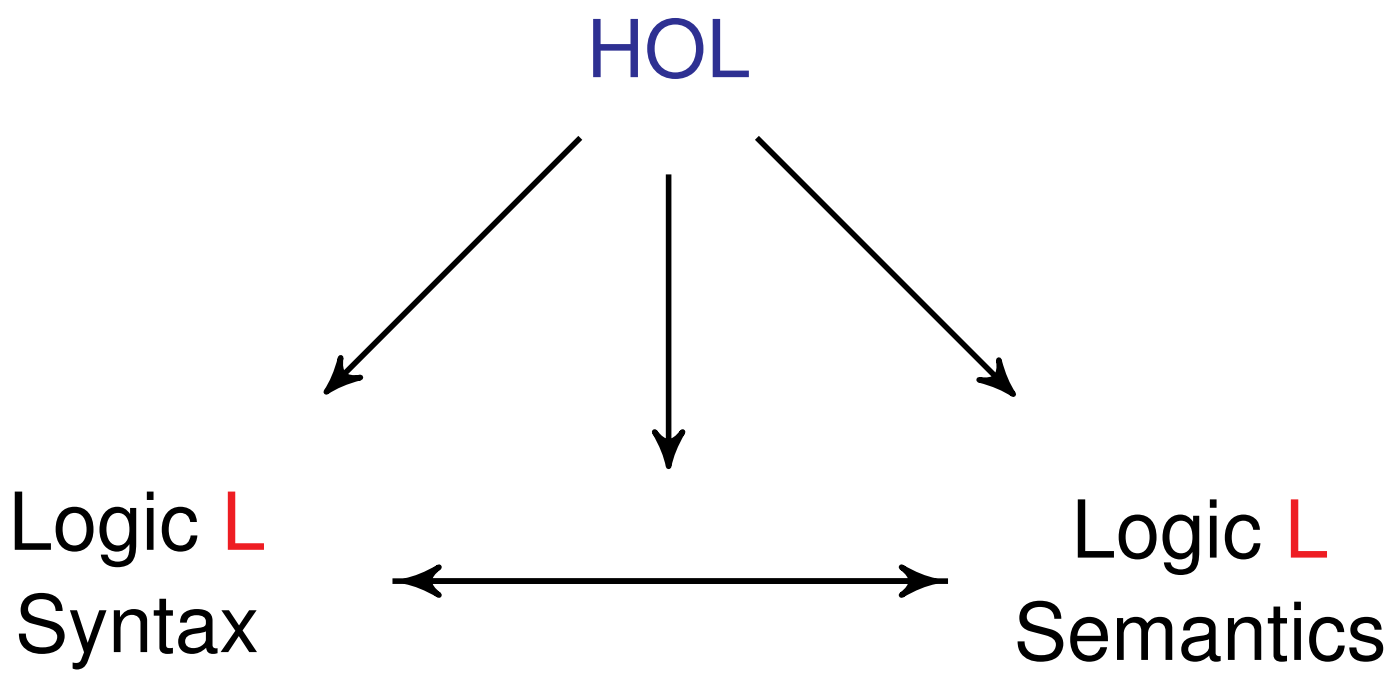
Uniform Proofs via Shallow Semantic Embeddings?

Christoph Benzmüller[1] — FU Berlin

Dagstuhl Seminar on “Universality of Proofs”, October 20, 2016

[1] Supported by DFG Heisenberg Fellowship BE 2501/9-1/2

HOL as a (quite) Universal Metalogic via Shallow Semantic Embeddings



Examples for L we have already studied:

Modal Logics, Temporal Logics, Conditional Logics, Intuitionistic Logics, Nominal Logics, Multivalued Logics, (Mathematical) Fuzzy Logics, Access Control Logics, Paraconsistent Logics, Free Logics, Dynamic Logics, ...

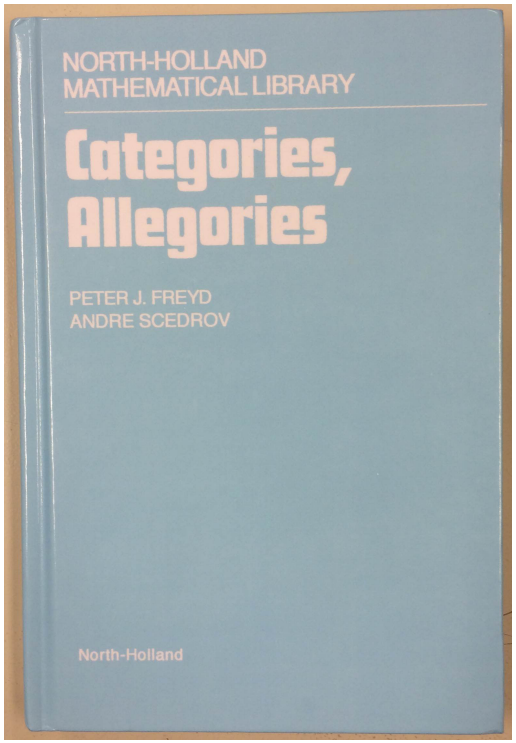
Works also for (first-order & higher-order) quantifiers

Application Example A: Computational Metaphysics

Leibniz: *Calculemus!*

- Objective: Analysis of Rational Arguments in Metaphysics
- Joint work with: Bruno Woltzenlogel Paleo (ANU)
- Main papers: ECAI'2014, IJCAI'2016
- Logics: Higher-Order Modal Logics
  - modalities: K, KB, S4, S5, ...
  - quantifiers: constant/varying/increasing/decreasing domains
  - (restricted comprehension schemes)
- Results: significant novel findings by ATPs

Application Example B: Theory Exploration — Category Theory



1.1. BASIC DEFINITIONS

The theory of CATEGORIES is given by two unary operations and a binary partial operation. In most contexts lower-case variables are used for the 'individuals' which are called *morphisms* or *maps*. The values of the operations are denoted and pronounced as:

$\Box x$ *the source of* x ,

$x\Box$ *the target of* x ,

xy *the composition of* x *and* y .

The axioms:

A1 xy *is defined iff* $x\Box = \Box y$,

A2a $(\Box x)\Box = \Box x$ *and* $\Box(x\Box) = x\Box$, A2b

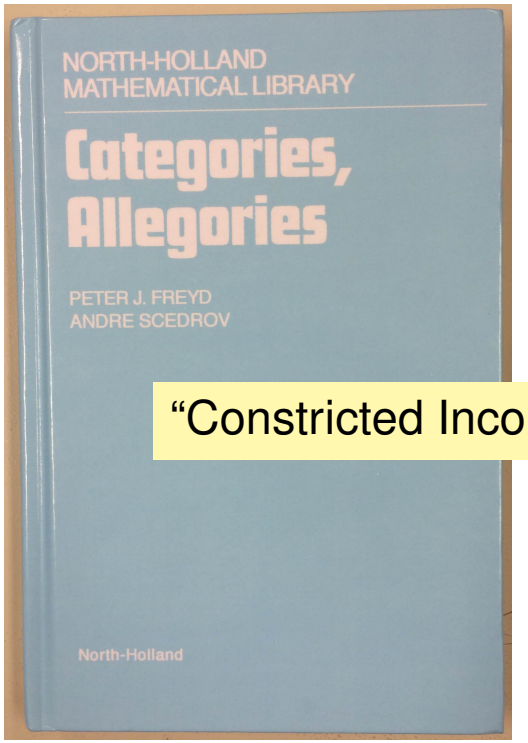
A3a $(\Box x)x = x$ *and* $x(x\Box) = x$, A3b

A4a $\Box(xy) = \Box(x(\Box y))$ *and* $(xy)\Box = ((x\Box)y)\Box$, A4b

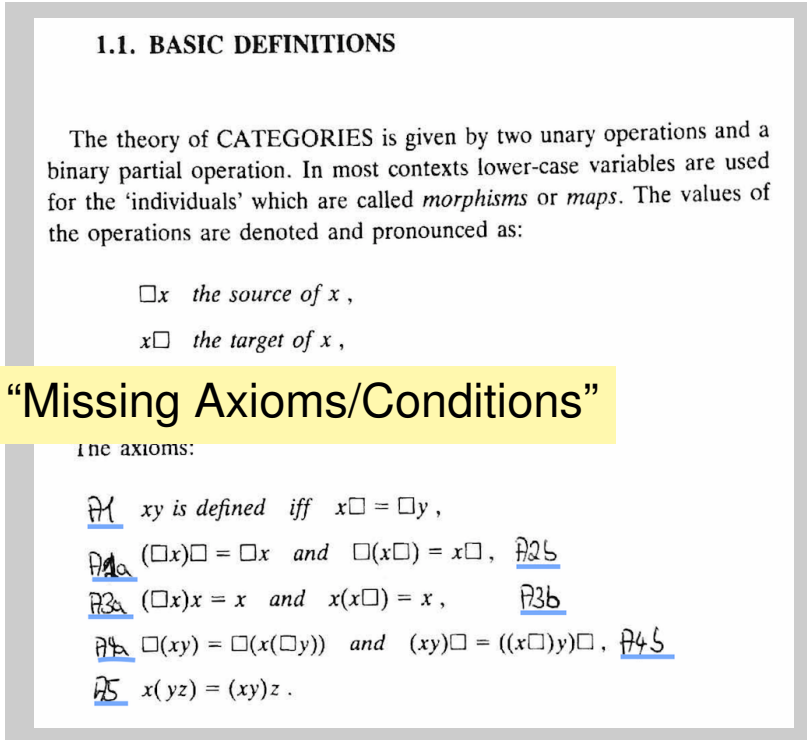
A5 $x(yz) = (xy)z$.

- Objective: Exploration and Analysis of Axiom Systems for Category Theory
- Joint work with: Dana Scott
- Main papers: ICMS'2016, arXiv'2016
- Logics: Free Logic
- Results:
  - development of 6 related (equivalent) axiom systems for category theory
  - from Monoids . . . via Scott's (1977) axiom system . . . to Freyd/Scedrov (1990)
  - for Freyd and Scedrov (1990) we revealed some flaws/issues

Application Example B: Theory Exploration — Category Theory



"Constricted Inconsistency" or "Missing Axioms/Conditions"



1.1. BASIC DEFINITIONS

The theory of CATEGORIES is given by two unary operations and a binary partial operation. In most contexts lower-case variables are used for the 'individuals' which are called *morphisms* or *maps*. The values of the operations are denoted and pronounced as:

$\Box x$ *the source of* x ,

$x\Box$ *the target of* x ,

The axioms:

A1 xy *is defined iff* $x\Box = \Box y$,

A1a $(\Box x)\Box = \Box x$ *and* $\Box(x\Box) = x\Box$, A2b

A3a $(\Box x)x = x$ *and* $x(x\Box) = x$, A3b

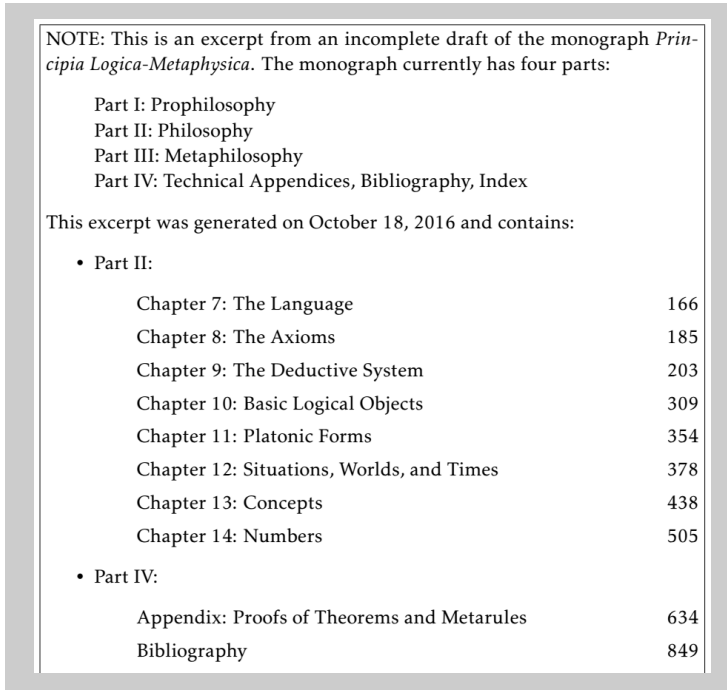
A4a $\Box(xy) = \Box(x(\Box y))$ *and* $(xy)\Box = ((x\Box)y)\Box$, A4b

A5 $x(yz) = (xy)z$.

- Objective: Exploration and Analysis of Axiom Systems for Category Theory
- Joint work with: Dana Scott
- Main papers: ICMS'2016, arXiv'2016
- Logics: Free Logic
- Results:
  - development of 6 related (equivalent) axiom systems for category theory
  - from Monoids . . . via Scott's (1977) axiom system . . . to Freyd/Scedrov (1990)
  - for Freyd and Scedrov (1990) we revealed some flaws/issues

Application Example C: Formalisation/Automation of Principia Metaphysica (Zalta)

–since ’83→



NOTE: This is an excerpt from an incomplete draft of the monograph *Principia Logica-Metaphysica*. The monograph currently has four parts:

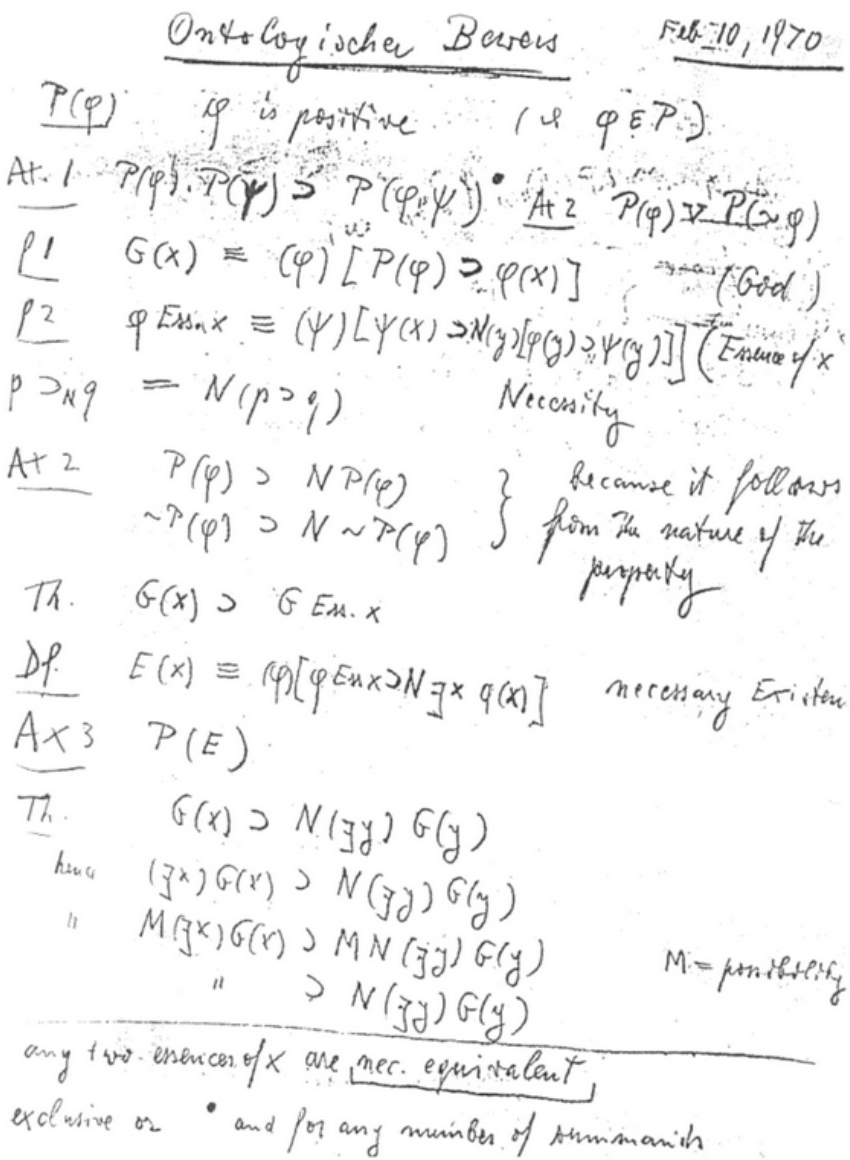
Part I: Prophilosophy
Part II: Philosophy
Part III: Metaphilosophy
Part IV: Technical Appendices, Bibliography, Index

This excerpt was generated on October 18, 2016 and contains:

Principia Logico-Metaphysica

- Objective: Formalisation/Automation of Principia Metaphysica
- Joint work with: Daniel Kirchner (student of mine), Ed Zalta (Stanford)
- Main papers: soon
- Logics: Hyper-Intensional Higher-Order Modal Logic
- Results:
  - first significant part of PM has been formalised (by David)
  - excellent degree of automation
  - minor issues detected

DEMO: Computational Metaphysics in Isabelle/HOL



Ontologischer Beweis Feb 10, 1970

P(φ) φ is positive (= φ ∈ P)

Ax. 1 P(φ) · P(ψ) ⊃ P(φ·ψ) • Ax 2 P(φ) ∨ P(¬φ)

D1 G(x) ≡ (φ)[P(φ) ⊃ φ(x)] (God)

D2 φ Ess. x ≡ (ψ)[ψ(x) ⊃ N(y)[φ(y) ⊃ ψ(y)]] (Essence of x)

p ⊃_N q = N(p ⊃ q) Necessity

Ax 2 P(φ) ⊃ N P(φ)
¬P(φ) ⊃ N ¬P(φ)
} because it follows from the nature of the property

Th. G(x) ⊃ G Ess. x

Df. E(x) ≡ (φ)[φ Ess. x ⊃_N ∃x φ(x)] necessary Existen

Ax 3 P(E)

Th. G(x) ⊃ N(∃y) G(y)

hence (∃x) G(x) ⊃ N(∃y) G(y)

" M(∃x) G(x) ⊃ MN(∃y) G(y)

" ⊃ N(∃y) G(y) M = possibility

any two essences of x are nec. equivalent

exclusive or • and for any number of summands

M(∃x) G(x) means (the system of) all pos. prop. is com-patible This is true because of:

Ax 4: P(φ) · φ ⊃_N ψ : ⊃ P(ψ) which imp

~~Axiom~~ { x = x is positive
~~end~~ { x ≠ x is negative

But if a system S of pos. propo. were in com
it would mean that the sum prop. s (which
is positive) would be x ≠ x

Positive means positive in the moral aesth
sense (independly of the accidental structure of
the world). Only then the ax. true. It m
also meant "attribution" as opposed to "privation
(or containg privation) - This interpret. implies proof

If φ positive: (x) N ¬φ(x) Otherwise φ(x) ⊃_N x ≠
hence x ≠ x positive not x = x by contrary Ax
or the exist. of pos. prop.

dwg

x i.e. the normal form in terms of elem. prop. contain
member without negation.

DEMO: Computational Metaphysics in Isabelle/HOL

Axiom A1 Either a property or its negation is positive, but not both: $\forall\phi[P(\neg\phi) \leftrightarrow \neg P(\phi)]$

Axiom A2 A property necessarily implied by a positive property is positive: $\forall\phi\forall\psi[(P(\phi) \wedge \Box\forall x[\phi(x) \rightarrow \psi(x)]) \rightarrow P(\psi)]$

Thm. T1 Positive properties are possibly exemplified: $\forall\phi[P(\phi) \rightarrow \Diamond\exists x\phi(x)]$

Def. D1 A *God-like* being possesses all positive properties: $G(x) \leftrightarrow \forall\phi[P(\phi) \rightarrow \phi(x)]$

Axiom A3 The property of being God-like is positive: $P(G)$

Cor. C Possibly, God exists: $\Diamond\exists xG(x)$

Axiom A4 Positive properties are necessarily positive: $\forall\phi[P(\phi) \rightarrow \Box P(\phi)]$

Def. D2 An *essence* of an individual is a property possessed by it and necessarily implying any of its properties: $\phi\ ess.\ x \leftrightarrow \phi(x) \wedge \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$

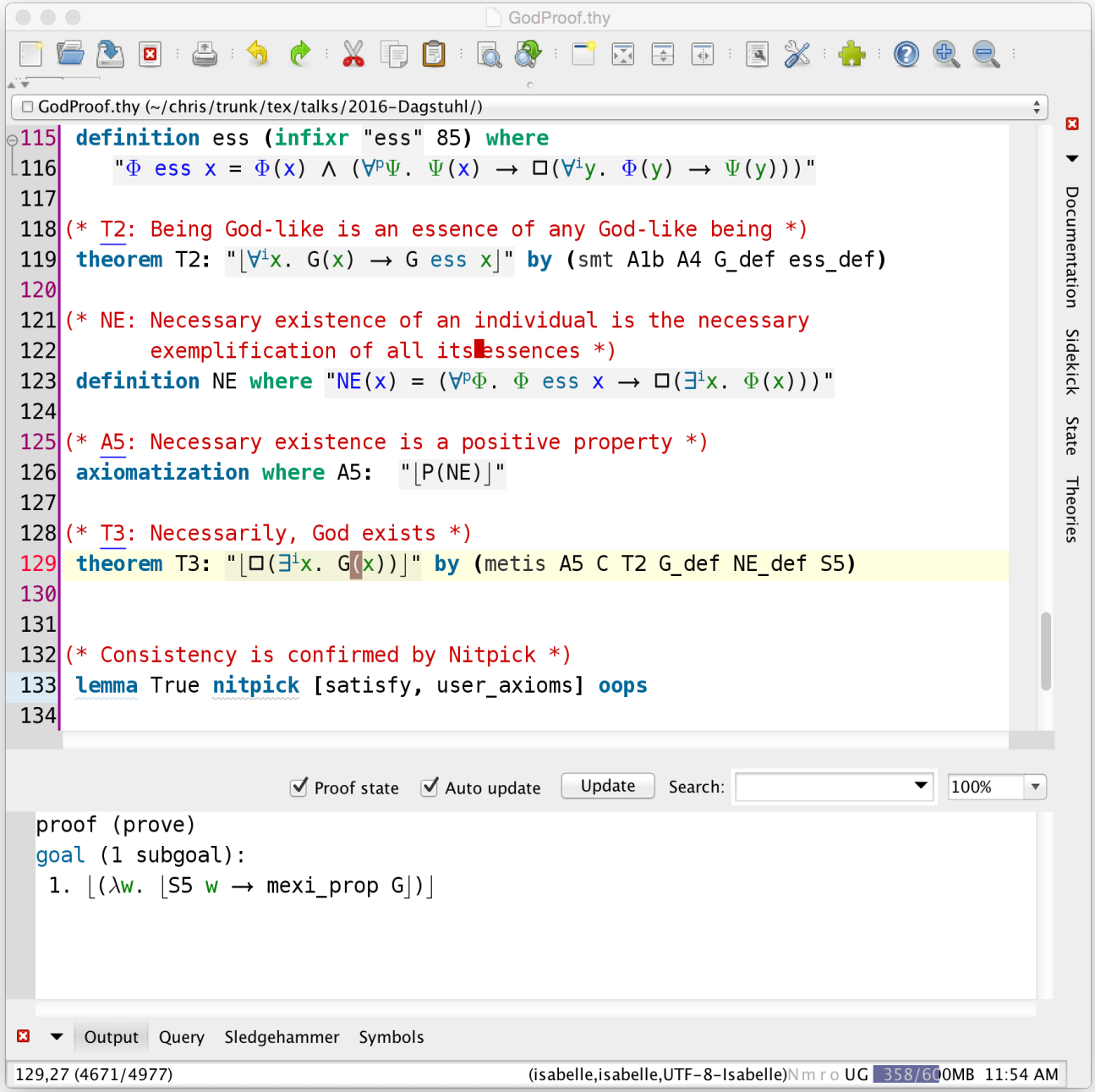
Thm. T2 Being God-like is an essence of any God-like being: $\forall x[G(x) \rightarrow G\ ess.\ x]$

Def. D3 *Necessary existence* of an individual is the necessary exemplification of all its essences: $NE(x) \leftrightarrow \forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y\phi(y)]$

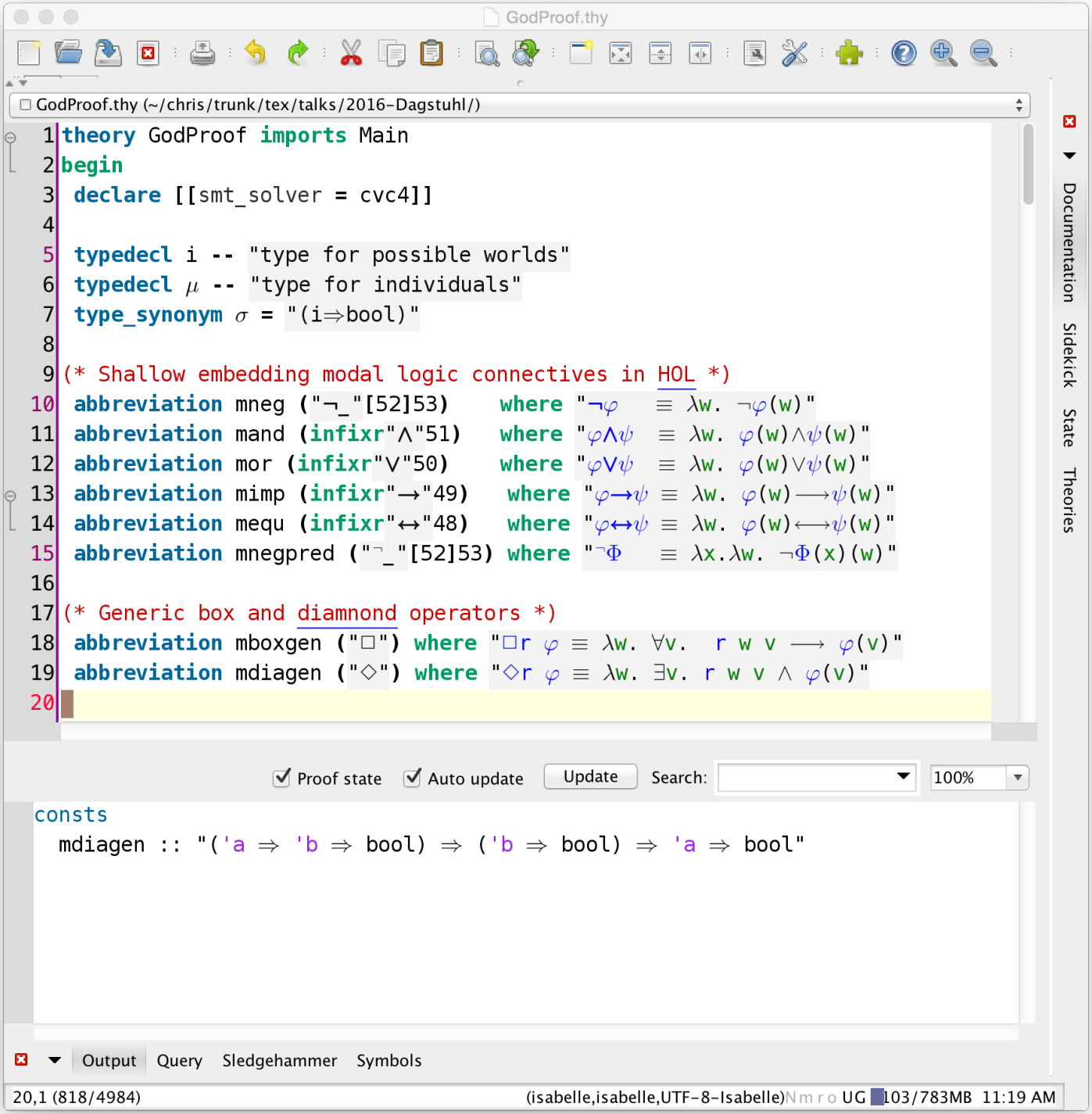
Axiom A5 Necessary existence is a positive property: $P(NE)$

Thm. T3 Necessarily, God exists: $\Box\exists xG(x)$

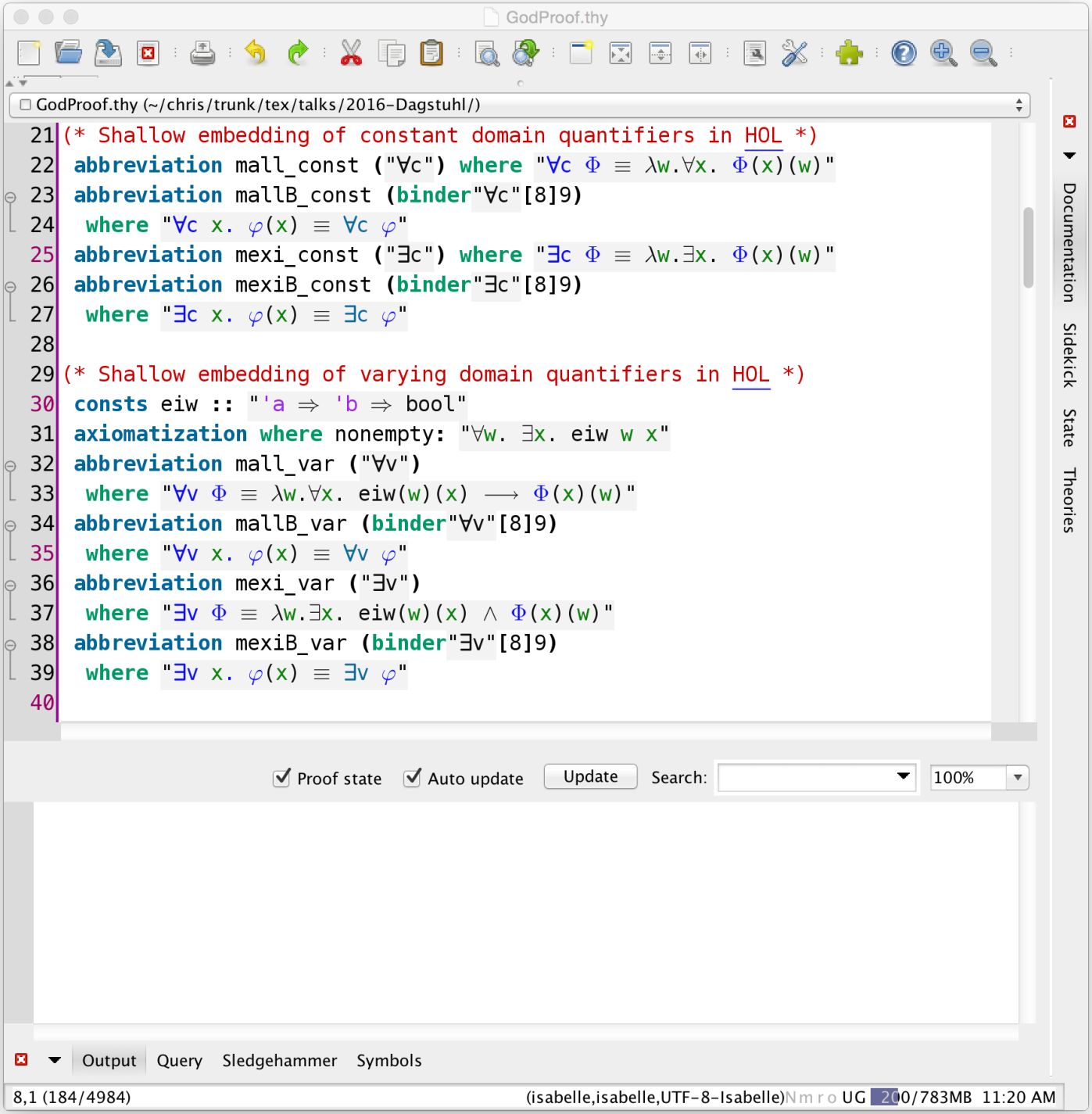
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
115 definition ess (infixr "ess" 85) where
116 "Φ ess x = Φ(x) ∧ (∀ᵖΨ. Ψ(x) ⟶ □(∀ⁱy. Φ(y) ⟶ Ψ(y)))"
117
118 (* T2: Being God-like is an essence of any God-like being *)
119 theorem T2: "⌊∀ⁱx. G(x) ⟶ G ess x⌋" by (smt A1b A4 G_def ess_def)
120
121 (* NE: Necessary existence of an individual is the necessary
122 exemplification of all its essences *)
123 definition NE where "NE(x) = (∀ᵖΦ. Φ ess x ⟶ □(∃ⁱx. Φ(x)))"
124
125 (* A5: Necessary existence is a positive property *)
126 axiomatization where A5: "⌊P(NE)⌋"
127
128 (* T3: Necessarily, God exists *)
129 theorem T3: "⌊□(∃ⁱx. G(x))⌋" by (metis A5 C T2 G_def NE_def S5)
130
131
132 (* Consistency is confirmed by Nitpick *)
133 lemma True nitpick [satisfy, user_axioms] oops
134
Documentation
Sidekick
State
Theories
Proof state
Auto update
Update
Search:
100%
proof (prove)
goal (1 subgoal):
1. ⌊(λw. ⌊S5 w ⟶ mexi_prop G⌋)⌋
Output
Query
Sledgehammer
Symbols
129,27 (4671/4977)
(isabelle,isabelle,UTF-8-Isabelle)
UG
11:54 AM

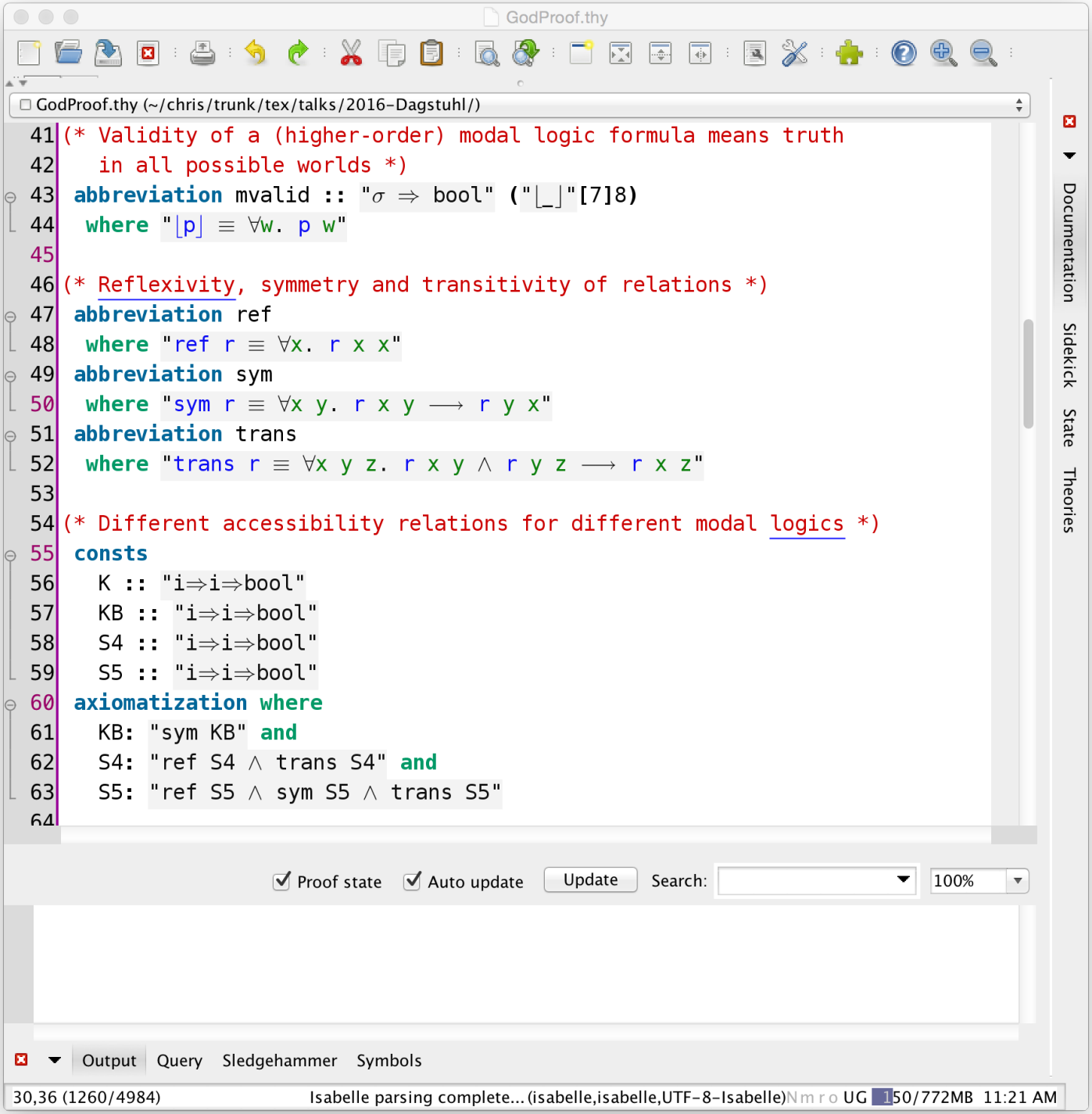
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
1 theory GodProof imports Main
2 begin
3 declare [[smt_solver = cvc4]]
4
5 typedecl i -- "type for possible worlds"
6 typedecl μ -- "type for individuals"
7 type_synonym σ = "(i⇒bool)"
8
9 (* Shallow embedding modal logic connectives in HOL *)
10 abbreviation mneg ("¬_"[52]53) where "¬φ ≡ λw. ¬φ(w)"
11 abbreviation mand (infixr"∧"51) where "φ∧ψ ≡ λw. φ(w)∧ψ(w)"
12 abbreviation mor (infixr"∨"50) where "φ∨ψ ≡ λw. φ(w)∨ψ(w)"
13 abbreviation mimp (infixr"→"49) where "φ→ψ ≡ λw. φ(w)⟶ψ(w)"
14 abbreviation mequ (infixr"↔"48) where "φ↔ψ ≡ λw. φ(w)⟷ψ(w)"
15 abbreviation mnegpred ("¬_"[52]53) where "¬Φ ≡ λx.λw. ¬Φ(x)(w)"
16
17 (* Generic box and diamnond operators *)
18 abbreviation mboxgen ("□") where "□r φ ≡ λw. ∀v. r w v ⟶ φ(v)"
19 abbreviation mdiagen ("◇") where "◇r φ ≡ λw. ∃v. r w v ∧ φ(v)"
20
Proof state
Auto update
Update
Search:
100%
consts
mdiagen :: "('a ⇒ 'b ⇒ bool) ⇒ ('b ⇒ bool) ⇒ 'a ⇒ bool"
Output
Query
Sledgehammer
Symbols
Documentation
Sidekick
State
Theories
20,1 (818/4984)
(isabelle,isabelle,UTF-8-Isabelle)
103/783MB
11:19 AM

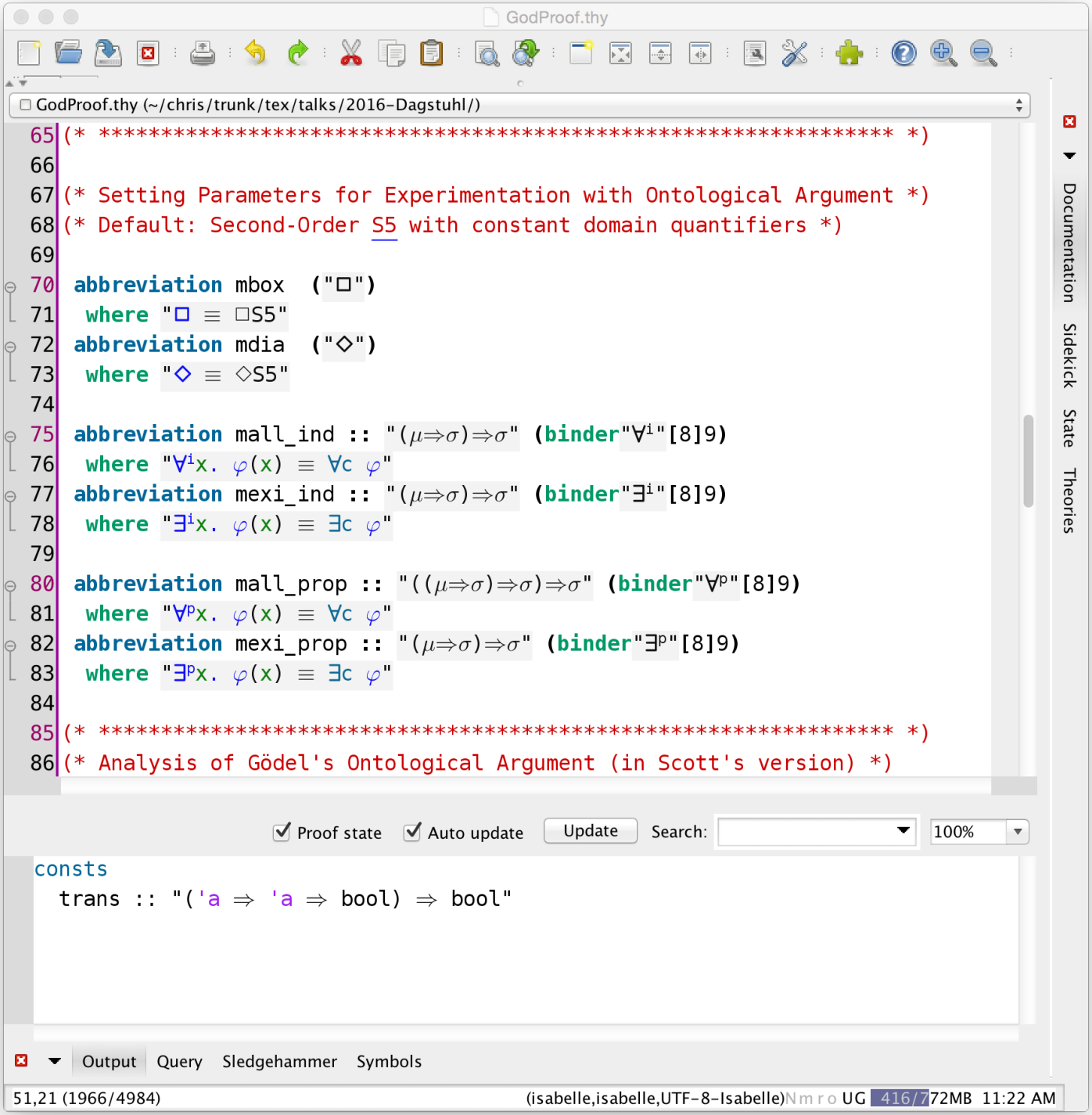
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
(* Shallow embedding of constant domain quantifiers in HOL *)
abbreviation mall_const ("∀c") where "∀c Φ ≡ λw.∀x. Φ(x)(w)"
abbreviation mallB_const (binder"∀c"[8]9)
where "∀c x. φ(x) ≡ ∀c φ"
abbreviation mexi_const ("∃c") where "∃c Φ ≡ λw.∃x. Φ(x)(w)"
abbreviation mexiB_const (binder"∃c"[8]9)
where "∃c x. φ(x) ≡ ∃c φ"
(* Shallow embedding of varying domain quantifiers in HOL *)
consts eiw :: "'a ⇒ 'b ⇒ bool"
axiomatization where nonempty: "∀w. ∃x. eiw w x"
abbreviation mall_var ("∀v")
where "∀v Φ ≡ λw.∀x. eiw(w)(x) ⟶ Φ(x)(w)"
abbreviation mallB_var (binder"∀v"[8]9)
where "∀v x. φ(x) ≡ ∀v φ"
abbreviation mexi_var ("∃v")
where "∃v Φ ≡ λw.∃x. eiw(w)(x) ∧ Φ(x)(w)"
abbreviation mexiB_var (binder"∃v"[8]9)
where "∃v x. φ(x) ≡ ∃v φ"
Proof state
Auto update
Update
Search:
100%
Output
Query
Sledgehammer
Symbols
Documentation
Sidekick
State
Theories
8,1 (184/4984)
(isabelle,isabelle,UTF-8-Isabelle)
UG
0/783MB
11:20 AM

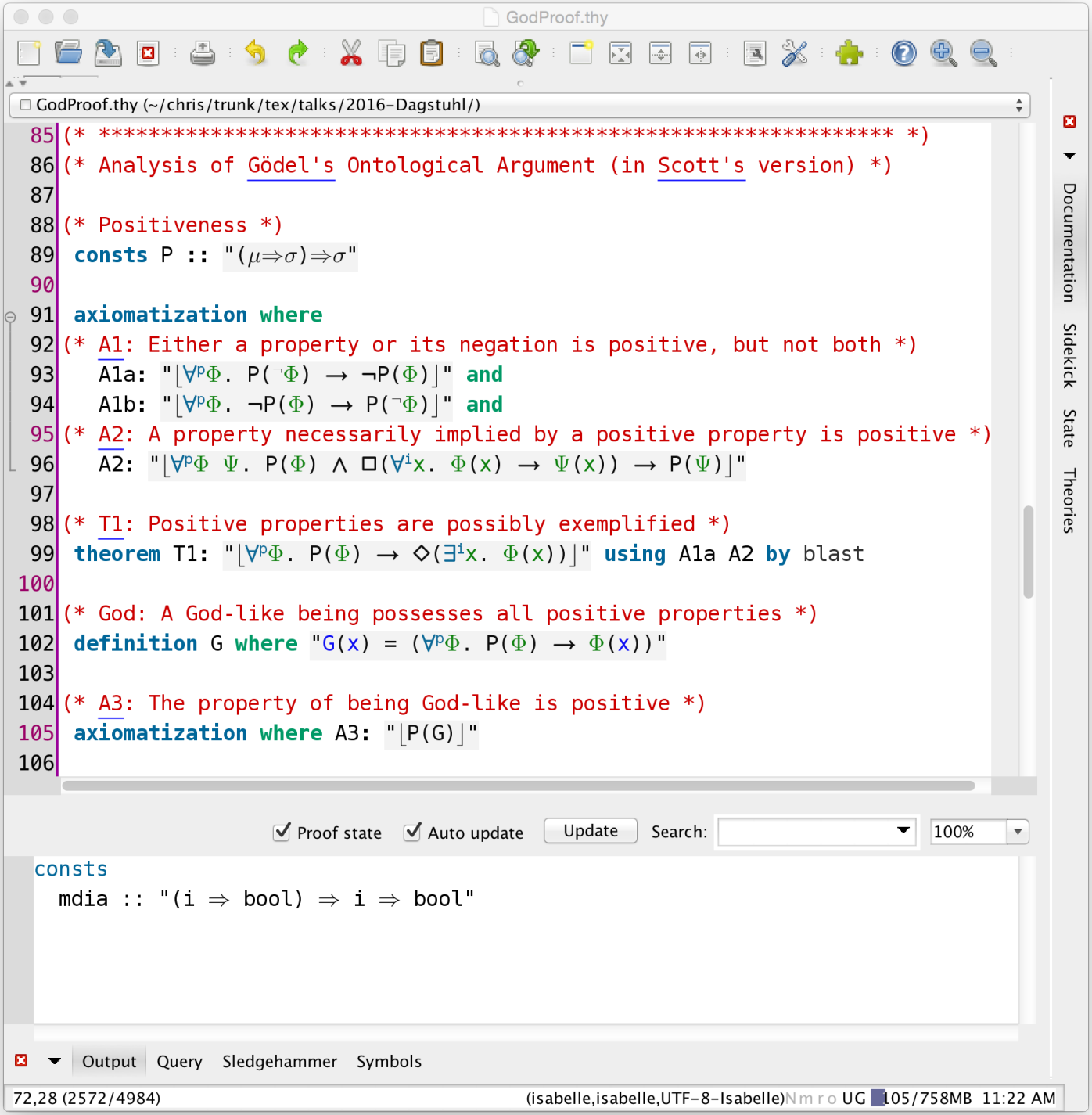
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
(* Validity of a (higher-order) modal logic formula means truth
in all possible worlds *)
abbreviation mvalid :: "σ ⇒ bool" ("⌊_⌋"[7]8)
where "⌊p⌋ ≡ ∀w. p w"
(* Reflexivity, symmetry and transitivity of relations *)
abbreviation ref
where "ref r ≡ ∀x. r x x"
abbreviation sym
where "sym r ≡ ∀x y. r x y ⟶ r y x"
abbreviation trans
where "trans r ≡ ∀x y z. r x y ∧ r y z ⟶ r x z"
(* Different accessibility relations for different modal logics *)
consts
K :: "i⇒i⇒bool"
KB :: "i⇒i⇒bool"
S4 :: "i⇒i⇒bool"
S5 :: "i⇒i⇒bool"
axiomatization where
KB: "sym KB" and
S4: "ref S4 ∧ trans S4" and
S5: "ref S5 ∧ sym S5 ∧ trans S5"
Proof state
Auto update
Update
Search:
100%
Output
Query
Sledgehammer
Symbols
Documentation
Sidekick
State
Theories
30,36 (1260/4984)
Isabelle parsing complete... (isabelle,isabelle,UTF-8-Isabelle)
11:21 AM

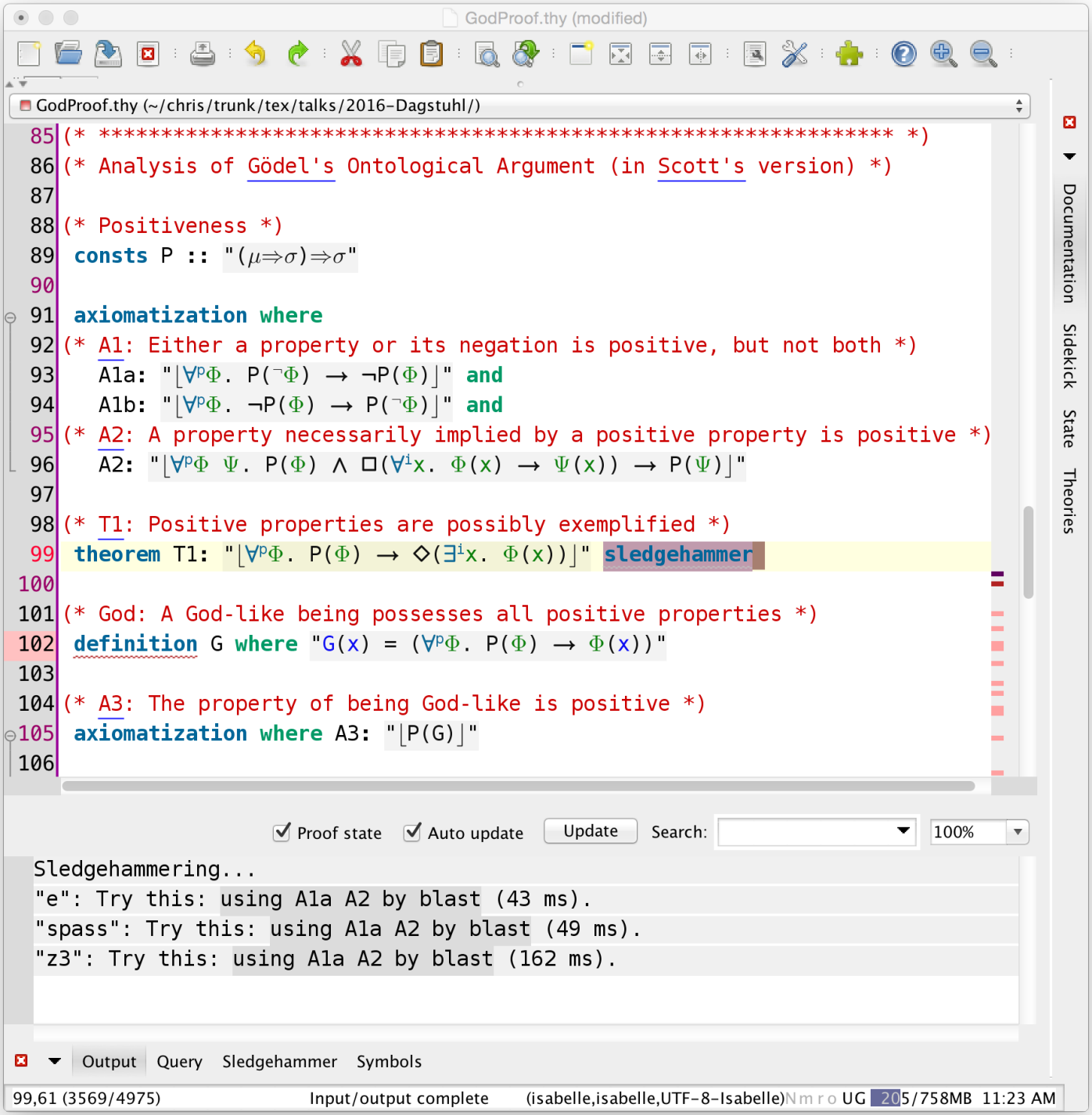
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
(* ********** *)
(* Setting Parameters for Experimentation with Ontological Argument *)
(* Default: Second-Order S5 with constant domain quantifiers *)
abbreviation mbox ("□")
where "□ ≡ □S5"
abbreviation mdia ("◇")
where "◇ ≡ ◇S5"
abbreviation mall_ind :: "(μ⇒σ)⇒σ" (binder"∀ⁱ"[8]9)
where "∀ⁱx. φ(x) ≡ ∀c φ"
abbreviation mexi_ind :: "(μ⇒σ)⇒σ" (binder"∃ⁱ"[8]9)
where "∃ⁱx. φ(x) ≡ ∃c φ"
abbreviation mall_prop :: "((μ⇒σ)⇒σ)⇒σ" (binder"∀ᵖ"[8]9)
where "∀ᵖx. φ(x) ≡ ∀c φ"
abbreviation mexi_prop :: "(μ⇒σ)⇒σ" (binder"∃ᵖ"[8]9)
where "∃ᵖx. φ(x) ≡ ∃c φ"
(* ********** *)
(* Analysis of Gödel's Ontological Argument (in Scott's version) *)
Proof state
Auto update
Update
Search:
100%
consts
trans :: "('a ⇒ 'a ⇒ bool) ⇒ bool"
Output
Query
Sledgehammer
Symbols
Documentation
Sidekick
State
Theories
51,21 (1966/4984)
(isabelle,isabelle,UTF-8-Isabelle)
UG
416/772MB
11:22 AM

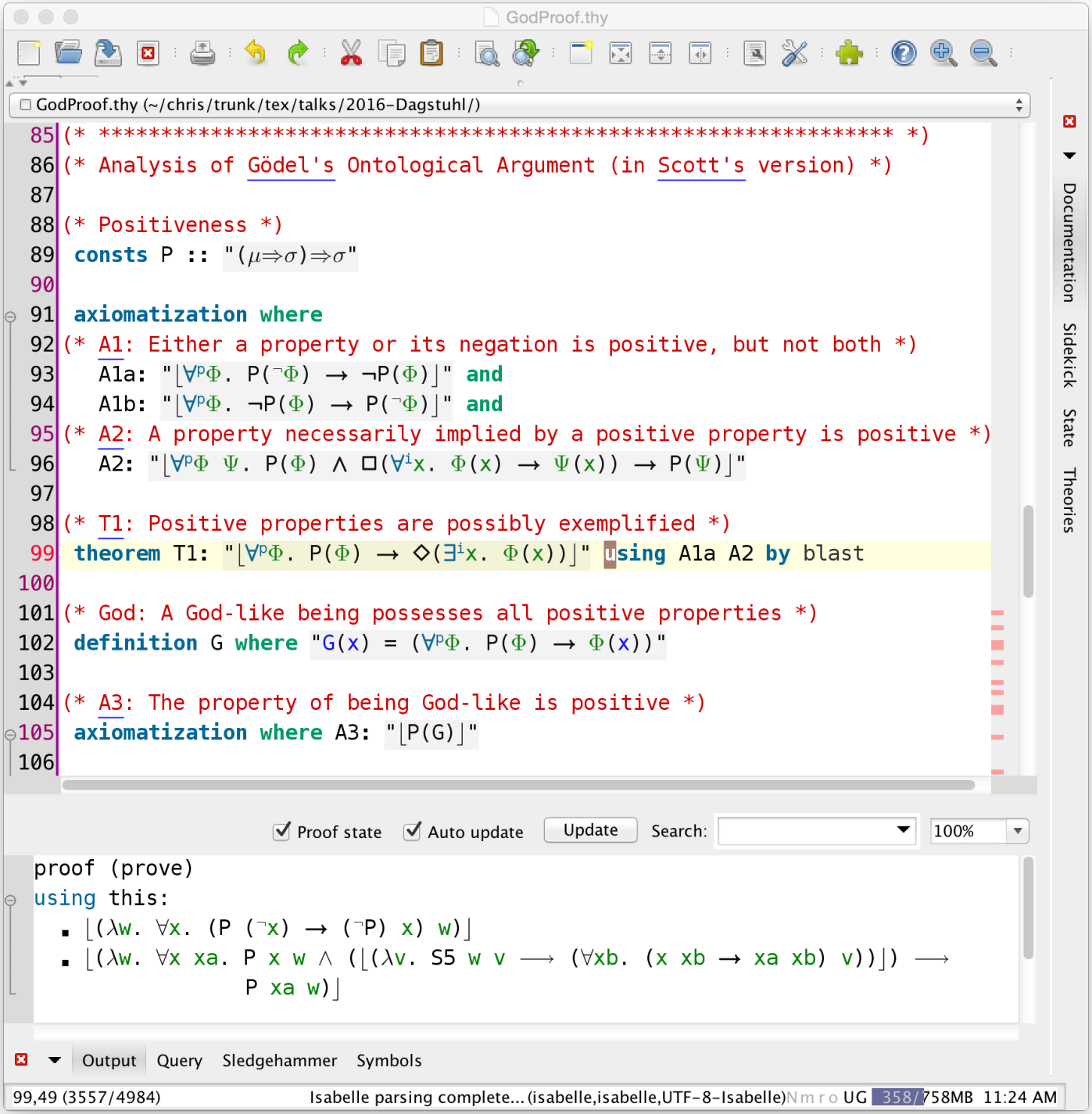
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
(* ********** *)
(* Analysis of Gödel's Ontological Argument (in Scott's version) *)
(* Positiveness *)
consts P :: "(μ⇒σ)⇒σ"
axiomatization where
(* A1: Either a property or its negation is positive, but not both *)
A1a: "⌊∀ᵖΦ. P(¬Φ) ⟶ ¬P(Φ)⌋" and
A1b: "⌊∀ᵖΦ. ¬P(Φ) ⟶ P(¬Φ)⌋" and
(* A2: A property necessarily implied by a positive property is positive *)
A2: "⌊∀ᵖΦ Ψ. P(Φ) ∧ □(∀ⁱx. Φ(x) ⟶ Ψ(x)) ⟶ P(Ψ)⌋"
(* T1: Positive properties are possibly exemplified *)
theorem T1: "⌊∀ᵖΦ. P(Φ) ⟶ ◇(∃ⁱx. Φ(x))⌋" using A1a A2 by blast
(* God: A God-like being possesses all positive properties *)
definition G where "G(x) = (∀ᵖΦ. P(Φ) ⟶ Φ(x))"
(* A3: The property of being God-like is positive *)
axiomatization where A3: "⌊P(G)⌋"
Proof state
Auto update
Update
Search:
100%
consts
mdia :: "(i ⇒ bool) ⇒ i ⇒ bool"
Output
Query
Sledgehammer
Symbols
72,28 (2572/4984)
(isabelle,isabelle,UTF-8-Isabelle)
105/758MB
11:22 AM
Documentation
Sidekick
State
Theories

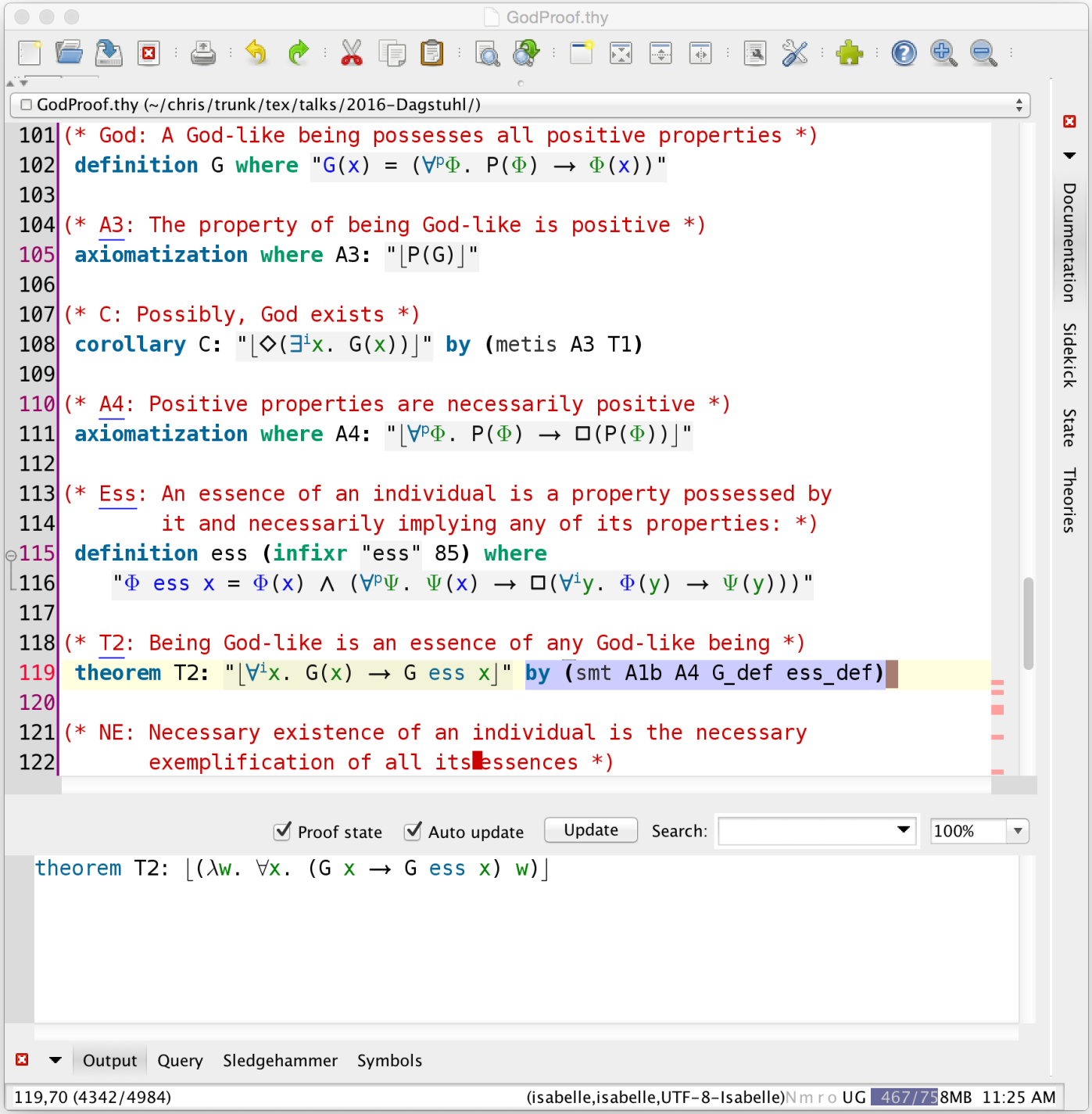
DEMO:


GodProof.thy (modified)
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
(* ********** *)
(* Analysis of Gödel's Ontological Argument (in Scott's version) *)
(* Positiveness *)
consts P :: "(μ⇒σ)⇒σ"
axiomatization where
(* A1: Either a property or its negation is positive, but not both *)
A1a: "⌊∀ᵖΦ. P(¬Φ) ⟶ ¬P(Φ)⌋" and
A1b: "⌊∀ᵖΦ. ¬P(Φ) ⟶ P(¬Φ)⌋" and
(* A2: A property necessarily implied by a positive property is positive *)
A2: "⌊∀ᵖΦ Ψ. P(Φ) ∧ □(∀ⁱx. Φ(x) ⟶ Ψ(x)) ⟶ P(Ψ)⌋"
(* T1: Positive properties are possibly exemplified *)
theorem T1: "⌊∀ᵖΦ. P(Φ) ⟶ ◇(∃ⁱx. Φ(x))⌋" sledgehammer
(* God: A God-like being possesses all positive properties *)
definition G where "G(x) = (∀ᵖΦ. P(Φ) ⟶ Φ(x))"
(* A3: The property of being God-like is positive *)
axiomatization where A3: "⌊P(G)⌋"
Proof state
Auto update
Update
Search:
100%
Sledgehammering...
"e": Try this: using A1a A2 by blast (43 ms).
"spass": Try this: using A1a A2 by blast (49 ms).
"z3": Try this: using A1a A2 by blast (162 ms).
Output
Query
Sledgehammer
Symbols
Documentation
Sidekick
State
Theories
99,61 (3569/4975)
Input/output complete
(isabelle,isabelle,UTF-8-Isabelle)
11:23 AM

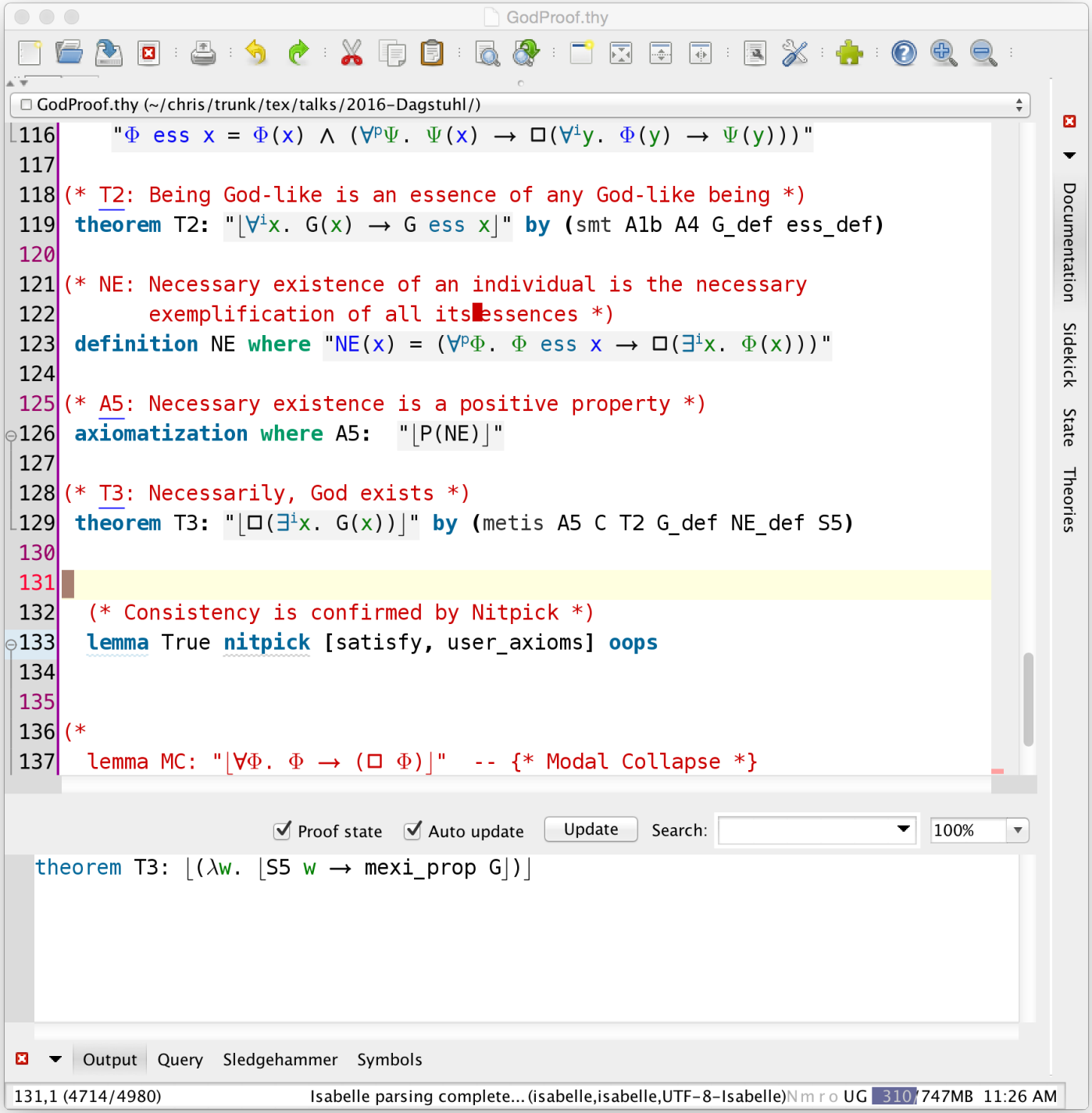
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
(* ********** *)
(* Analysis of Gödel's Ontological Argument (in Scott's version) *)
(* Positiveness *)
consts P :: "(μ⇒σ)⇒σ"
axiomatization where
(* A1: Either a property or its negation is positive, but not both *)
A1a: "⌊∀ᵖΦ. P(¬Φ) ⟶ ¬P(Φ)⌋" and
A1b: "⌊∀ᵖΦ. ¬P(Φ) ⟶ P(¬Φ)⌋" and
(* A2: A property necessarily implied by a positive property is positive *)
A2: "⌊∀ᵖΦ Ψ. P(Φ) ∧ □(∀ⁱx. Φ(x) ⟶ Ψ(x)) ⟶ P(Ψ)⌋"
(* T1: Positive properties are possibly exemplified *)
theorem T1: "⌊∀ᵖΦ. P(Φ) ⟶ ◇(∃ⁱx. Φ(x))⌋" using A1a A2 by blast
(* God: A God-like being possesses all positive properties *)
definition G where "G(x) = (∀ᵖΦ. P(Φ) ⟶ Φ(x))"
(* A3: The property of being God-like is positive *)
axiomatization where A3: "⌊P(G)⌋"
Proof state
Auto update
Update
Search:
100%
proof (prove)
using this:
⌊(λw. ∀x. (P (¬x) ⟶ (¬P) x) w)⌋
⌊(λw. ∀x xa. P x w ∧ (⌊(λv. S5 w v ⟶ (∀xb. (x xb ⟶ xa xb) v))⌋) ⟶
P xa w)⌋
Output
Query
Sledgehammer
Symbols
Documentation
Sidekick
State
Theories
99,49 (3557/4984)
Isabelle parsing complete... (isabelle,isabelle,UTF-8-Isabelle)
UG 358/758MB 11:24 AM

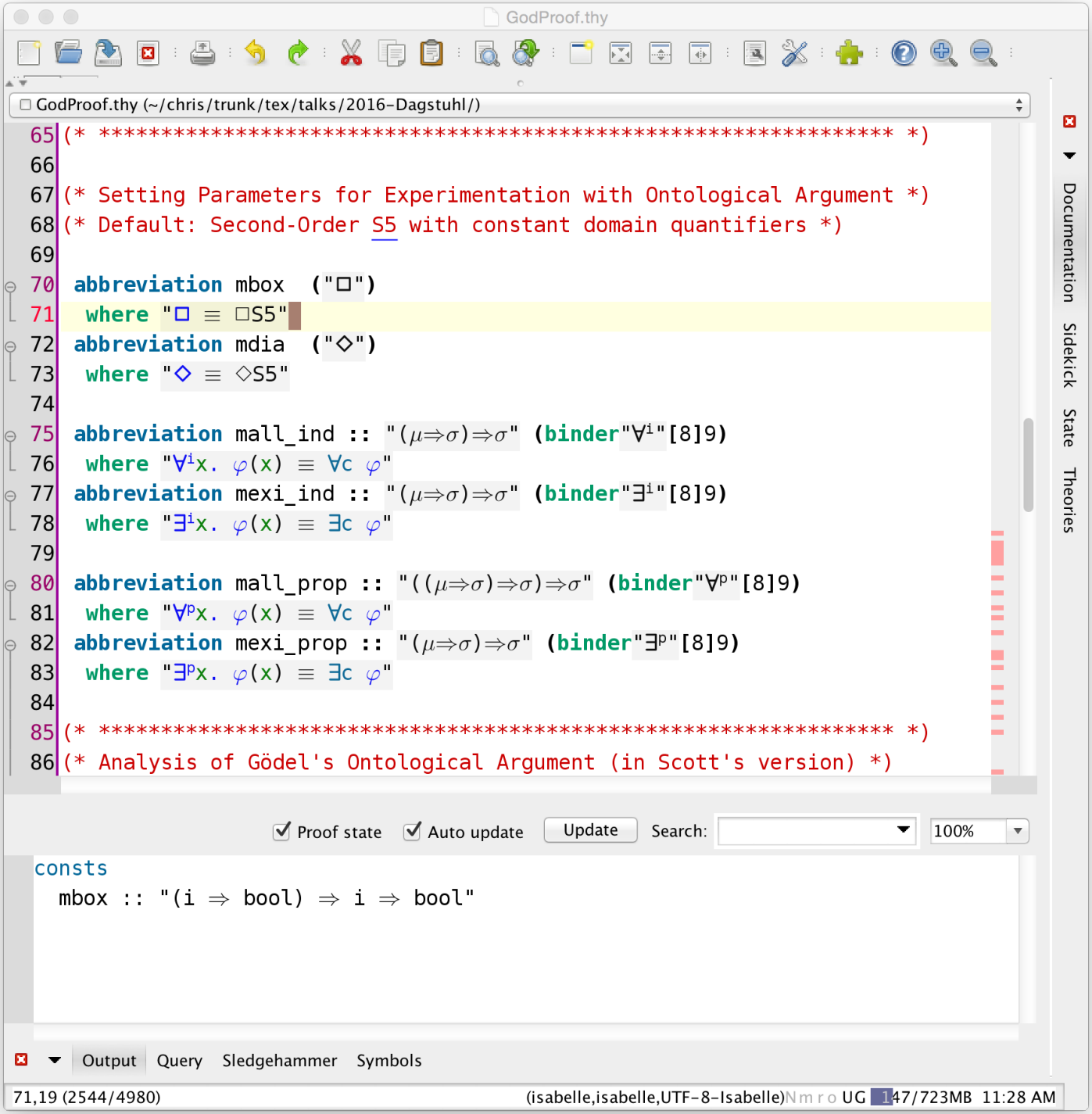
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
(* God: A God-like being possesses all positive properties *)
definition G where "G(x) = (∀ᵖΦ. P(Φ) → Φ(x))"
(* A3: The property of being God-like is positive *)
axiomatization where A3: "⌊P(G)⌋"
(* C: Possibly, God exists *)
corollary C: "⌊◇(∃ⁱx. G(x))⌋" by (metis A3 T1)
(* A4: Positive properties are necessarily positive *)
axiomatization where A4: "⌊∀ᵖΦ. P(Φ) → □(P(Φ))⌋"
(* Ess: An essence of an individual is a property possessed by
it and necessarily implying any of its properties: *)
definition ess (infixr "ess" 85) where
"Φ ess x = Φ(x) ∧ (∀ᵖΨ. Ψ(x) → □(∀ⁱy. Φ(y) → Ψ(y)))"
(* T2: Being God-like is an essence of any God-like being *)
theorem T2: "⌊∀ⁱx. G(x) → G ess x⌋" by (smt A1b A4 G_def ess_def)
(* NE: Necessary existence of an individual is the necessary
exemplification of all its essences *)
Proof state
Auto update
Update
Search:
100%
theorem T2: ⌊(λw. ∀x. (G x → G ess x) w)⌋
Output
Query
Sledgehammer
Symbols
119,70 (4342/4984)
(isabelle,isabelle,UTF-8-Isabelle)
UG
467/758MB
11:25 AM
Documentation
Sidekick
State
Theories

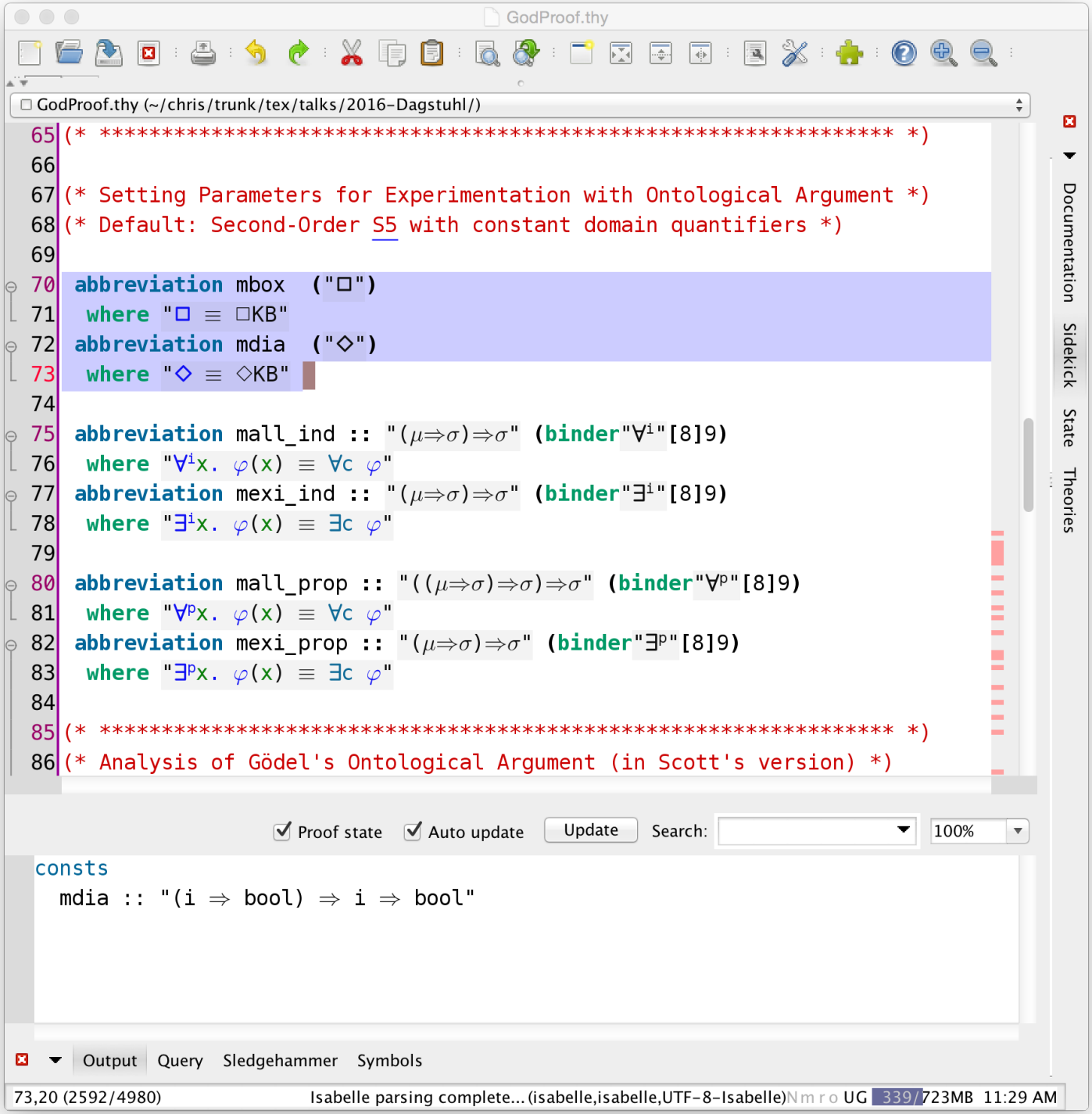
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
"Φ ess x = Φ(x) ∧ (∀ᵖΨ. Ψ(x) → □(∀ⁱy. Φ(y) → Ψ(y)))"
(* T2: Being God-like is an essence of any God-like being *)
theorem T2: "⌊∀ⁱx. G(x) → G ess x⌋" by (smt A1b A4 G_def ess_def)
(* NE: Necessary existence of an individual is the necessary
exemplification of all its essences *)
definition NE where "NE(x) = (∀ᵖΦ. Φ ess x → □(∃ⁱx. Φ(x)))"
(* A5: Necessary existence is a positive property *)
axiomatization where A5: "⌊P(NE)⌋"
(* T3: Necessarily, God exists *)
theorem T3: "⌊□(∃ⁱx. G(x))⌋" by (metis A5 C T2 G_def NE_def S5)
(* Consistency is confirmed by Nitpick *)
lemma True nitpick [satisfy, user_axioms] oops
(*
lemma MC: "⌊∀Φ. Φ → (□ Φ)⌋" -- {* Modal Collapse *}
Proof state
Auto update
Update
Search:
100%
theorem T3: ⌊(λw. ⌊S5 w → mexi_prop G⌋)⌋
Output
Query
Sledgehammer
Symbols
Documentation
Sidekick
State
Theories
131,1 (4714/4980)
Isabelle parsing complete... (isabelle,isabelle,UTF-8-Isabelle)
747MB 11:26 AM

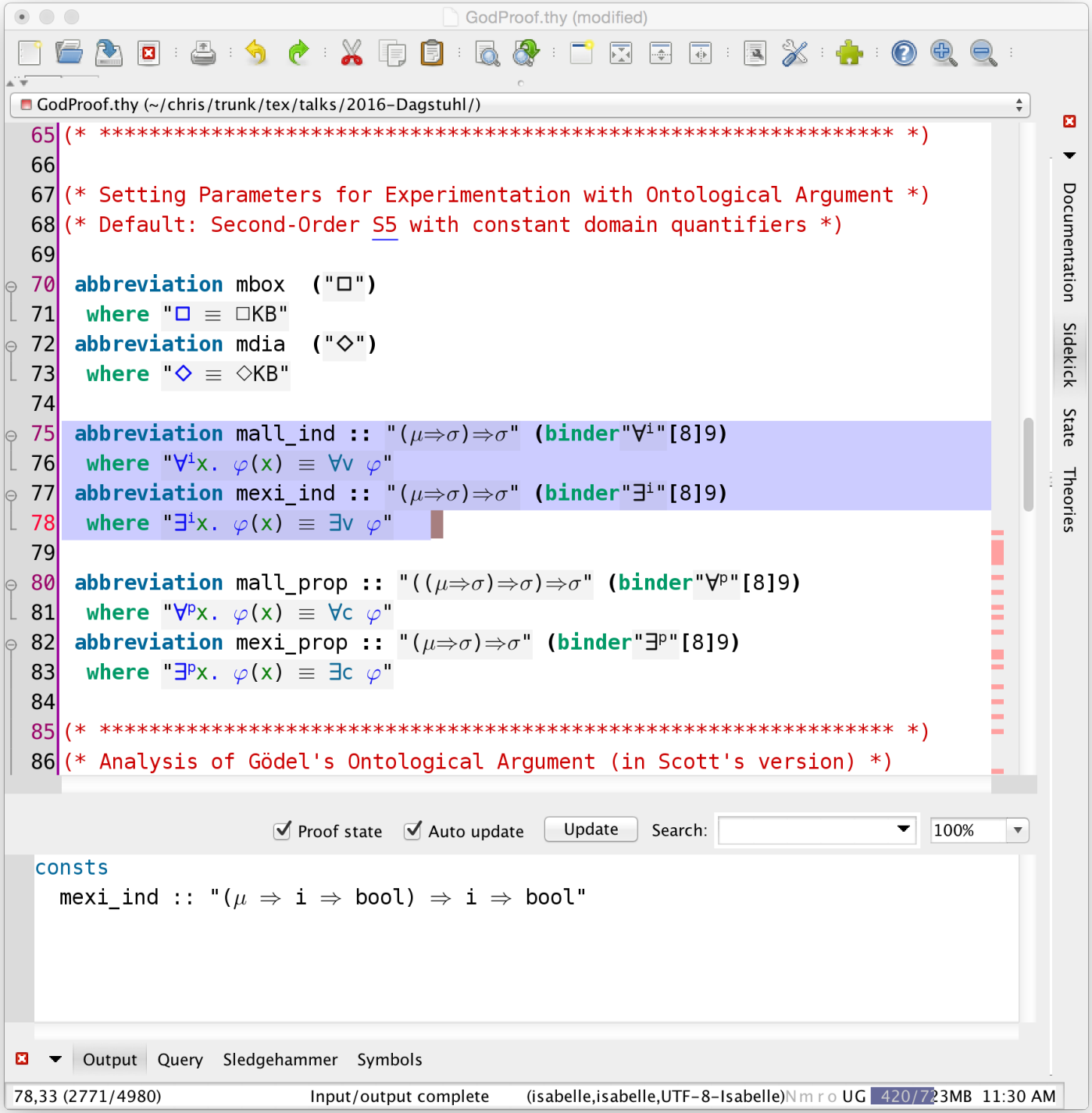
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
65 (* ********** *)
66
67 (* Setting Parameters for Experimentation with Ontological Argument *)
68 (* Default: Second-Order S5 with constant domain quantifiers *)
69
70 abbreviation mbox ("□")
71 where "□ ≡ □S5"
72 abbreviation mdia ("◇")
73 where "◇ ≡ ◇S5"
74
75 abbreviation mall_ind :: "(μ⇒σ)⇒σ" (binder"∀ⁱ"[8]9)
76 where "∀ⁱx. φ(x) ≡ ∀c φ"
77 abbreviation mexi_ind :: "(μ⇒σ)⇒σ" (binder"∃ⁱ"[8]9)
78 where "∃ⁱx. φ(x) ≡ ∃c φ"
79
80 abbreviation mall_prop :: "((μ⇒σ)⇒σ)⇒σ" (binder"∀ᵖ"[8]9)
81 where "∀ᵖx. φ(x) ≡ ∀c φ"
82 abbreviation mexi_prop :: "(μ⇒σ)⇒σ" (binder"∃ᵖ"[8]9)
83 where "∃ᵖx. φ(x) ≡ ∃c φ"
84
85 (* ********** *)
86 (* Analysis of Gödel's Ontological Argument (in Scott's version) *)
Documentation
Sidekick
State
Theories
Proof state
Auto update
Update
Search:
100%
consts
mbox :: "(i ⇒ bool) ⇒ i ⇒ bool"
Output
Query
Sledgehammer
Symbols
71,19 (2544/4980)
(isabelle,isabelle,UTF-8-Isabelle)
UG
747/723MB
11:28 AM

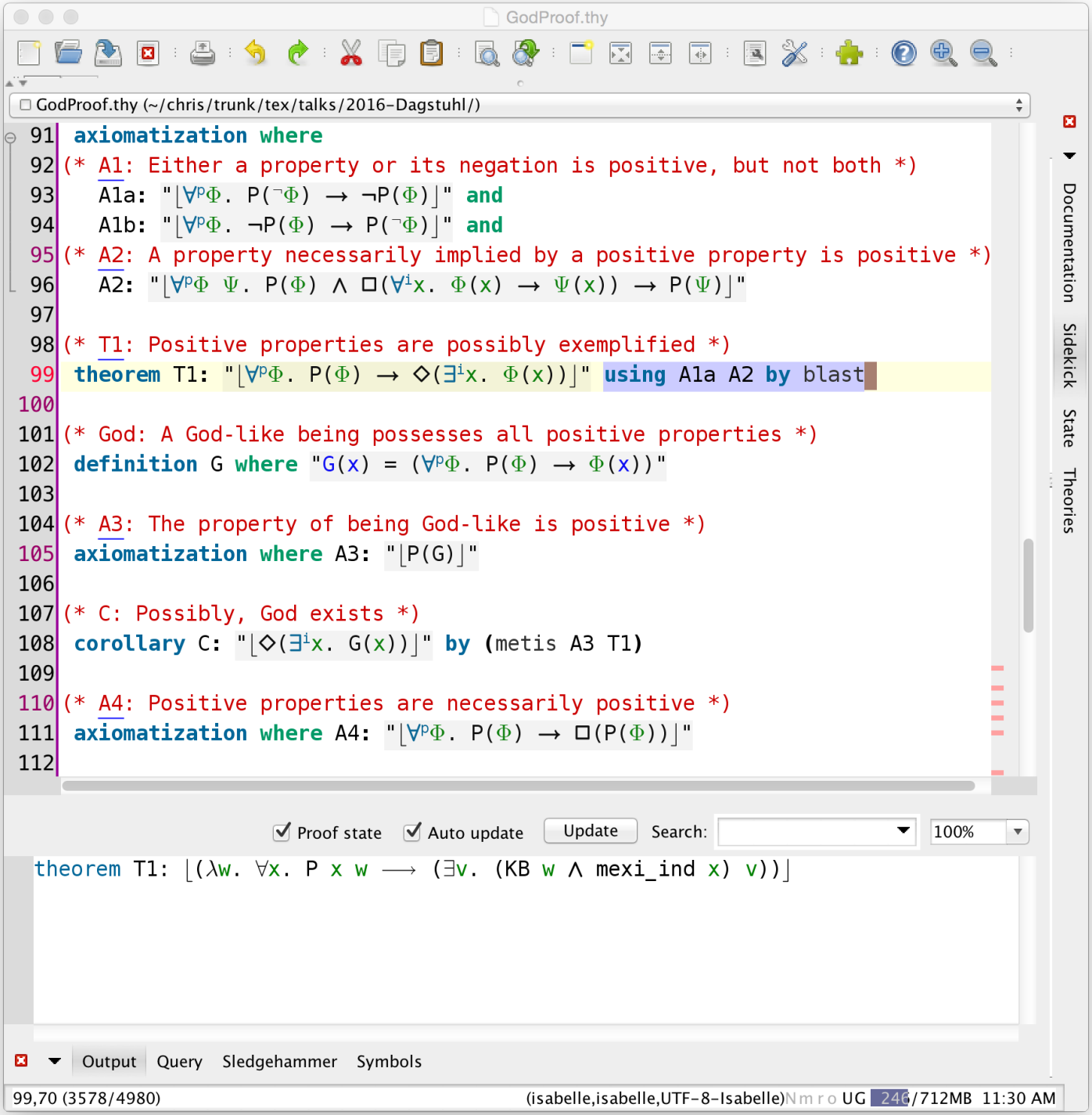
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
(* ********** *)
(* Setting Parameters for Experimentation with Ontological Argument *)
(* Default: Second-Order S5 with constant domain quantifiers *)
abbreviation mbox ("□")
where "□ ≡ □KB"
abbreviation mdia ("◇")
where "◇ ≡ ◇KB"
abbreviation mall_ind :: "(μ⇒σ)⇒σ" (binder"∀ⁱ"[8]9)
where "∀ⁱx. φ(x) ≡ ∀c φ"
abbreviation mexi_ind :: "(μ⇒σ)⇒σ" (binder"∃ⁱ"[8]9)
where "∃ⁱx. φ(x) ≡ ∃c φ"
abbreviation mall_prop :: "((μ⇒σ)⇒σ)⇒σ" (binder"∀ᵖ"[8]9)
where "∀ᵖx. φ(x) ≡ ∀c φ"
abbreviation mexi_prop :: "(μ⇒σ)⇒σ" (binder"∃ᵖ"[8]9)
where "∃ᵖx. φ(x) ≡ ∃c φ"
(* ********** *)
(* Analysis of Gödel's Ontological Argument (in Scott's version) *)
Documentation
Sidekick
State
Theories
Proof state
Auto update
Update
Search:
100%
consts
mdia :: "(i ⇒ bool) ⇒ i ⇒ bool"
Output
Query
Sledgehammer
Symbols
73,20 (2592/4980)
Isabelle parsing complete... (isabelle,isabelle,UTF-8-Isabelle)
339/723MB
11:29 AM

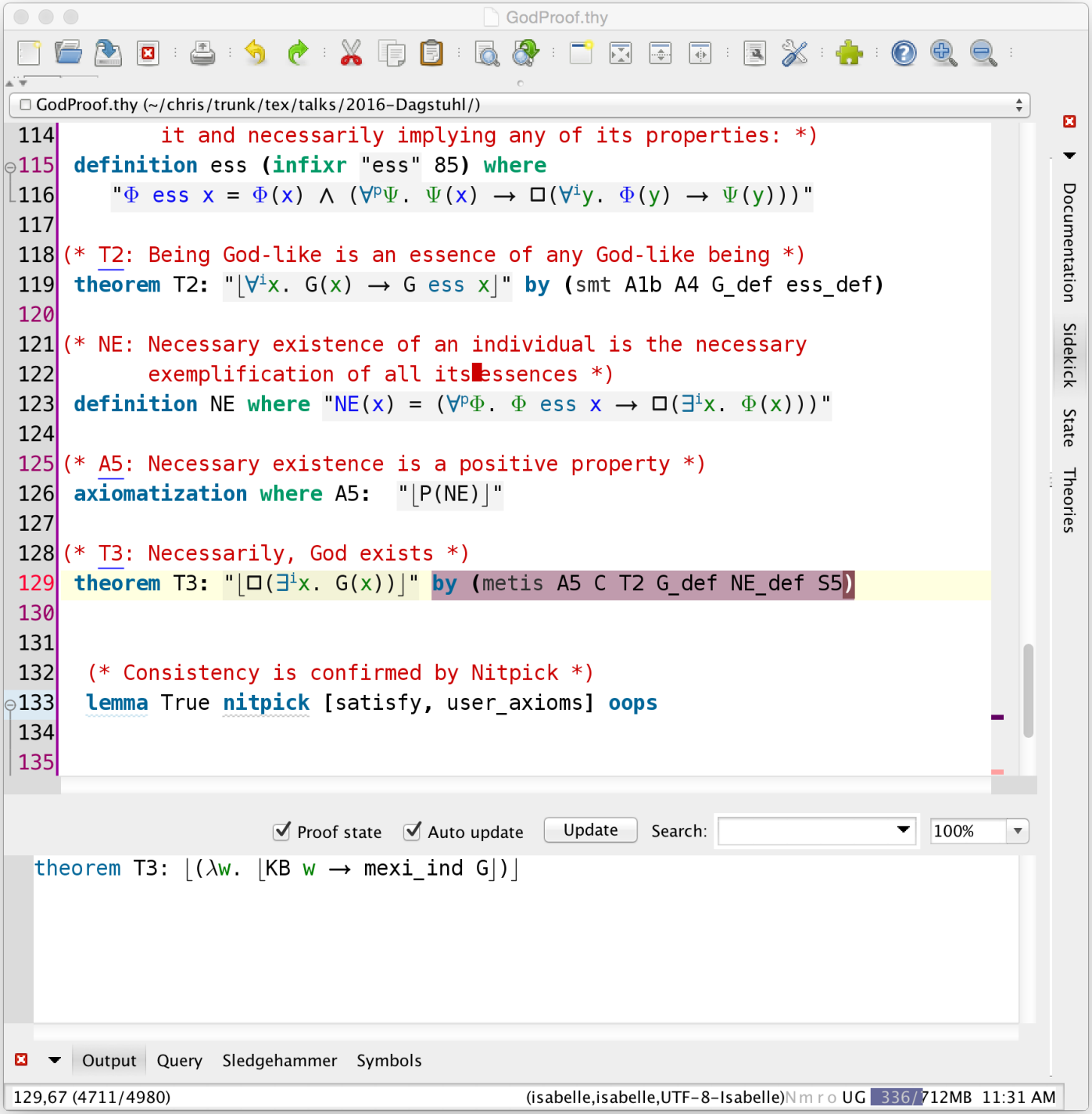
DEMO:


GodProof.thy (modified)
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
(* ********** *)
(* Setting Parameters for Experimentation with Ontological Argument *)
(* Default: Second-Order S5 with constant domain quantifiers *)
abbreviation mbox ("□")
where "□ ≡ □KB"
abbreviation mdia ("◇")
where "◇ ≡ ◇KB"
abbreviation mall_ind :: "(μ⇒σ)⇒σ" (binder"∀ⁱ"[8]9)
where "∀ⁱx. φ(x) ≡ ∀v φ"
abbreviation mexi_ind :: "(μ⇒σ)⇒σ" (binder"∃ⁱ"[8]9)
where "∃ⁱx. φ(x) ≡ ∃v φ"
abbreviation mall_prop :: "((μ⇒σ)⇒σ)⇒σ" (binder"∀ᵖ"[8]9)
where "∀ᵖx. φ(x) ≡ ∀c φ"
abbreviation mexi_prop :: "(μ⇒σ)⇒σ" (binder"∃ᵖ"[8]9)
where "∃ᵖx. φ(x) ≡ ∃c φ"
(* ********** *)
(* Analysis of Gödel's Ontological Argument (in Scott's version) *)
Documentation
Sidekick
State
Theories
Proof state
Auto update
Update
Search:
100%
consts
mexi_ind :: "(μ ⇒ i ⇒ bool) ⇒ i ⇒ bool"
Output
Query
Sledgehammer
Symbols
78,33 (2771/4980)
Input/output complete
(isabelle,isabelle,UTF-8-Isabelle)
UG
420/723MB
11:30 AM

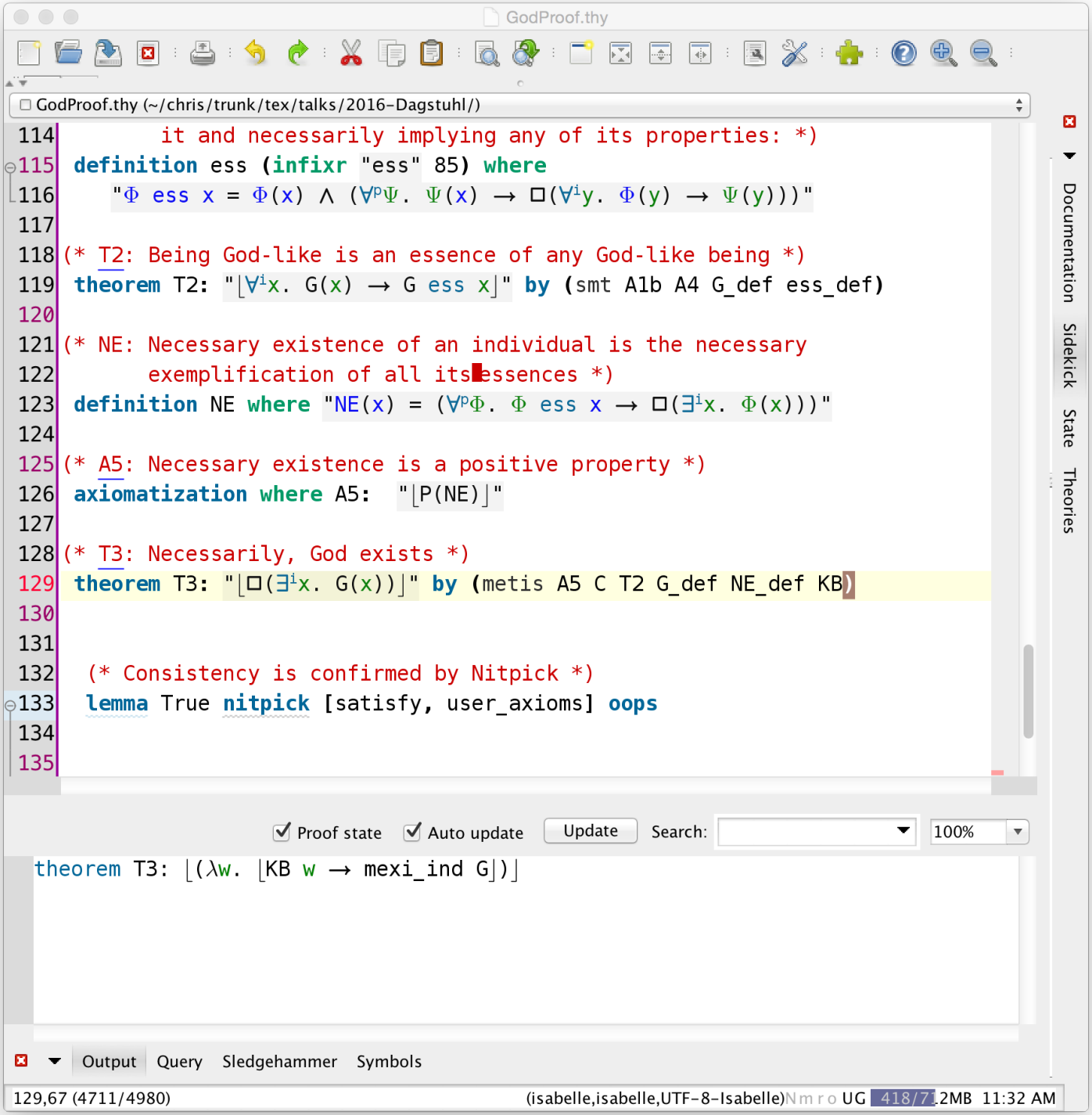
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
axiomatization where
(* A1: Either a property or its negation is positive, but not both *)
A1a: "⌊∀ᵖΦ. P(¬Φ) ⟶ ¬P(Φ)⌋" and
A1b: "⌊∀ᵖΦ. ¬P(Φ) ⟶ P(¬Φ)⌋" and
(* A2: A property necessarily implied by a positive property is positive *)
A2: "⌊∀ᵖΦ Ψ. P(Φ) ∧ □(∀ⁱx. Φ(x) ⟶ Ψ(x)) ⟶ P(Ψ)⌋"
(* T1: Positive properties are possibly exemplified *)
theorem T1: "⌊∀ᵖΦ. P(Φ) ⟶ ◇(∃ⁱx. Φ(x))⌋" using A1a A2 by blast
(* God: A God-like being possesses all positive properties *)
definition G where "G(x) = (∀ᵖΦ. P(Φ) ⟶ Φ(x))"
(* A3: The property of being God-like is positive *)
axiomatization where A3: "⌊P(G)⌋"
(* C: Possibly, God exists *)
corollary C: "⌊◇(∃ⁱx. G(x))⌋" by (metis A3 T1)
(* A4: Positive properties are necessarily positive *)
axiomatization where A4: "⌊∀ᵖΦ. P(Φ) ⟶ □(P(Φ))⌋"
Documentation
Sidekick
State
Theories
Proof state
Auto update
Update
Search:
100%
theorem T1: ⌊(λw. ∀x. P x w ⟶ (∃v. (KB w ∧ mexi_ind x) v))⌋
Output
Query
Sledgehammer
Symbols
99,70 (3578/4980)
(isabelle,isabelle,UTF-8-Isabelle)
UG
246/712MB
11:30 AM

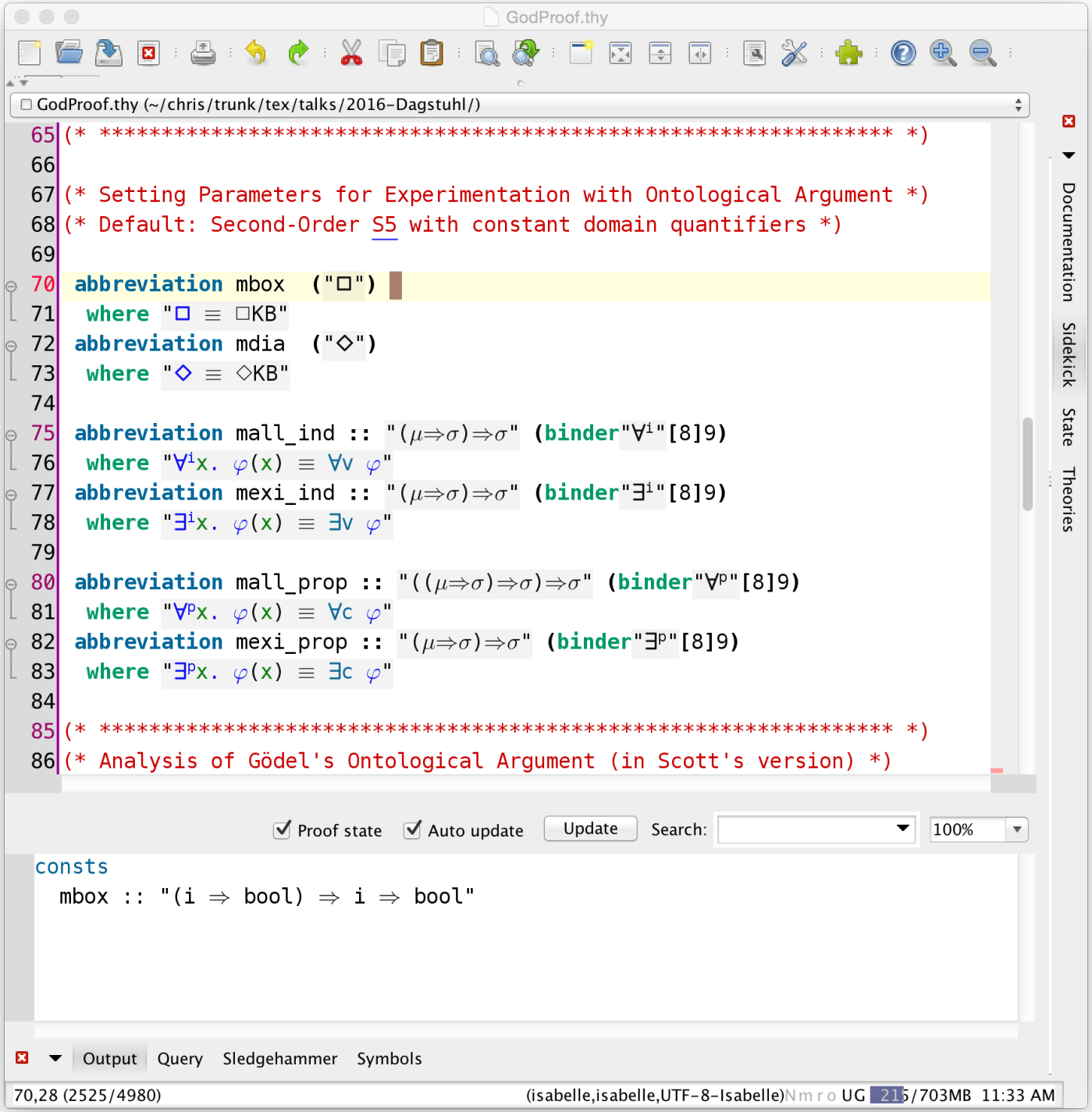
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
it and necessarily implying any of its properties: *)
definition ess (infixr "ess" 85) where
"Φ ess x = Φ(x) ∧ (∀ᵖΨ. Ψ(x) → □(∀ⁱy. Φ(y) → Ψ(y)))"
(* T2: Being God-like is an essence of any God-like being *)
theorem T2: "⌊∀ⁱx. G(x) → G ess x⌋" by (smt A1b A4 G_def ess_def)
(* NE: Necessary existence of an individual is the necessary
exemplification of all its essences *)
definition NE where "NE(x) = (∀ᵖΦ. Φ ess x → □(∃ⁱx. Φ(x)))"
(* A5: Necessary existence is a positive property *)
axiomatization where A5: "⌊P(NE)⌋"
(* T3: Necessarily, God exists *)
theorem T3: "⌊□(∃ⁱx. G(x))⌋" by (metis A5 C T2 G_def NE_def S5)
(* Consistency is confirmed by Nitpick *)
lemma True nitpick [satisfy, user_axioms] oops
Proof state
Auto update
Update
Search:
100%
theorem T3: ⌊(λw. ⌊KB w → mexi_ind G⌋)⌋
Output
Query
Sledgehammer
Symbols
Documentation
Sidekick
State
Theories
129,67 (4711/4980)
(isabelle,isabelle,UTF-8-Isabelle)
336/712MB
11:31 AM

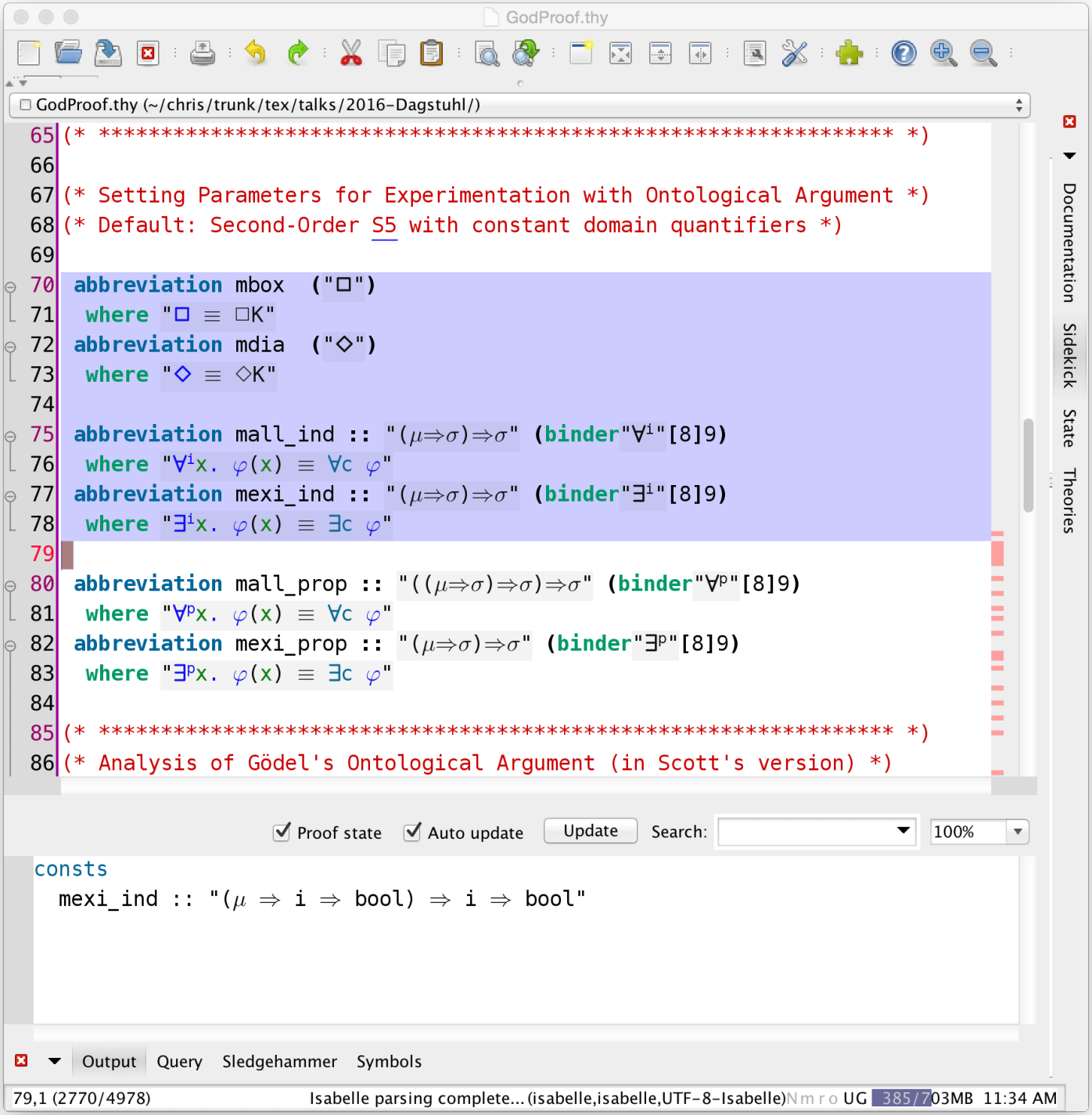
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
it and necessarily implying any of its properties: *)
definition ess (infixr "ess" 85) where
"Φ ess x = Φ(x) ∧ (∀ᵖΨ. Ψ(x) → □(∀ⁱy. Φ(y) → Ψ(y)))"
(* T2: Being God-like is an essence of any God-like being *)
theorem T2: "⌊∀ⁱx. G(x) → G ess x⌋" by (smt A1b A4 G_def ess_def)
(* NE: Necessary existence of an individual is the necessary
exemplification of all its essences *)
definition NE where "NE(x) = (∀ᵖΦ. Φ ess x → □(∃ⁱx. Φ(x)))"
(* A5: Necessary existence is a positive property *)
axiomatization where A5: "⌊P(NE)⌋"
(* T3: Necessarily, God exists *)
theorem T3: "⌊□(∃ⁱx. G(x))⌋" by (metis A5 C T2 G_def NE_def KB)
(* Consistency is confirmed by Nitpick *)
lemma True nitpick [satisfy, user_axioms] oops
Proof state
Auto update
Update
Search:
100%
theorem T3: ⌊(λw. ⌊KB w → mexi_ind G⌋)⌋
Output
Query
Sledgehammer
Symbols
Documentation
Sidekick
State
Theories
129,67 (4711/4980)
(isabelle,isabelle,UTF-8-Isabelle)
UG
11:32 AM

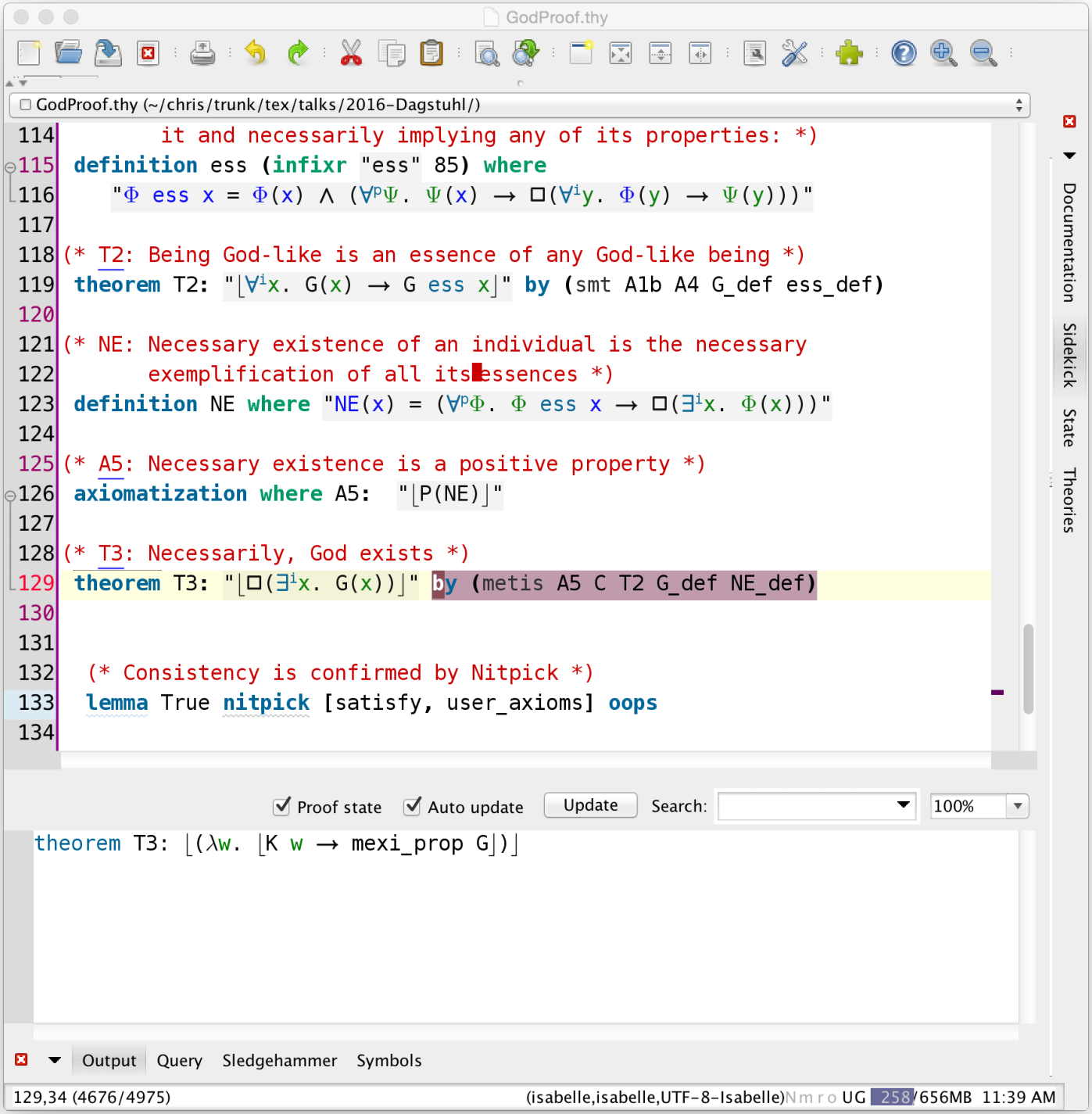
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
65 (* ********** *)
66
67 (* Setting Parameters for Experimentation with Ontological Argument *)
68 (* Default: Second-Order S5 with constant domain quantifiers *)
69
70 abbreviation mbox ("□")
71 where "□ ≡ □KB"
72 abbreviation mdia ("◇")
73 where "◇ ≡ ◇KB"
74
75 abbreviation mall_ind :: "(μ⇒σ)⇒σ" (binder"∀ⁱ"[8]9)
76 where "∀ⁱx. φ(x) ≡ ∀v φ"
77 abbreviation mexi_ind :: "(μ⇒σ)⇒σ" (binder"∃ⁱ"[8]9)
78 where "∃ⁱx. φ(x) ≡ ∃v φ"
79
80 abbreviation mall_prop :: "((μ⇒σ)⇒σ)⇒σ" (binder"∀ᵖ"[8]9)
81 where "∀ᵖx. φ(x) ≡ ∀c φ"
82 abbreviation mexi_prop :: "(μ⇒σ)⇒σ" (binder"∃ᵖ"[8]9)
83 where "∃ᵖx. φ(x) ≡ ∃c φ"
84
85 (* ********** *)
86 (* Analysis of Gödel's Ontological Argument (in Scott's version) *)
Documentation
Sidekick
State
Theories
Proof state
Auto update
Update
Search:
100%
consts
mbox :: "(i ⇒ bool) ⇒ i ⇒ bool"
Output
Query
Sledgehammer
Symbols
70,28 (2525/4980)
(isabelle,isabelle,UTF-8-Isabelle)
UG
/703MB
11:33 AM

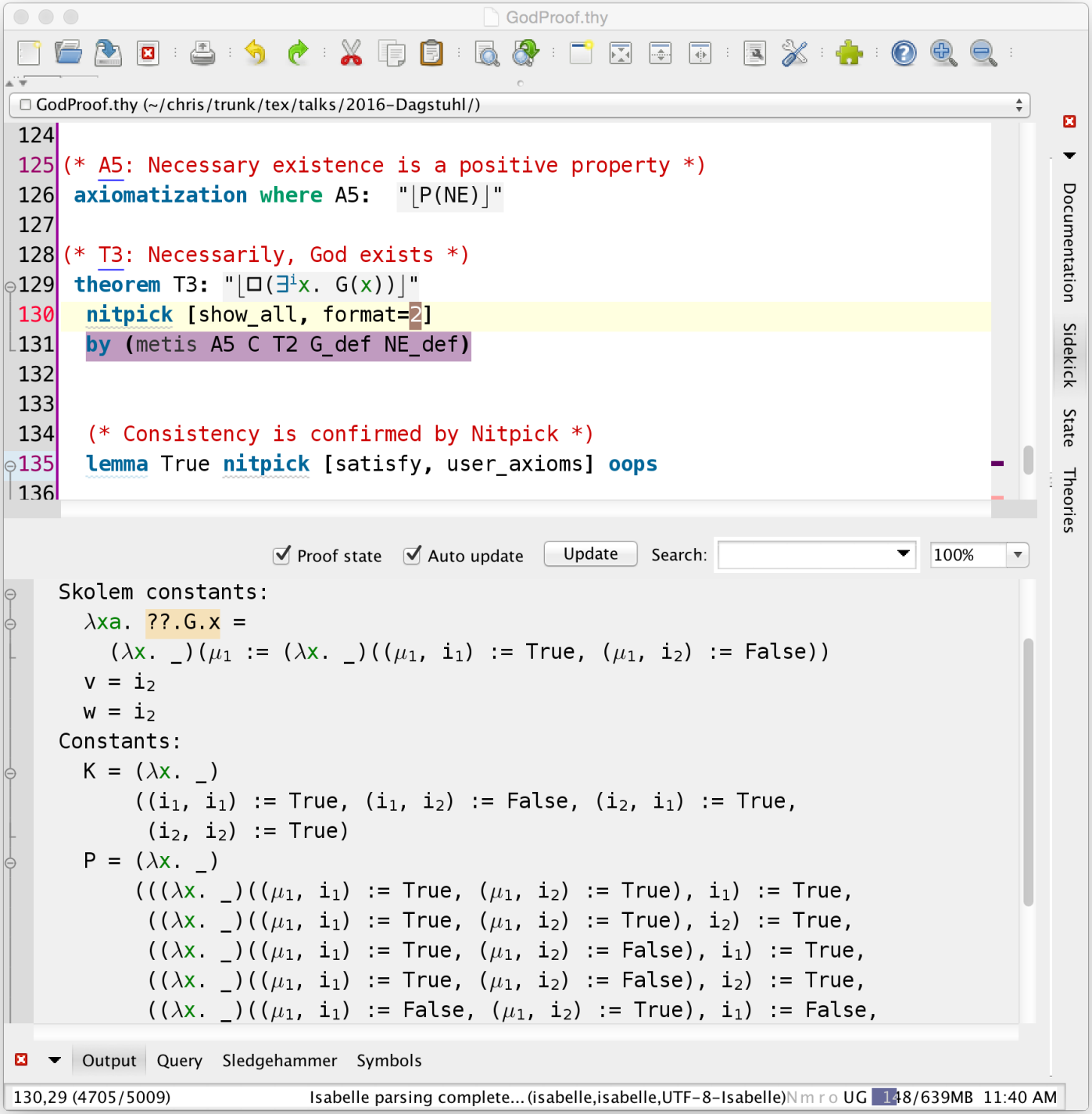
DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
65 (* ********** *)
66
67 (* Setting Parameters for Experimentation with Ontological Argument *)
68 (* Default: Second-Order S5 with constant domain quantifiers *)
69
70 abbreviation mbox ("□")
71 where "□ ≡ □K"
72 abbreviation mdia ("◇")
73 where "◇ ≡ ◇K"
74
75 abbreviation mall_ind :: "(μ⇒σ)⇒σ" (binder"∀ⁱ"[8]9)
76 where "∀ⁱx. φ(x) ≡ ∀c φ"
77 abbreviation mexi_ind :: "(μ⇒σ)⇒σ" (binder"∃ⁱ"[8]9)
78 where "∃ⁱx. φ(x) ≡ ∃c φ"
79
80 abbreviation mall_prop :: "((μ⇒σ)⇒σ)⇒σ" (binder"∀ᵖ"[8]9)
81 where "∀ᵖx. φ(x) ≡ ∀c φ"
82 abbreviation mexi_prop :: "(μ⇒σ)⇒σ" (binder"∃ᵖ"[8]9)
83 where "∃ᵖx. φ(x) ≡ ∃c φ"
84
85 (* ********** *)
86 (* Analysis of Gödel's Ontological Argument (in Scott's version) *)
Documentation
Sidekick
State
Theories
Proof state
Auto update
Update
Search:
100%
consts
mexi_ind :: "(μ ⇒ i ⇒ bool) ⇒ i ⇒ bool"
Output
Query
Sledgehammer
Symbols
79,1 (2770/4978)
Isabelle parsing complete... (isabelle,isabelle,UTF-8-Isabelle)
385/703MB
11:34 AM

DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
it and necessarily implying any of its properties: *)
definition ess (infixr "ess" 85) where
"Φ ess x = Φ(x) ∧ (∀ᵖΨ. Ψ(x) → □(∀ⁱy. Φ(y) → Ψ(y)))"
(* T2: Being God-like is an essence of any God-like being *)
theorem T2: "⌊∀ⁱx. G(x) → G ess x⌋" by (smt A1b A4 G_def ess_def)
(* NE: Necessary existence of an individual is the necessary
exemplification of all its essences *)
definition NE where "NE(x) = (∀ᵖΦ. Φ ess x → □(∃ⁱx. Φ(x)))"
(* A5: Necessary existence is a positive property *)
axiomatization where A5: "⌊P(NE)⌋"
(* T3: Necessarily, God exists *)
theorem T3: "⌊□(∃ⁱx. G(x))⌋" by (metis A5 C T2 G_def NE_def)
(* Consistency is confirmed by Nitpick *)
lemma True nitpick [satisfy, user_axioms] oops
Proof state
Auto update
Update
Search:
100%
theorem T3: ⌊(λw. ⌊K w → mexi_prop G⌋)⌋
Output
Query
Sledgehammer
Symbols
Documentation
Sidekick
State
Theories
129,34 (4676/4975)
(isabelle,isabelle,UTF-8-Isabelle)
UG
258/656MB
11:39 AM

DEMO:


GodProof.thy
GodProof.thy (~/chris/trunk/tex/talks/2016-Dagstuhl/)
124
125 (* A5: Necessary existence is a positive property *)
126 axiomatization where A5: "⌊P(NE)⌋"
127
128 (* T3: Necessarily, God exists *)
129 theorem T3: "⌊□(∃ⁱx. G(x))⌋"
130 nitpick [show_all, format=2]
131 by (metis A5 C T2 G_def NE_def)
132
133
134 (* Consistency is confirmed by Nitpick *)
135 lemma True nitpick [satisfy, user_axioms] oops
136
Proof state
Auto update
Update
Search:
100%
Skolem constants:
λxa. ??.G.x =
(λx. _)(μ1 := (λx. _)((μ1, i1) := True, (μ1, i2) := False))
v = i2
w = i2
Constants:
K = (λx. _)
((i1, i1) := True, (i1, i2) := False, (i2, i1) := True,
(i2, i2) := True)
P = (λx. _)
(((λx. _)((μ1, i1) := True, (μ1, i2) := True), i1) := True,
((λx. _)((μ1, i1) := True, (μ1, i2) := True), i2) := True,
((λx. _)((μ1, i1) := True, (μ1, i2) := False), i1) := True,
((λx. _)((μ1, i1) := True, (μ1, i2) := False), i2) := True,
((λx. _)((μ1, i1) := False, (μ1, i2) := True), i1) := False,
Output
Query
Sledgehammer
Symbols
Documentation
Sidekick
State
Theories
130,29 (4705/5009)
Isabelle parsing complete... (isabelle,isabelle,UTF-8-Isabelle)
UG
11:40 AM

Conclusion

HOL as a (quite) Universal Metalogic via Shallow Semantic Embeddings:

- Very lean approach to integrate and combine logics
- High degree of automation (via the embeddings)
- Uniform proofs (via the embedding embeddings)
- Works surprisingly well in (our recent) practical applications
- Intuitive interaction at abstract level supported by proof assistants
- Novel results in different application domains
- Approach well suited also for teaching logics

Approach has recently been used in my interdisciplinary lecture course on Computational Metaphysics (which won the central teaching award of FU Berlin)

- Quickly adopted by novice students
- Very good feedback from students
- Some impressive results from student projects