to ensure type correctness.

(a) phi takes an (intensional) individual concept as argument:

abbreviation *mextIndiv* :: $\uparrow\langle O\rangle \Rightarrow \uparrow O \Rightarrow io$ (**infix** $\rfloor$ *60*)
  where $\varphi \rfloor c \equiv \lambda w.\ \varphi\ (c\ w)\ w$

(b) phi 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) phi 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) phi 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 $\varphi \downarrow_1 P \equiv \lambda z.\ \lambda w.\ \varphi\ (\lambda x.\ P\ x\ w)\ z\ w$

1.6 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* :: $\uparrow\langle\uparrow O, \uparrow O\rangle$ (**infixr**$\approx^C$ *52*) — equality for individual concepts
  where $x \approx^C y \equiv \lambda w.\ \forall v.\ (x\ v) = (y\ v)$
abbreviation *meqL* :: $\uparrow\langle O, O\rangle$ (**infixr**$\approx^L$ *52*) — Leibniz Equ. for individuals
  where $x \approx^L y \equiv \forall \varphi.\ \varphi(x) \rightarrow \varphi(y)$

1.7 Miscelaneous

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

1.8 Meta-logical predicates

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

1.9 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

Instances of the Barcan and converse Barcan Formulas are satisintensionalfied 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*

... but not 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

1.10 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. We recommend to use the semantic constraints for improved performance.

2 Examples in book

2.1 Chapter 7 - Modal Logic - Syntax and Semantics

2.1.1 beta/eta-redex Considerations (page 94)

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

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

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

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

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

lemma $\lfloor((\lambda\alpha.\ \Box\varphi\ \alpha)\ \downharpoonleft(\tau::\uparrow O)) \leftrightarrow (\Box\varphi\ \downharpoonleft\tau)\rfloor$ **nitpick oops** — countersatisfiable

lemma $\lfloor((\lambda\alpha.\ \Diamond\varphi\ \alpha)\ \downharpoonleft(\tau::\uparrow O)) \leftrightarrow (\Diamond\varphi\ \downharpoonleft\tau)\rfloor$ **nitpick oops** — countersatisfiable

Example 7.13 page 96:

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

with other types for P:

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

Example 7.14 page 98:

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

with other types for P:

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

Example 7.15 page 99:

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

for other types:

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

Example 7.16 page 100:

lemma $\lfloor\Box(P\ \downharpoonleft(c::\uparrow O)) \rightarrow (\exists\, x::O.\ \Box(P\ x))\rfloor$ **nitpick**[*card 't=2*, *card i=2*] **oops** — countersatisfiable (using only two worlds!)

Example 7.17 page 101:

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

2.1.2 Exercises page 101

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

Exercise 7.3:

lemma $\lfloor \Diamond\exists\,(P{::}\uparrow\langle O\rangle) \to (\exists\, X{::}\uparrow O.\ \Diamond(P\downharpoonleft X))\rfloor$ **by** *auto*
lemma $\lfloor \Diamond\exists\,(P{::}\uparrow\langle\uparrow\langle O\rangle\rangle) \to (\exists\, X{::}\uparrow\langle O\rangle.\ \Diamond(P\Downarrow X))\rfloor$ **nitpick**[*card 't=1*, *card i=2*] **oops**

Exercise 7.4:

lemma $\lfloor \Diamond(\exists\, x{::}O.\ \ (\lambda Y.\ \ Y\ x)\ \Downarrow(P{::}\uparrow\langle O\rangle)) \ \to\ (\exists\, x.\ \ (\lambda Y.\ \ \Diamond(Y\ x))\ \Downarrow P)\rfloor$ **nitpick**[*card 't=1*, *card i=2*] **oops** — countersatisfiable

For Exercise 7.5 see Example 7.17 above.

2.2 Chapter 9 - Miscellaneous Matters

2.2.1 (1.1) Equality

Example 9.1:

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

2.2.2 (1.2) Extensionality

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

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

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

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

2.2.3 (2) De re de dicto

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

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

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

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

Proposition 9.6 - Equivalences between de dicto and de re:

abbreviation *deDictoEquDeRe*::$\uparrow\langle\uparrow O\rangle$ **where** *deDictoEquDeRe* $\tau \equiv \forall\,\alpha.\ ((\lambda\beta.\ \Box(\alpha\ \beta))\ \downharpoonleft\tau) \leftrightarrow \Box((\lambda\beta.\ (\alpha\ \beta))\ \downharpoonleft\tau)$
abbreviation *deDictoImpliesDeRe*::$\uparrow\langle\uparrow O\rangle$ **where** *deDictoImpliesDeRe* $\tau \equiv \forall\,\alpha.\ \Box((\lambda\beta.\ (\alpha\ \beta))\ \downharpoonleft\tau) \rightarrow ((\lambda\beta.\ \Box(\alpha\ \beta))\ \downharpoonleft\tau)$
abbreviation *deReImpliesDeDicto*::$\uparrow\langle\uparrow O\rangle$ **where** *deReImpliesDeDicto* $\tau \equiv \forall\,\alpha.\ ((\lambda\beta.\ \Box(\alpha\ \beta))\ \downharpoonleft\tau) \rightarrow \Box((\lambda\beta.\ (\alpha\ \beta))\ \downharpoonleft\tau)$

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

The following are valid only when using global consequence:

(TODO: solvers need some help to find the proofs)

lemma $\lfloor$*deDictoImpliesDeRe* $(\tau{::}\uparrow O)\rfloor \longrightarrow \lfloor$*deReImpliesDeDicto* $\tau\rfloor$ **oops**
lemma $\lfloor$*deReImpliesDeDicto* $(\tau{::}\uparrow O)\rfloor \longrightarrow \lfloor$*deDictoImpliesDeRe* $\tau\rfloor$ **oops**
lemma $\lfloor$*deDictoImpliesDeRe-pred* $(\tau{::}\uparrow\langle O\rangle)\rfloor \longrightarrow \lfloor$*deReImpliesDeDicto-pred* $\tau\rfloor$ **oops**
lemma $\lfloor$*deReImpliesDeDicto-pred* $(\tau{::}\uparrow\langle O\rangle)\rfloor \longrightarrow \lfloor$*deDictoImpliesDeRe-pred* $\tau\rfloor$ **oops**

2.2.4 (3) Rigidity

Rigidity for intensional individuals:

abbreviation *rigidIndiv*::$\uparrow\langle\uparrow O\rangle$ **where**
rigidIndiv $\tau \equiv (\lambda\beta.\ \Box((\lambda z.\ \beta \approx z)\ \downharpoonleft\tau))\ \downharpoonleft\tau$

... and 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 $\lfloor rigidIndiv\ (\tau{::}\uparrow O) \rightarrow deReImpliesDeDicto\ \tau\rfloor$ **by** *simp*
lemma $\lfloor deReImpliesDeDicto\ (\tau{::}\uparrow O) \rightarrow rigidIndiv\ \tau\rfloor$ **by** *auto*
lemma $\lfloor rigidPred\ (\tau{::}\uparrow\langle O\rangle) \rightarrow deReImpliesDeDicto\text{-}pred\ \tau\rfloor$ **by** *simp*
lemma $\lfloor deReImpliesDeDicto\text{-}pred\ (\tau{::}\uparrow\langle O\rangle) \rightarrow rigidPred\ \tau\rfloor$ **by** *auto*

2.2.5 (4) Stability Conditions

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

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

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

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

3 Argument Part I - God's existence is possible

3.1 General definitions

abbreviation $existencePredicate{::}\uparrow\langle O\rangle\ (E!)$ **where** $E!\ x\ \equiv \lambda w.\ (\exists^{E} y.\ y{\approx}x)\ w$

Safety check. Existence predicate correctly matches its meta-logical counterpart:

lemma $E!\ x\ w \longleftrightarrow existsAt\ x\ w$ **by** *simp*

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

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

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

abbreviation *God-star*::$\uparrow\langle O\rangle$ $(G*)$ **where** $G* \equiv (\lambda x.\ \forall\, Y.\ \mathcal{P}\ Y \leftrightarrow\ Y\ x)$

Definitions needed to formalize A3:

abbreviation *appliesToPositiveProps*::$\uparrow\langle\uparrow\langle\uparrow\langle O\rangle\rangle\rangle$ (*pos*) **where** $pos\ Z \equiv\ \forall\, X.\ Z\ X \rightarrow \mathcal{P}\ X$
abbreviation *intersectionOf*::$\uparrow\langle\uparrow\langle O\rangle,\uparrow\langle\uparrow\langle O\rangle\rangle\rangle$ (*intersec*) **where** $intersec\ X\ Z \equiv \Box(\forall\, x.(X\ x \leftrightarrow (\forall\, Y.\ (Z\ Y) \rightarrow (Y\ x))))$
abbreviation *Entailment*::$\uparrow\langle\uparrow\langle O\rangle,\uparrow\langle O\rangle\rangle$ (**infix** $\Rrightarrow$ *60*) **where** $X \Rrightarrow Y \equiv \Box(\forall^E z.\ X\ z \rightarrow\ Y\ z)$

3.2 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** — Axioms are consistent

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

3.3 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 Scott's proposal of axiomatizing this (replacing A3):

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 Argument Part II - God's existence is necessary if possible

4.1 General definitions

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

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

4.2 Axioms from Part I

Following Scott's proposal we 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** — Axioms are consistent

4.3 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

4.4 Axioms for Part II

P satisfies so-called stability conditions (Page 124). This means P 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** — So far all axioms consistent

4.5 Theorems

abbreviation *essenceOf*::$\uparrow\langle\uparrow\langle O\rangle,O\rangle$ $(\mathcal{E})$ **where**
$\mathcal{E}\ Y\ x \equiv (Y\ x) \wedge (\forall Z.\ Z\ x \rightarrow Y \Rrightarrow Z)$
abbreviation *beingIdenticalTo*::$O{\Rightarrow}\uparrow\langle O\rangle$ (id) **where**

$id\ x\ \equiv (\lambda y.\ y{\approx}x)$ — 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*::$\uparrow\langle O\rangle$ (*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** — 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:

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

lemma *True* **nitpick**[*satisfy*] **oops** — Axioms still consistent

lemma *MC*: $\lfloor\forall\,\Phi.(\Phi \rightarrow (\Box\ \Phi))\rfloor$ **oops** — With S5 modal collapse 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* — which 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* — Using S5 semantic frame conditions
 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*

4.6 Going Further

After introducing the S5 axioms Nitpick cannot find a countermodel for Modal Collapse:

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

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 (Page 162):

lemma *GodExistenceIsValid*: $\lfloor\exists^{E}\ G\rfloor$ **using** *GodNecExists refl* **by** *auto* — useful lemma

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* — Theorem 11.20 follows from Axioms 11.11 and 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* — 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* — 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*

Proposition 11.31 - Positive properties are necessarily instantiated:

lemma *PosPropertiesNecExist*: $\lfloor\forall Y.\ \mathcal{P}\ Y \rightarrow \Box\exists^E\ Y\rfloor$ **using** *GodNecExists A4a* **by** *meson* — Corollary 11.28 and A4a needed for the proof

4.7 Objections and Criticism

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

Modal Collapse (Sobel's version as shown in Pages 163-164):

lemma *MC*: $\lfloor \forall \Phi.(\Phi \rightarrow (\Box\ \Phi)) \rfloor$
proof –
 {
 fix w
  {
  fix Q
   have $(\forall x.\ G\ x \rightarrow (\mathcal{E}\ \ G\ x))\ w$ **using** *GodIsEssential* **by** (*rule allE*) — GodIsEssential 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 $(Q \rightarrow \Box((\exists^E z.\ G\ z) \rightarrow Q))\ w$ **by** *force* — we can use rule 'useful' too
  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 Proof (pages 165-166)

5.1 Implicit extensionality assumptions in Isabelle/HOL

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

5.2 General Definitions

Following enables using extensional objects as if they were rigid intensional objects (needed only for type correctness):

abbreviation *trivialExpansion*::*bool*⇒*io* $(\llparenthesis\text{-}\rrparenthesis)$ **where** $\llparenthesis\varphi\rrparenthesis \equiv \lambda w.\ \varphi$

abbreviation *existencePredicate*::$\uparrow\langle O\rangle$ $(E!)$ **where**

$E!\ x \ \equiv (\lambda w.\ (\exists^E y.\ y{\approx}x)\ w)$

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

abbreviation *Entailment*::$\uparrow\langle\langle O\rangle,\langle O\rangle\rangle$ (**infix** $\Rrightarrow$ *60*) **where**
$X \Rrightarrow Y \equiv \Box(\forall^E z.\ (\!|X\ z|\!) \to (\!|Y\ z|\!))$

5.3 Part I - God's existence is possible

Following Scott's proposal we take T2 directly as an axiom.

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

lemma *True* **nitpick**[*satisfy*] **oops** — 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 formalized in two different ways:

theorem *T1a*: $\lfloor\forall X{::}\langle O\rangle.\ \mathcal{P}\ X \to \Diamond(\exists^E z.\ (\!|X\ z|\!))\rfloor$ **using** *A1a A2* **by** *blast* — The one used in the book
theorem *T1b*: $\lfloor\forall X{::}\uparrow\langle O\rangle.\ \mathcal{P}\ {\downarrow}X \to \Diamond(\exists^E z.\ X\ z)\rfloor$ **nitpick oops** — Since this one is not valid, we won't use it

Interesting (non-) equivalences:

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

T3 (God exists possibly) can be formalized 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

$T3_deReshouldbetheversionimpliedinthebook, becauseT3_deDictoisnotvalidunlessT1b(fromabove)we$

lemma assumes *T1b*: $\lfloor\forall X.\ \mathcal{P}\ {\downarrow}X \to \Diamond(\exists^E z.\ X\ z)\rfloor$
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

P satisfies so-called stability conditions (Page 124). This means P designates rigidly (an essentialist assumption!).

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

lemma *True* **nitpick**[*satisfy*] **oops** — So far all axioms consistent

abbreviation *essenceOf*::$\uparrow\langle\langle O\rangle,O\rangle$ ($\mathcal{E}$) **where**
 $\mathcal{E}\ Y\ x \equiv (\!|Y\ x|\!) \wedge (\forall Z{::}\langle O\rangle.\ (\!|Z\ x|\!) \to Y \Rrightarrow Z)$ — The one used in the book

Theorem 11.20 - Informal Proposition 5

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

Theorem 11.21

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

abbreviation *necExistencePred*:: $\uparrow\langle O\rangle$ (*NE*) **where**
 $NE\ x\ \equiv \lambda w.\ (\forall Y.\ \ \mathcal{E}\ Y\ x \to \Box(\exists^E z.\ (\!|Y\ z|\!)))\ w$ — the one used in the book

Informal Axiom 5

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

lemma *True* **nitpick**[*satisfy*] **oops** — 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 formalizing this theorem. Both of them are proved valid:

First version:

theorem *GodExImpliesNecEx-v1*: $\lfloor \exists\ \downarrow G \to\ \ \Box\exists^E\ \downarrow G\rfloor$
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 (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 (actualist) existence of God (Corollary 11.28)

Version I - De dicto reading:

lemma *GodNecExists-v1*: $\lfloor \Box\exists^{E}\ \downarrow G\rfloor$ **using** *GodExImpliesNecEx-v1 T3-deRe* **by** *fastforce*
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 reading shown 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 God-DefsAreEquivalent* **by** *simp*

5.6 Modal Collapse

Modal Collapse is countersatisfiable even in S5. Counterexamples with cardinality one for the domain of ground-level objects are found by Nitpick:

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

axiomatization where
S5: *equivalence aRel*

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