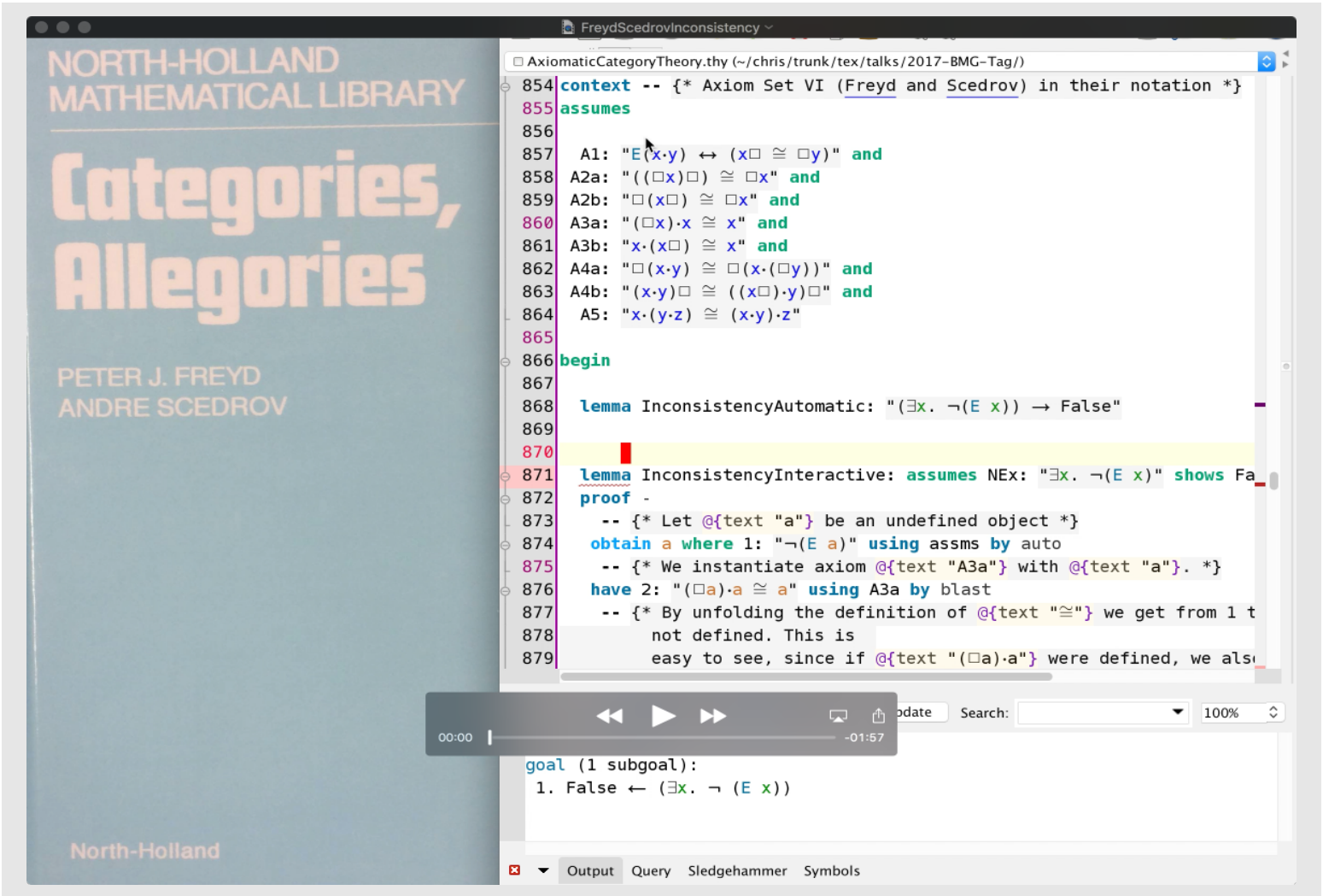
Experiments in Universal Logical Reasoning

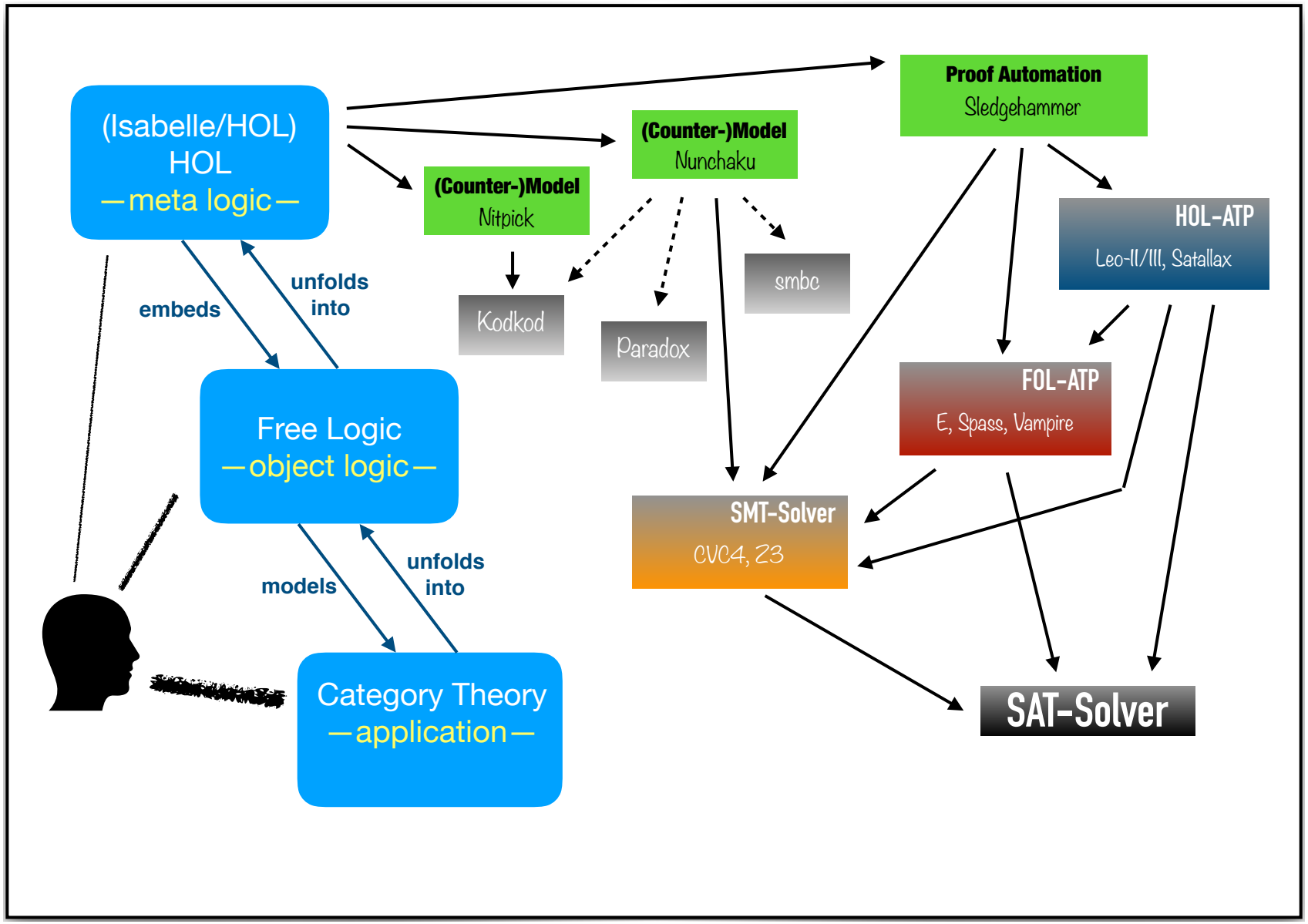
How to utilise ATPs and SMT solvers for the exploration of axiom systems for category theory in free logic?

Christoph Benzmüller (jww Dana Scott)



SMT 2018, July 12-13, Oxford, UK

Role of SMT and ATP in this work?



Presentation Outline

A Methodology: **Universal Logical Reasoning** in Metalogic HOL

B Instantiation: **Free Logic** in HOL

C Application: Exploration of **Axiom Systems for Category Theory**

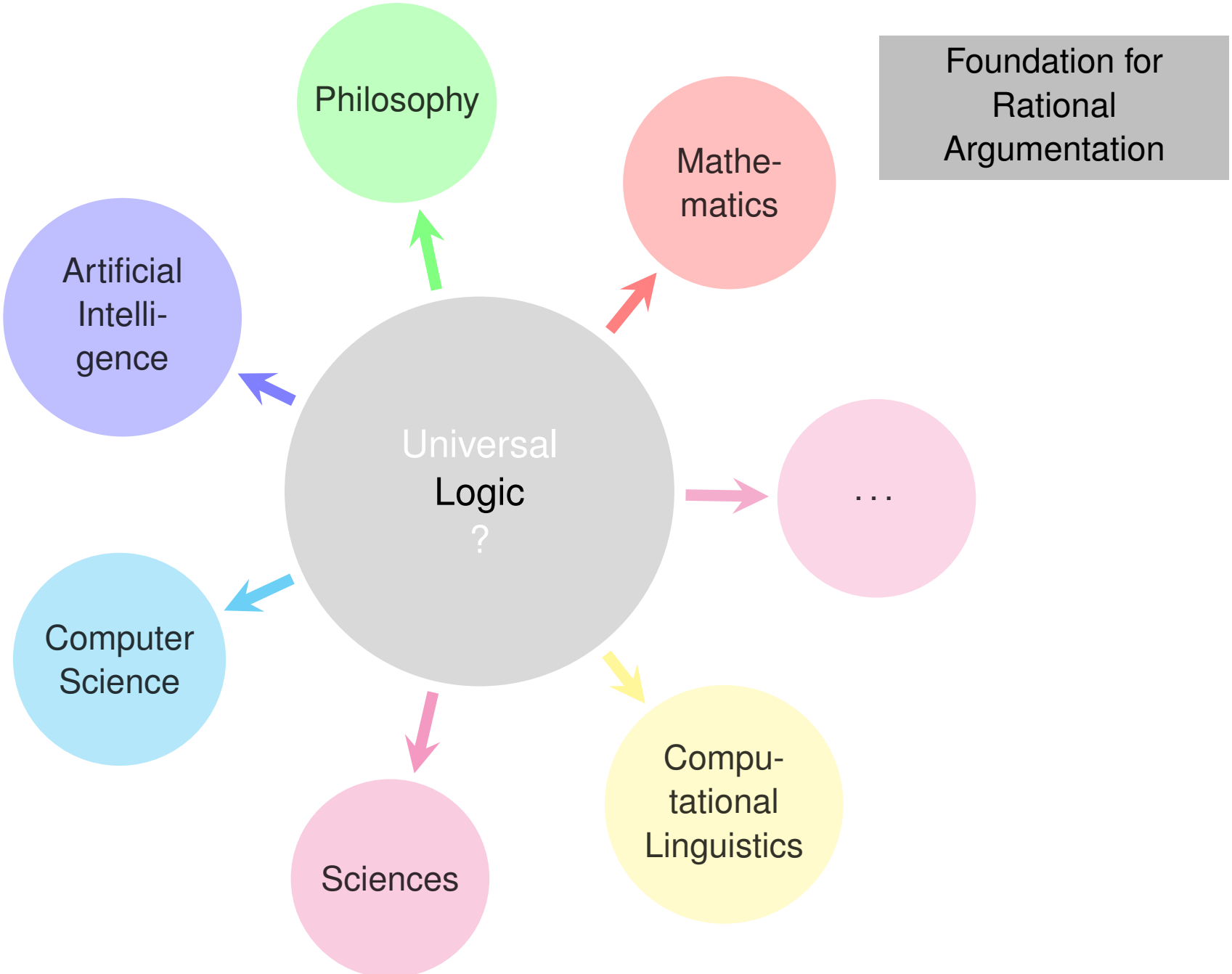
D Discussion: Role of **SMT solvers** and ATPs

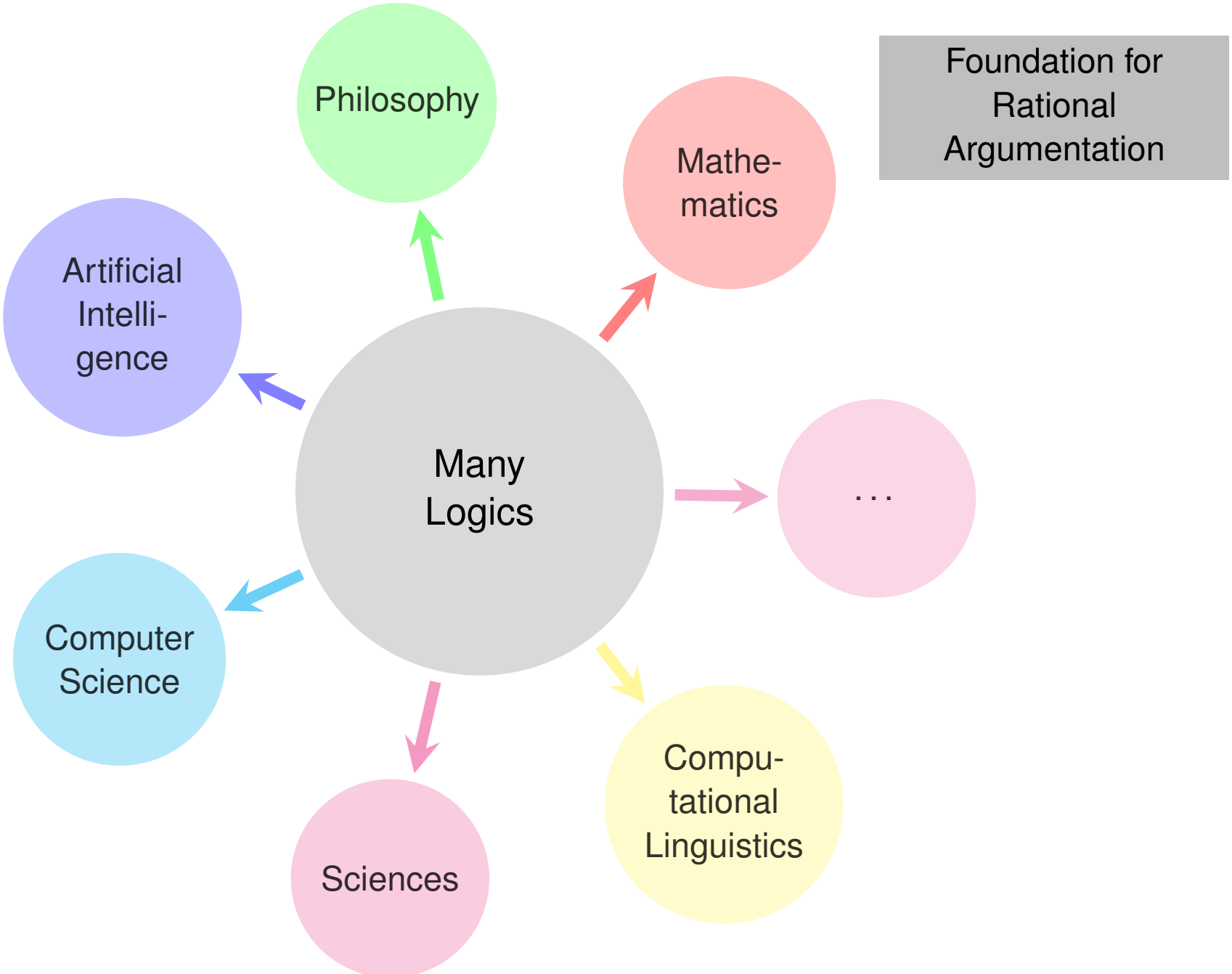
"If we had it [a *characteristica universalis*], we should be able to reason in metaphysics and morals in much the same way as in geometry and analysis."

(Leibniz, 1677)

Letter from Leibniz to Gallois, 1677 (GP VII, 21-22); translation by Russel, 1900

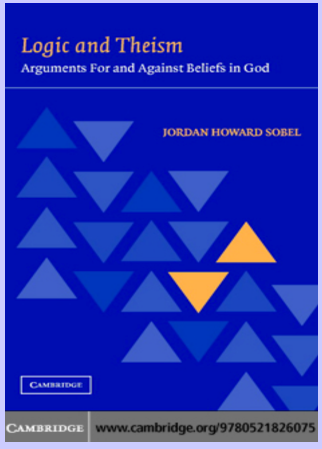
Part A — Methodology: Universal Logical Reasoning in Meta-Logic HOL (utilising Shallow Semantical Embeddings)

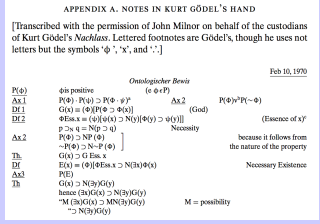

Philosophy
Mathe-
matics
Foundation for
Rational
Argumentation
Artificial
Intelli-
gence
Universal
Logic
?
. . .
Computer
Science
Sciences
Compu-
tational
Linguistics



C. Benzmüller (jww D. Scott), 2018

Logic Zoo



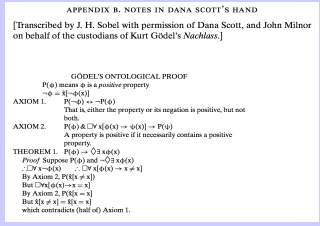

APPENDIX A. NOTES IN KURT GÖDEL'S HAND

[Transcribed with the permission of John Milnor on behalf of the custodians of Kurt Gödel's *Nachlass*. Lettered footnotes are Gödel's, though he uses not letters but the symbols '◆ ', '×', and ' .']

Feb 10, 1970

Ontologischer Beweis

P(φ)	φ is positive	(e φ⇔Pφ)	
Ax 1	P(¬φ) · P(φ) ⊃ P(Φ · ψ)ˣ	Ax 2	P(Φ)ᵃᵇP(¬ Φ)
Df 1	G(x) ≡ (φ)[P(Φ ⊃ Φ(x)]	(God)	
Df 2	Φ Ess.x ≡ (φ)[φ(x) ⊃ N(y)[Φ(y) ⊃ φ(y)]]		(Essence of xʸ
	P ⊃◇ q ≡ N(p ⊃ q)	Necessity	
Ax 2	P(Φ) ⊃ NP (Φ)		because it follows from the nature of the property
	¬P(Φ) ⊃ N¬P (Φ)		
Th.	G(x) ⊃ G Ess. x		
Df	E(x) ≡ (φ){Φ Ess.x ⊃ N(∃x)Φ(x)		Necessary Existence
Ax3	P(E)		
Th	G(x) ⊃ N(∃y)G(y)		
	hence (∃x)G(x) ⊃ N(∃y)G(y)		
	ˣM (∃x)G(x) ⊃ M(∃y)G(y)	M = possibility	
	ᵃᵇ⊃ N(∃y)G(y)		


APPENDIX B. NOTES IN DANA SCOTT'S HAND

[Transcribed by J. H. Sobel with permission of Dana Scott, and John Milnor on behalf of the custodians of Kurt Gödel's *Nachlass*.]

GÖDEL'S ONTOLOGICAL PROOF

P(φ) means φ is a *positive* property

$\neg\phi = \lambda[\neg\phi(x)]$

AXIOM 1. $P(\neg\phi) \leftrightarrow \neg P(\phi)$
That is, either the property or its negation is positive, but not both.

AXIOM 2. $P(\phi) \& \Box\forall x[\phi(x) \rightarrow \psi(x)] \rightarrow P(\psi)$
A property is positive if it necessarily contains a positive property.

THEOREM 1. $P(\phi) \rightarrow \Diamond \exists x \phi(x)$

Proof Suppose $P(\phi)$ and $\neg\Diamond\exists x\phi(x)$

$\therefore \Box\forall x \neg\phi(x) \quad \therefore \Box\forall x[\phi(x) \rightarrow x \neq x]$

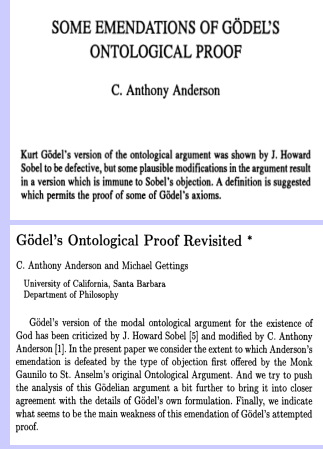
By Axiom 2, $P(\lambda[x \neq x])$

But $\Box\forall x[\phi(x) \rightarrow x = x]$

By Axiom 2, $P(\lambda[x = x])$

But $\lambda[x \neq x] = \lambda[x = x]$

which contradicts (half of) Axiom 1.


SOME EMENDATIONS OF GÖDEL'S ONTOLOGICAL PROOF

C. Anthony Anderson

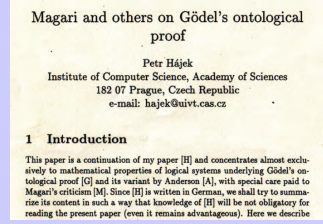
Kurt Gödel's version of the ontological argument was shown by J. Howard Sobel to be defective, but some plausible modifications in the argument result in a version which is immune to Sobel's objection. A definition is suggested which permits the proof of some of Gödel's axioms.

Gödel's Ontological Proof Revisited *

C. Anthony Anderson and Michael Gettings

University of California, Santa Barbara
Department of Philosophy

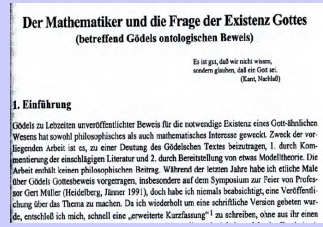
Gödel's version of the modal ontological argument for the existence of God has been criticized by J. Howard Sobel [5] and modified by C. Anthony Anderson [1]. In the present paper we consider the extent to which Anderson's emendation is defeated by the type of objection first offered by the Monk Gaunilo to St. Anselm's original Ontological Argument. And we try to push the analysis of this Gödelian argument a bit further to bring it into closer agreement with the details of Gödel's own formulation. Finally, we indicate what seems to be the main weakness of this emendation of Gödel's attempted proof.


Magari and others on Gödel's ontological proof

Petr Hájek
Institute of Computer Science, Academy of Sciences
182 07 Prague, Czech Republic
e-mail: hajek@uivt.cas.cz

1 Introduction

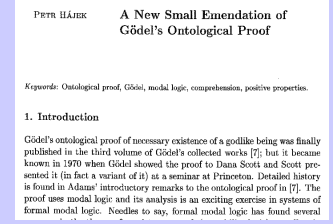
This paper is a continuation of my paper [H] and concentrates almost exclusively to mathematical properties of logical systems underlying Gödel's ontological proof [G] and its variant by Anderson [A], with special care paid to Magari's criticism [M]. Since [H] is written in German, we shall try to summarize its content in such a way that knowledge of [H] will be not obligatory for reading the present paper (even it remains advantageous). Here we describe


Der Mathematiker und die Frage der Existenz Gottes
(betreffend Gödels ontologischen Beweis)

Es ist gut, daß wir nicht wissen,
sondern glauben, daß ein Gott ist.
(Kant, Nachlaß)

1. Einführung

Gödels zu Lebzeiten unveröffentlichter Beweis für die notwendige Existenz eines Gott-ähnlichen Wesens hat sowohl philosophisches als auch mathematisches Interesse geweckt. Zweck der vorliegenden Arbeit ist es, zu einer Deutung des Gödelschen Textes beizutragen, i. durch Kommentierung der einschlägigen Literatur und 2. durch Bereitstellung von etwas Modelltheorie. Die Arbeit enthält keinen philosophischen Beitrag. Während der letzten Jahre habe ich etliche Male über Gödels Gottesbeweis vorgetragen, insbesondere auf dem Symposium zur Feier von Professor Gert Müller (Heidelberg, Jänner 1991), doch habe ich niemals beabsichtigt, eine Veröffentlichung über das Thema zu machen. Da ich wiederholt um eine schriftliche Version gebeten wurde, entschloß ich mich, schnell eine „erweiterte Kurzfassung"[1] zu schreiben, ohne aus ihr einen

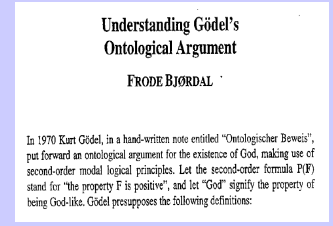

PETR HÁJEK

A New Small Emendation of Gödel's Ontological Proof

Keywords: Ontological proof, Gödel, modal logic, comprehension, positive properties.

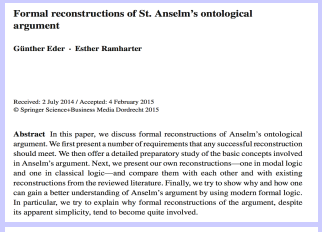
1. Introduction

Gödel's ontological proof of necessary existence of a godlike being was finally published in the third volume of Gödel's collected works [7]; but it became known in 1970 when Gödel showed the proof to Dana Scott and Scott presented it (in fact a variant of it) at a seminar at Princeton. Detailed history is found in Adams' introductory remarks to the ontological proof in [7]. The proof uses modal logic and its analysis is an exciting exercise in systems of formal modal logic. Needles to say, formal modal logic has found several


Understanding Gödel's Ontological Argument

FRODE BJØRDAL

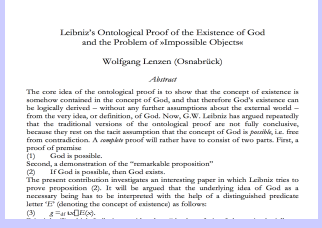
In 1970 Kurt Gödel, in a hand-written note entitled "Ontologischer Beweis", put forward an ontological argument for the existence of God, making use of second-order modal logical principles. Let the second-order formula P(F) stand for "the property F is positive", and let "God" signify the property of being God-like. Gödel presupposes the following definitions:


Formal reconstructions of St. Anselm's ontological argument

Günther Eder · Esther Ramharter

Received: 2 July 2014 / Accepted: 4 February 2015

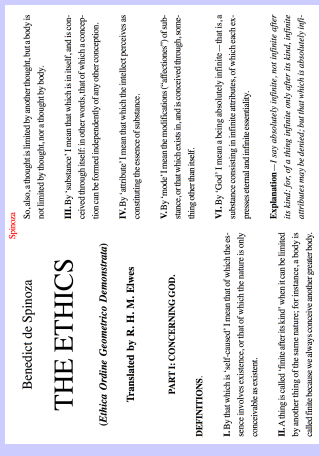
Abstract In this paper, we discuss formal reconstructions of Anselm's ontological argument. We first present a number of requirements that any successful reconstruction should meet. We then offer a detailed preparatory study of the basic concepts involved in Anselm's argument. Next, we present our own reconstructions—one in modal logic and one in classical logic—and compare them with each other and with existing reconstructions from the reviewed literature. Finally, we try to show why and how one can gain a better understanding of Anselm's argument by using modern formal logic. In particular, we try to explain why formal reconstructions of the argument, despite its apparent simplicity, tend to become quite involved.


Leibniz's Ontological Proof of the Existence of God
and the Problem of »Impossible Objects«

Wolfgang Lenzen (Osnabrück)

Abstract

The core idea of the ontological proof is to show that the concept of existence is somehow contained in the concept of God, and that therefore God's existence can be logically derived – without any further assumptions about the external world – from the very idea, or definition, of God. Now, G.W. Leibniz has argued repeatedly that the traditional versions of the ontological proof are not fully conclusive, because they rest on the tacit assumption that the concept of God is *possible*, i.e. free from contradiction. A *complete* proof will rather have to consist of two parts. First, a proof of premise
(1) God is possible.
Second, a demonstration of the "remarkable proposition"
(2) If God is possible, then God exists.
The present contribution investigates an interesting paper in which Leibniz tries to prove proposition (2). It will be argued that the underlying idea of God as a necessary being has to be interpreted with the help of a distinguished predicate letter 'E' (denoting the concept of existence) as follows:
(3) $g =_{df} \iota x[E(x)]$.


Spinoza

Benedict de Spinoza

THE ETHICS

(*Ethica Ordine Geometrico Demonstrata*)

Translated by R. H. M. Elwes

PART I: CONCERNING GOD.

DEFINITIONS.

So, also, a thought is limited by another thought, but a body is not limited by thought, nor a thought by body.

III. By 'substance' I mean that which is in itself, and is conceived through itself: in other words, that of which a conception can be formed independently of any other conception.

IV. By 'attribute' I mean that which the intellect perceives as constituting the essence of substance.

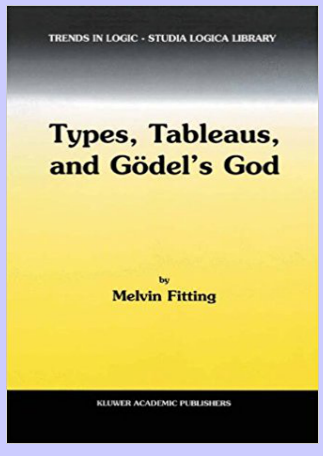
V. By 'mode' I mean the modifications ("affections") of substance, or that which exists in, and is conceived through, something other than itself.

VI. By 'God' I mean a being absolutely infinite—that is, a substance consisting in infinite attributes, of which each expresses eternal and infinite essentiality.

Explanation—*I say absolutely infinite, not infinite after its kind: for, of a thing infinite only after its kind, infinite attributes may be denied; but that which is absolutely infi-*

I. By that which is "self-caused" I mean that of which the essence involves existence, or that of which the nature is only conceivable as existent.

II. A thing is called "finite after its kind" when it can be limited by another thing of the same nature: for instance, a body is called finite because we always conceive another greater body.


TRENDS IN LOGIC - STUDIA LOGICA LIBRARY

Types, Tableaus, and Gödel's God

by

Melvin Fitting

KLUWER ACADEMIC PUBLISHERS

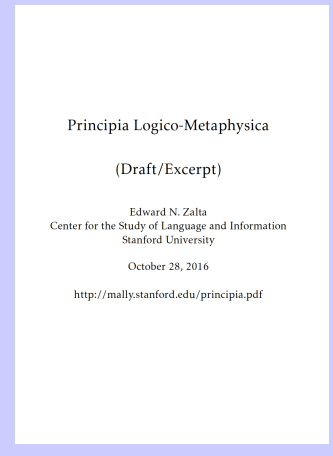

SYNTHESE LIBRARY / VOLUME 160

EDWARD N. ZALTA

ABSTRACT OBJECTS

An Introduction to Axiomatic Metaphysics

D. REIDEL PUBLISHING COMPANY
DORDRECHT / BOSTON / LANCASTER

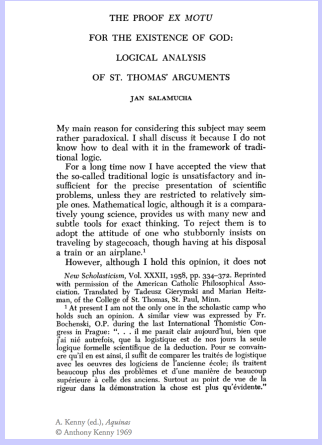

Principia Logico-Metaphysica

(Draft/Excerpt)

Edward N. Zalta
Center for the Study of Language and Information
Stanford University

October 28, 2016

http://mally.stanford.edu/principia.pdf


THE PROOF *EX MOTU*

FOR THE EXISTENCE OF GOD:

LOGICAL ANALYSIS

OF ST. THOMAS' ARGUMENTS

JAN SALAMUCHA

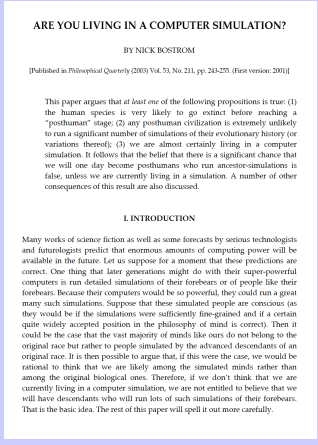
My main reason for considering this subject may seem rather paradoxical. I shall discuss it because I do not know how to deal with it in the framework of traditional logic.

For a long time now I have accepted the view that the so-called traditional logic is unsatisfactory and insufficient for the precise presentation of scientific problems, unless they are restricted to relatively simple ones. Mathematical logic, although it is a comparatively young science, provides us with many new and subtle tools for exact thinking. To reject them is to adopt the attitude of one who stubbornly insists on traveling by stagecoach, though having at his disposal a train or an airplane.[1]

However, although I hold this opinion, it does not

Nova Scholastica, Vol. XXII, 1958, pp. 334–372. Reprinted with permission of the American Catholic Philosophical Association. Translated by Tadeusz Gieymski and Marian Heitzman, of the College of St. Thomas, St. Paul, Minn.

[1] At present I am not the only one in the scholastic camp who holds such an opinion. A similar view was expressed by Fr. Bocheński, O.P. during the last International Thomistic Congress in Prague: ". . . il me paraît clair aujourd'hui, bien que j'ai nié autrefois, que la logistique est de nos jours la seule logique formelle scientifique de la déduction. Pour se convaincre qu'il en est ainsi, il suffit de comparer les traités de logistique avec les oeuvres des logiciens de l'ancienne école; ils traitent beaucoup plus des problèmes et d'une manière de beaucoup supérieure à celle des anciens. Surtout au point de vue de la rigueur dans la démonstration la chose est plus qu'évidente."


ARE YOU LIVING IN A COMPUTER SIMULATION?

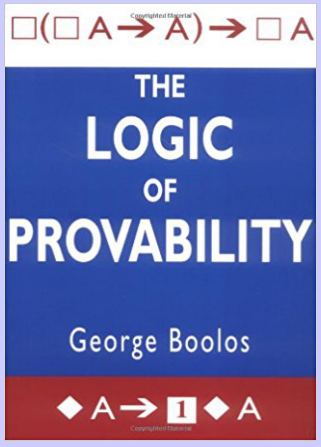
BY NICK BOSTROM

[Published in *Philosophical Quarterly* (2003) Vol. 53, No. 211, pp. 243-255. (First version: 2001)]

This paper argues that *at least one* of the following propositions is true: (1) the human species is very likely to go extinct before reaching a "posthuman" stage; (2) any posthuman civilization is extremely unlikely to run a significant number of simulations of their evolutionary history (or variations thereof); (3) we are almost certainly living in a computer simulation. It follows that the belief that there is a significant chance that we will one day become posthumans who run ancestor-simulations is false, unless we are currently living in a simulation. A number of other consequences of this result are also discussed.

I. INTRODUCTION

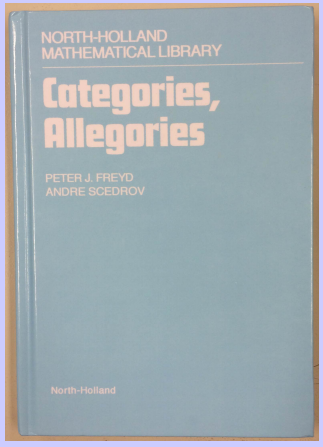
Many works of science fiction as well as some forecasts by serious technologists and futurologists predict that enormous amounts of computing power will be available in the future. Let us suppose for a moment that these predictions are correct. One thing that later generations might do with their super-powerful computers is run detailed simulations of their forebears or of people like their forebears. Because their computers would be so powerful, they could run a great many such simulations. Suppose that these simulated people are conscious (as they would be if the simulations were sufficiently fine-grained and if a certain quite widely accepted position in the philosophy of mind is correct). Then it could be the case that the vast majority of minds like ours do not belong to the original race but rather to people simulated by the advanced descendants of an original race. It is then possible to argue that, if this were the case, we would be rational to think that we are likely among the simulated minds rather than among the original biological ones. Therefore, if we don't think that we are currently living in a computer simulation, we are not entitled to believe that we will have descendants who will run lots of such simulations of their forebears. That is the basic idea. The rest of this paper will spell it out more carefully.


□(□A→A)→□A

THE LOGIC OF PROVABILITY

George Boolos

◆A→1◆A

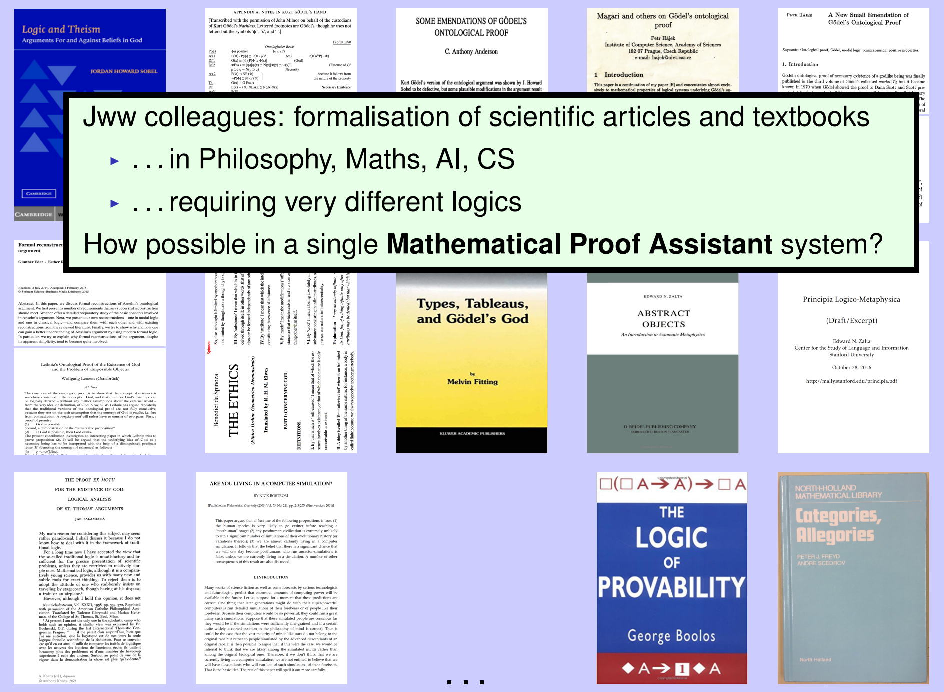

NORTH-HOLLAND MATHEMATICAL LIBRARY

Categories, Allegories

PETER J. FREYD
ANDRE SCEDROV

North-Holland

C. Benzmüller (jww D. Scott), 2018

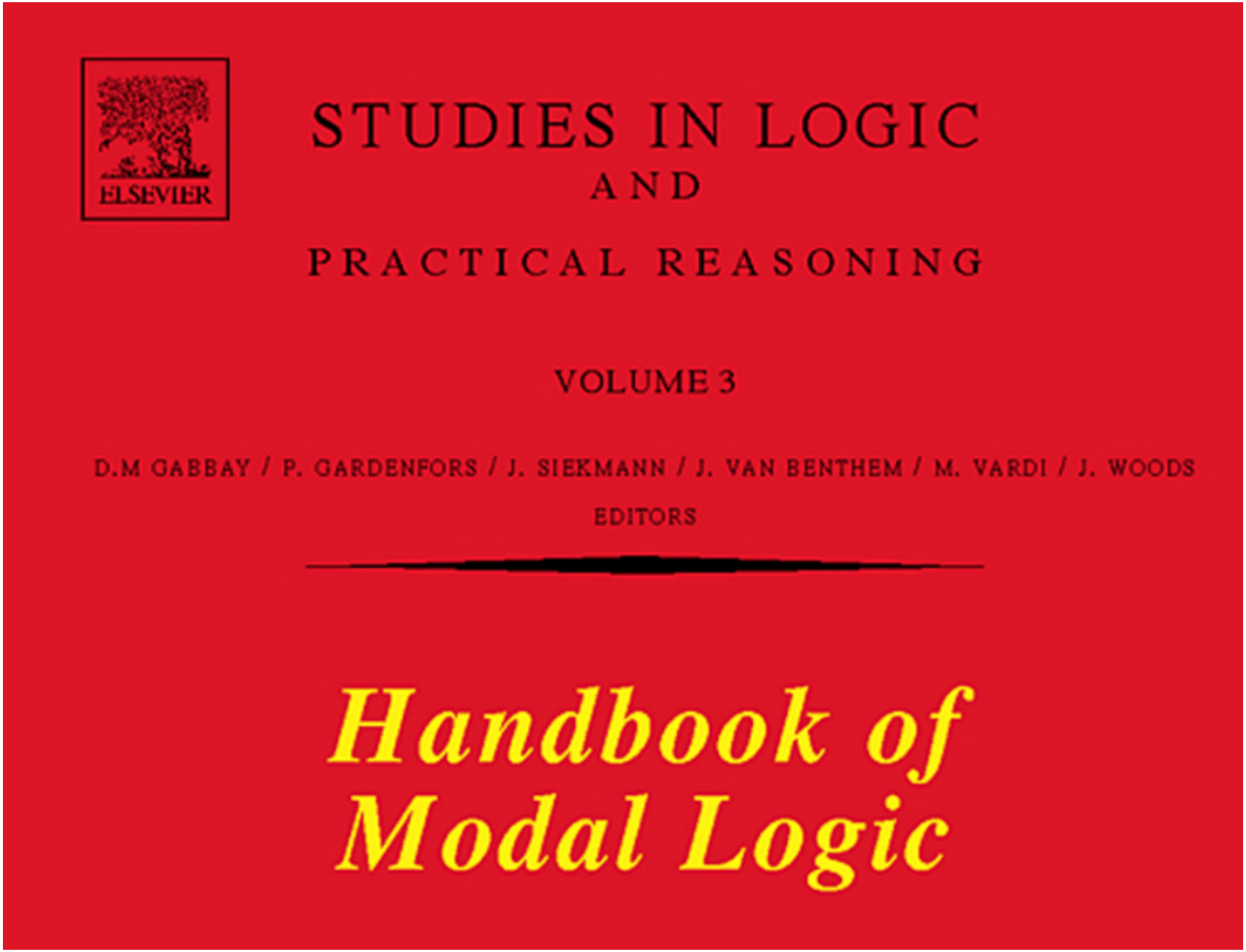


Jww colleagues: formalisation of scientific articles and textbooks

- . . . in Philosophy, Maths, AI, CS
- . . . requiring very different logics

How possible in a single **Mathematical Proof Assistant** system?

Example: Modal Logic Textbook



Example: Modal Logic Textbook

2 BASIC MODAL LOGIC

In this section we introduce the basic modal language and its relational semantics. We define basic modal syntax, introduce models and frames, and give the satisfaction definition. We then draw the reader's attention to the internal perspective that modal languages offer on relational structure, and explain why models and frames should be thought of as graphs. Following this we give the standard translation. This enables us to convert any basic modal formula into a first-order formula with one free variable. The standard translation is a bridge between the modal and classical worlds, a bridge that underlies much of the work of this chapter.

2.1 *First steps in relational semantics*

Suppose we have a set of proposition symbols (whose elements we typically write as p, q, r and so on) and a set of modality symbols (whose elements we typically write as m, m', m'', and so on). The choice of PROP and MOD is called the *signature* (or *similarity type*) of the language; in what follows we'll tacitly assume that PROP is denumerably infinite, and we'll often work with signatures in which MOD contains only a single element. Given a signature, we define the *basic modal language* (over the signature) as follows:

$$\varphi \quad ::= \quad p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \varphi \leftrightarrow \psi \mid \langle m \rangle \varphi \mid [m]\varphi.$$

That is, a basic modal formula is either a proposition symbol, a boolean constant, a boolean combination of basic modal formulas, or (most interesting of all) a formula prefixed by a diamond

Example: Modal Logic Textbook

2 BASIC MODAL LOGIC

In this section we introduce the basic modal language and its relational semantics. We define basic modal syntax, introduce models and frames, and give the satisfaction definition. We then draw the reader's attention to the internal perspective that modal languages offer on relational structure, and explain why models and frames should be thought of as graphs. Following this we give the standard translation. This enables us to convert any basic modal formula into a first-order formula with one free variable. The standard translation is a bridge between the modal and classical worlds, a bridge that underlies much of the work of this chapter.

2.1 First steps in relational semantics

Syntax

Metalanguage

ose elements we typically write as p, q, r and
ents we typically write as m, m', m'', and so
nature (or *similarity type*) of the language; in
what follows we'll tacitly assume that PROP is denumerably infinite, and we'll often work with signatures in which MOD contains only a single element. Given a signature, we define the *basic modal language* (over the signature) as follows:

$$\varphi \quad ::= \quad p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \varphi \leftrightarrow \psi \mid \langle m \rangle \varphi \mid [m]\varphi.$$

That is, a basic modal formula is either a proposition symbol, a boolean constant, a boolean combination of basic modal formulas, or (most interesting of all) a formula prefixed by a diamond

Example: Modal Logic Textbook

A *model* (or *Kripke model*) $\mathfrak{M}$ for the basic modal language (over some fixed signature) is a triple $\mathfrak{M} = (W, \{R^m\}_{m \in \text{MOD}}, V)$. Here W, the *domain*, is a non-empty set, whose elements we usually call *points*, but which, for reasons which will soon be clear, are sometimes called *states*, *times*, *situations*, *worlds* and other things besides. Each R^m in a model is a binary relation on W, and V is a function (the valuation) that assigns to each proposition symbol p in PROP a subset $V(p)$ of W; think of $V(p)$ as the set of points in $\mathfrak{M}$ where p is true. The first two components $(W, \{R^m\}_{m \in \text{MOD}})$ of $\mathfrak{M}$ are called the *frame* underlying the model. If there is only one relation in the model, we typically write (W, R) for its frame, and (W, R, V) for the model itself. We encourage the reader to think of Kripke models as graphs (or to be slightly more precise, *directed graphs*, that is, graphs whose points are linked by directed arrows) and will shortly give some examples which show why this is helpful.

Suppose w is a point in a model $\mathfrak{M} = (W, \{R^m\}_{m \in \text{MOD}}, V)$. Then we inductively define the notion of a formula φ being *satisfied* (or *true*) in $\mathfrak{M}$ at point w as follows (we omit some of the clauses for the booleans):

$\mathfrak{M}, w \models p$	iff	$w \in V(p)$,
$\mathfrak{M}, w \models \top$		always,
$\mathfrak{M}, w \models \bot$		never,
$\mathfrak{M}, w \models \neg\varphi$	iff	not $\mathfrak{M}, w \models \varphi$ (notation: $\mathfrak{M}, w \not\models \varphi$),
$\mathfrak{M}, w \models \varphi \wedge \psi$	iff	$\mathfrak{M}, w \models \varphi$ and $\mathfrak{M}, w \models \psi$,
$\mathfrak{M}, w \models \varphi \rightarrow \psi$	iff	$\mathfrak{M}, w \not\models \varphi$ or $\mathfrak{M}, w \models \psi$,
$\mathfrak{M}, w \models \langle m \rangle \varphi$	iff	for some $v \in W$ such that $R^m wv$ we have $\mathfrak{M}, v \models \varphi$,
$\mathfrak{M}, w \models [m] \varphi$	iff	for all $v \in W$ such that $R^m wv$ we have $\mathfrak{M}, v \models \varphi$.

Example: Modal Logic Textbook

A *model* (or *Kripke model*) $\mathfrak{M}$ for the basic modal language (over some fixed signature) is a triple $\mathfrak{M} = (W, \{R^m\}_{m \in \text{MOD}}, V)$. Here W, the *domain*, is a non-empty set, whose elements we usually call *points*, but which, for reasons which will soon be clear, are sometimes called *states*, *times*

in a model is a binary relation on W,

and

position symbol p in PROP a subset

$V(p)$

p is true. The first two components

$(W, \{$

model. If there is only one relation

in the

(W, R, V) for the model itself. We encourage the reader to think of Kripke models as graphs (or to be slightly more precise, *directed graphs*, that is, graphs whose points are linked by directed arrows) and will shortly give some examples which show why this is helpful.

Metalanguage

Suppose w is a point in a model $\mathfrak{M} = (W, \{R^m\}_{m \in \text{MOD}}, V)$. Then we inductively define the notion of a formula φ being *satisfied* (or *true*) in $\mathfrak{M}$ at point w as follows (we omit some of the clauses for the booleans):

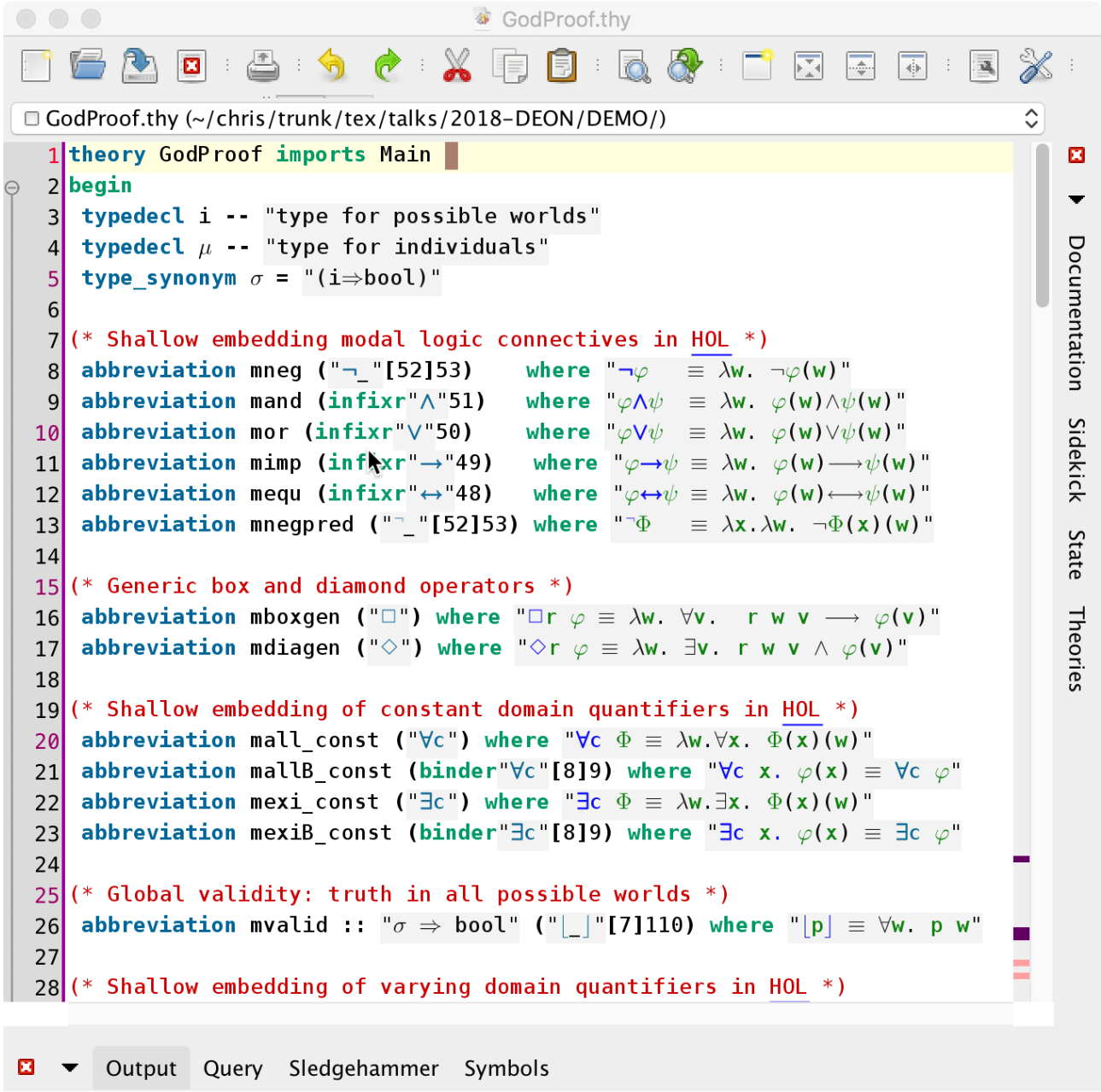
Semantics

$\mathfrak{M}, w \models p$	iff	$w \in V(p)$,
$\mathfrak{M}, w \models \top$		always,
$\mathfrak{M}, w \models \bot$		never,
$\mathfrak{M}, w \models \neg\varphi$	iff	not $\mathfrak{M}, w \models \varphi$ (notation: $\mathfrak{M}, w \not\models \varphi$),
$\mathfrak{M}, w \models \varphi \wedge \psi$	iff	$\mathfrak{M}, w \models \varphi$ and $\mathfrak{M}, w \models \psi$,
$\mathfrak{M}, w \models \varphi \rightarrow \psi$	iff	$\mathfrak{M}, w \not\models \varphi$ or $\mathfrak{M}, w \models \psi$,
$\mathfrak{M}, w \models \langle m \rangle \varphi$	iff	for some $v \in W$ such that $R^m wv$ we have $\mathfrak{M}, v \models \varphi$,
$\mathfrak{M}, w \models [m]\varphi$	iff	for all $v \in W$ such that $R^m wv$ we have $\mathfrak{M}, v \models \varphi$.

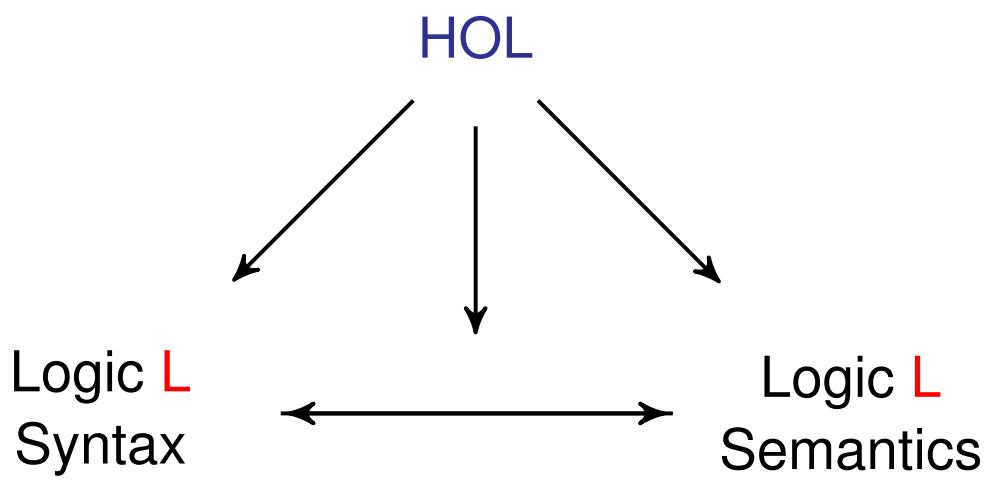
Example: Modal Logic Textbook

$$
\begin{array}{rll}
\mathfrak{M}, w \models p & \text{iff} & w \in V(p), \\
\mathfrak{M}, w \models \top & & \text{always}, \\
\mathfrak{M}, w \models \bot & & \text{never}, \\
\mathfrak{M}, w \models \neg\varphi & \text{iff} & \text{not } \mathfrak{M}, w \models \varphi \text{ (notation: } \mathfrak{M}, w \not\models \varphi), \\
\mathfrak{M}, w \models \varphi \wedge \psi & \text{iff} & \mathfrak{M}, w \models \varphi \text{ and } \mathfrak{M}, w \models \psi, \\
\mathfrak{M}, w \models \varphi \rightarrow \psi & \text{iff} & \mathfrak{M}, w \not\models \varphi \text{ or } \mathfrak{M}, w \models \psi, \\
\mathfrak{M}, w \models \langle m \rangle \varphi & \text{iff} & \text{for some } v \in W \text{ such that } R^m wv \text{ we have } \mathfrak{M}, v \models \varphi, \\
\mathfrak{M}, w \models [m]\varphi & \text{iff} & \text{for all } v \in W \text{ such that } R^m wv \text{ we have } \mathfrak{M}, v \models \varphi.
\end{array}
$$

Universal Logical Reasoning in Isabelle/HOL



Universal Logical Reasoning in HOL



Examples for L we have already studied:
Intuitionistic Logics, Modal Logics, Description Logics, Conditional Logics, Access Control Logics, Hybrid Logics, Multivalued Logics, Paraconsistent Logics, **Hyper-intensional Higher-Order Modal Logic**, **Free Logic**, **Dyadic Deontic Logic**, **Input/Output Logic**, ...

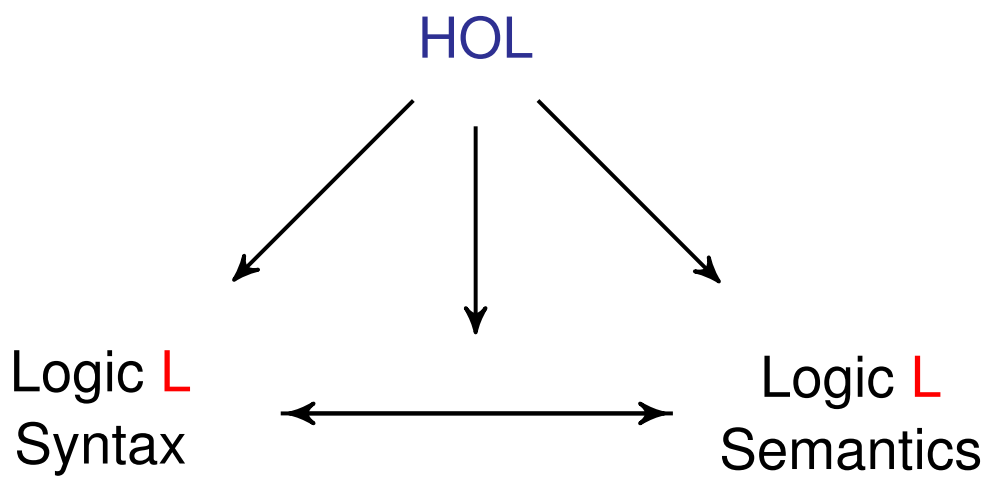
Embedding works also for quantifiers (first-order & higher-order)

HOL provers become universal logic reasoning engines!

interactive: Isabelle/HOL, PVS, HOL4, Hol Light, Coq/HOL, ...

automated: Leo-III, LEO-II, Satallax, TPS, Nitpick, Isabelle/HOL, ...

Universal Logical Reasoning in HOL



Examples for L we have already studied:
Intuitionistic Logics, Modal Logics, Description Logics, Conditional Logics, Access Control Logics, Hybrid Logics, Multivalued Logics, Paraconsistent Logics, **Hyper-intensional Higher-Order Modal Logic**, **Free Logic**, **Dyadic Deontic Logic**, **Input/Output Logic**, . . .

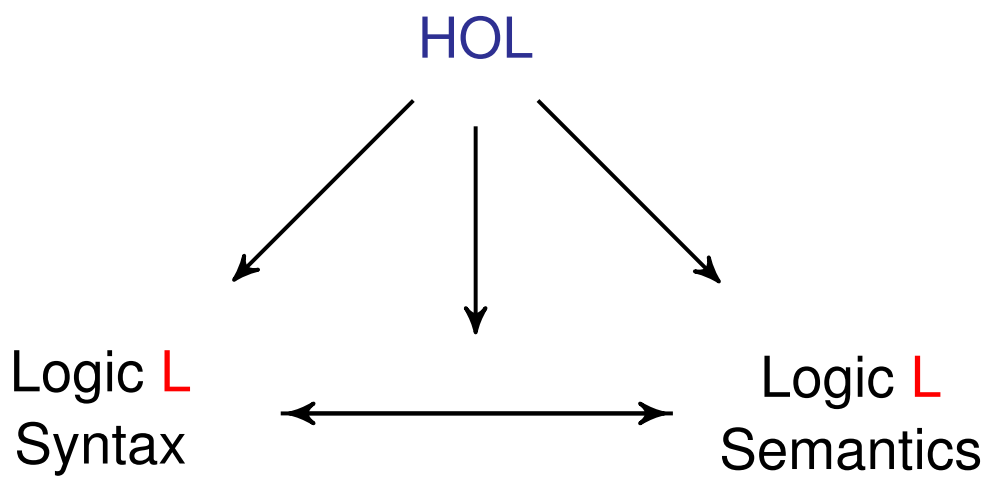
Embedding works also for quantifiers (first-order & higher-order)

HOL provers become universal logic reasoning engines!

interactive: Isabelle/HOL, PVS, HOL4, Hol Light, Coq/HOL, . . .

automated: Leo-III, LEO-II, Satallax, TPS, Nitpick, Isabelle/HOL, . . .

Universal Logical Reasoning in HOL



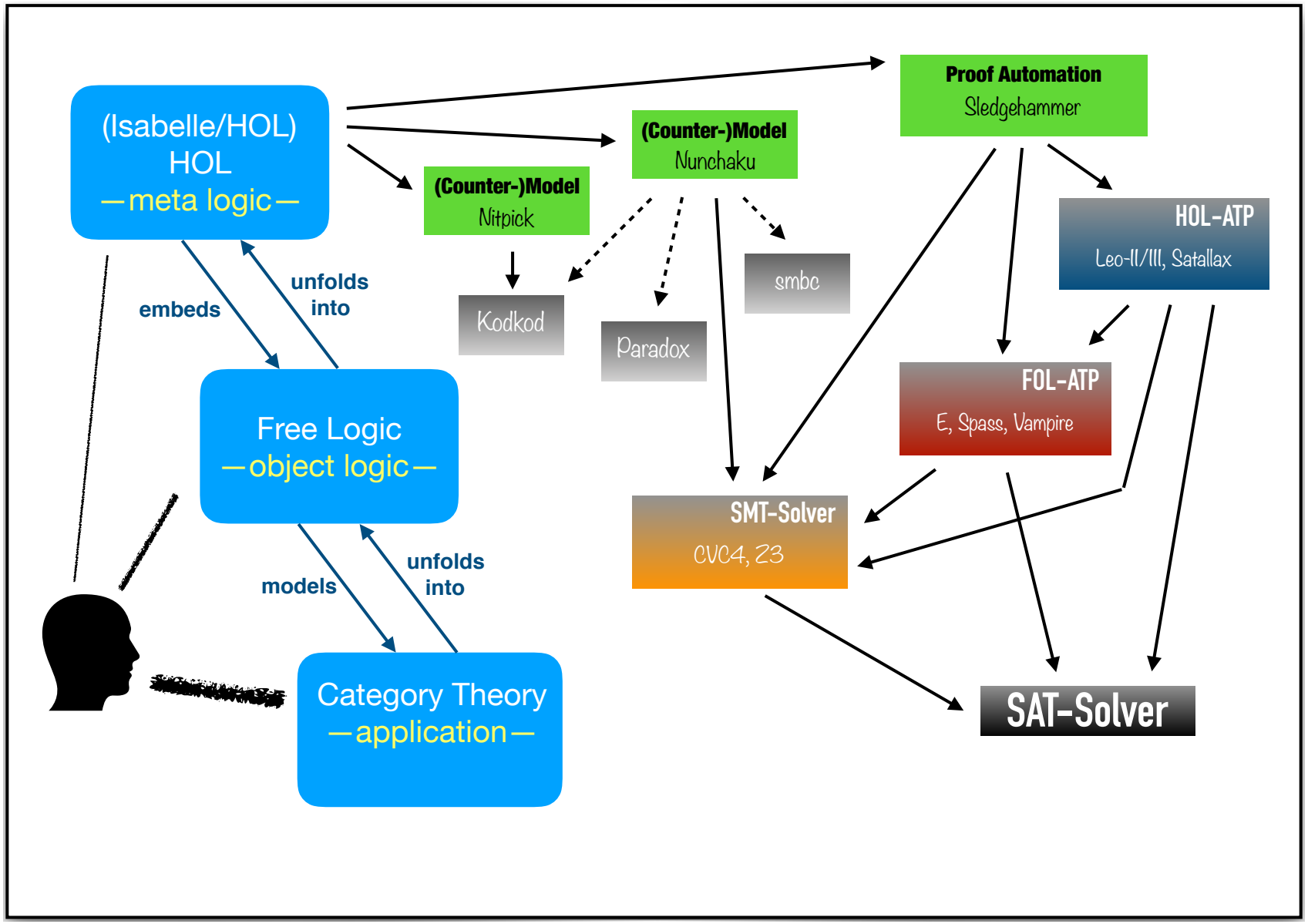
Examples for L we have already studied:
Intuitionistic Logics, Modal Logics, Description Logics, Conditional Logics, Access Control Logics, Hybrid Logics, Multivalued Logics, Paraconsistent Logics, **Hyper-intensional Higher-Order Modal Logic**, **Free Logic**, **Dyadic Deontic Logic**, **Input/Output Logic**, . . .

Embedding works also for quantifiers (first-order & higher-order)

HOL provers become universal logic reasoning engines!

interactive:	Isabelle/HOL, PVS, HOL4, Hol Light, Coq/HOL, . . .
automated:	Leo-III, LEO-II, Satallax, TPS, Nitpick, Isabelle/HOL, . . .

Role of SMT and ATP in this work?



Part B — Instantiation: Free Logic in HOL

[Free Logic in Isabelle/HOL, ICMS, 2016]
[Axiomatizing Category Theory in Free Logic, arXiv:1609.01493, 2016 (submitted preprint)]
[Axiom Systems for Category Theory in Free Logic, Archive of Formal Proofs, 2018]

Free Logic: Elegant Approach to Definite Description and Undefinedness

Dana Scott. "Existence and description in formal logic." In: Bertrand Russell: Philosopher of the Century, edited by R. Schoenman. George Allen & Unwin, London, 1967, pp. 181-200. Reprinted with additions in: Philosophical Application of Free Logic, edited by K. Lambert. Oxford Universitry Press, 1991, pp. 28 - 48.

16

DANA SCOTT

Existence and Description in Formal Logic

The problem of what to do with improper descriptive phrases has bothered logicians for a long time. There have been three major suggestions of how to treat descriptions usually associated with the names of Russell, Frege and Hilbert-Bernays. The author does not consider any of these approaches really satisfactory. In many ways Russell's idea is most attractive because of its simplicity. However, on second thought one is saddened to find that the Russellian method of elimination depends heavily on the scope of the elimination.

Previous Approaches (rough sketch)

The present King of France is bald.

.

Russel (first approach) $pkof$ **:= present King of France**

$bald(\iota x.pkof(x))$

iff

$(\exists x.pkof(x)) \wedge (\forall x, y.((pkof(x) \wedge pkof(y)) \rightarrow x = y) \wedge (\forall x.pkof((x) \rightarrow bald(x))$

Hence, **false**.

Frege

$\iota x.pkof(x)$ does not denote; $bald(\iota x.pkof(x))$ has **no truth value**.

Hilbert-Bernays

If the existence and uniqueness conditions cannot be proved, then the term $\iota x.pkof(x)$ is **not part of the language**.

Previous Approaches (rough sketch)

The present King of France is bald.

.

Russel (first approach) $pkof$:= present King of France

$bald(\iota x.pkof(x))$

iff

$(\exists x.pkof(x)) \wedge (\forall x, y.((pkof(x) \wedge pkof(y)) \rightarrow x = y) \wedge (\forall x.pkof((x) \rightarrow bald(x))$

Hence, **false**.

Frege

$\iota x.pkof(x)$ does not denote; $bald(\iota x.pkof(x))$ has **no truth value**.

Hilbert-Bernays

If the existence and uniqueness conditions cannot be proved, then the term $\iota x.pkof(x)$ is **not part of the language**.

Previous Approaches (rough sketch)

The present King of France is bald.

.

Russel (first approach)

$pkof$:= present King of France

$bald(\iota x.pkof(x))$

iff

$(\exists x.pkof(x)) \wedge (\forall x, y.((pkof(x) \wedge pkof(y)) \rightarrow x = y) \wedge (\forall x.pkof((x) \rightarrow bald(x))$

Hence, **false**.

Frege

$\iota x.pkof(x)$ does not denote; $bald(\iota x.pkof(x))$ has **no truth value**.

Hilbert-Bernays

If the existence and uniqueness conditions cannot be proved, then the term $\iota x.pkof(x)$ is **not part of the language**.

Free Logic: Elegant Approach to Definite Description and Undefinedness

Existence and Description in Formal Logic (Dana Scott), 1967

Principle 1: Bound individual variables range over domain $E \subset D$

Principle 2: Values of terms and free variables are in D, not necessarily in E only.

Principle 3: Domain E may be empty

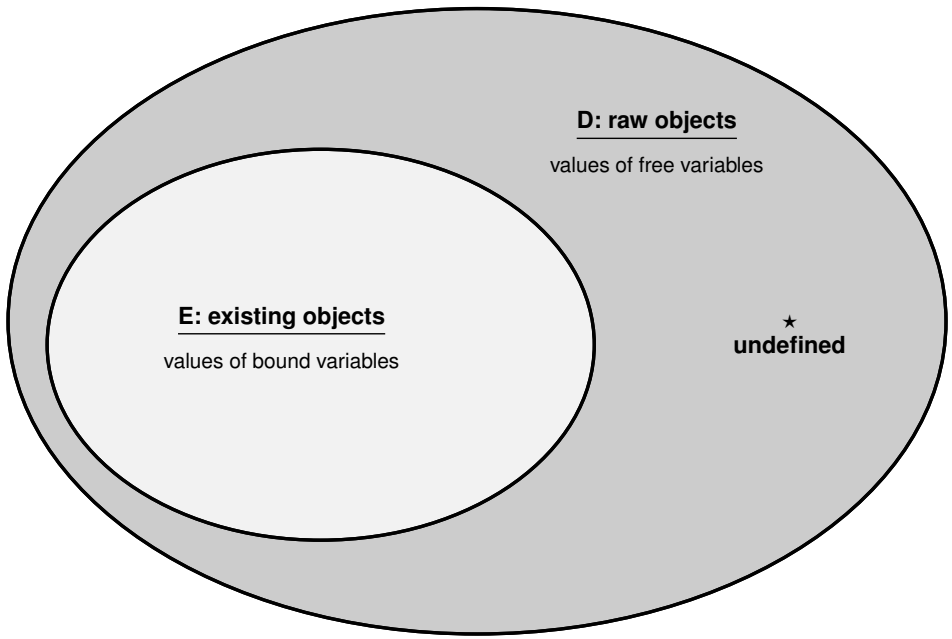
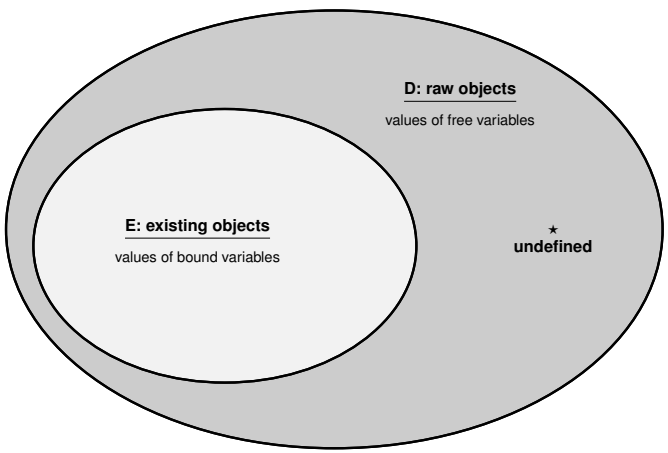


Figure: Illustration of the semantical domains of free logic

Free Logic in HOL



FreeFOLminimal.thy

FreeFOLminimal.thy (~/GITHUBS/PrincipiaMetaphysica/FreeLogic/2016-ICMS/)

```
typedecl i   — "the type for indiviuals"
consts fExistence:: "i⇒bool" ("E")    — "Existence predicate"
consts fStar:: "i" ("⋆")   — "Distinguished symbol for undefinedness"

axiomatization where fStarAxiom: "¬E(⋆)"

abbreviation fNot:: "bool⇒bool" ("¬")
 where "¬φ ≡ ¬φ"
abbreviation fImplies:: "bool⇒bool⇒bool" (infixr "→" 49)
 where "φ→ψ ≡ φ⟶ψ"
abbreviation fForall:: "(i⇒bool)⇒bool" ("∀")
 where "∀Φ ≡ ∀x. E(x)⟶Φ(x)"
abbreviation fForallBinder:: "(i⇒bool)⇒bool" (binder "∀" [8] 9)
 where "∀x. φ(x) ≡ ∀φ"
abbreviation fThat:: "(i⇒bool)⇒i" ("I")
 where "IΦ ≡ if ∃x. E(x) ∧ Φ(x) ∧ (∀y. (E(y) ∧ Φ(y)) ⟶ (y = x))
              then THE x. E(x) ∧ Φ(x)
              else ⋆"
abbreviation fThatBinder:: "(i⇒bool)⇒i"  (binder "I" [8] 9)
 where "Ix. φ(x) ≡ I(φ)"
abbreviation fOr (infixr "∨" 51) where "φ∨ψ ≡ (¬φ)→ψ"
abbreviation fAnd (infixr "∧" 52) where "φ∧ψ ≡ ¬(¬φ∨¬ψ)"
abbreviation fEquiv (infixr "↔" 50) where "φ↔ψ ≡ (φ→ψ)∧(ψ→φ)"
abbreviation fEquals (infixr "=" 56) where "x=y ≡ x=y"
abbreviation fExists ("∃") where "∃Φ ≡ ¬(∀(λy.¬(Φ y)))"
abbreviation fExistsBinder (binder "∃" [8]9) where "∃x. φ(x) ≡ ∃φ"
```

Documentation Sidekick State Theories

Proof state Auto update Update Search: 100%

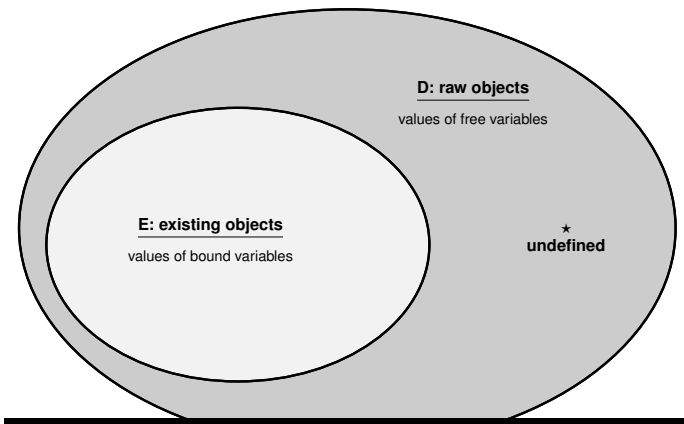
```
consts
  fForall :: "(i ⇒ bool) ⇒ bool"
```

Output Query Sledgehammer Symbols

17,24 (511/4534) (isabelle,isabelle,UTF-8-Isabelle)Nmro UG 548/781MB 1:36 AM

Free Logic in HOL

```
abbreviation fForall ("∀") (*Free universal quantification*)
  where "∀Φ ≡ ∀x. E x ⟶ Φ x"
abbreviation fForallBinder (binder "∀" [8] 9) (*Binder notation*)
  where "∀x. φ x ≡ ∀φ"
```

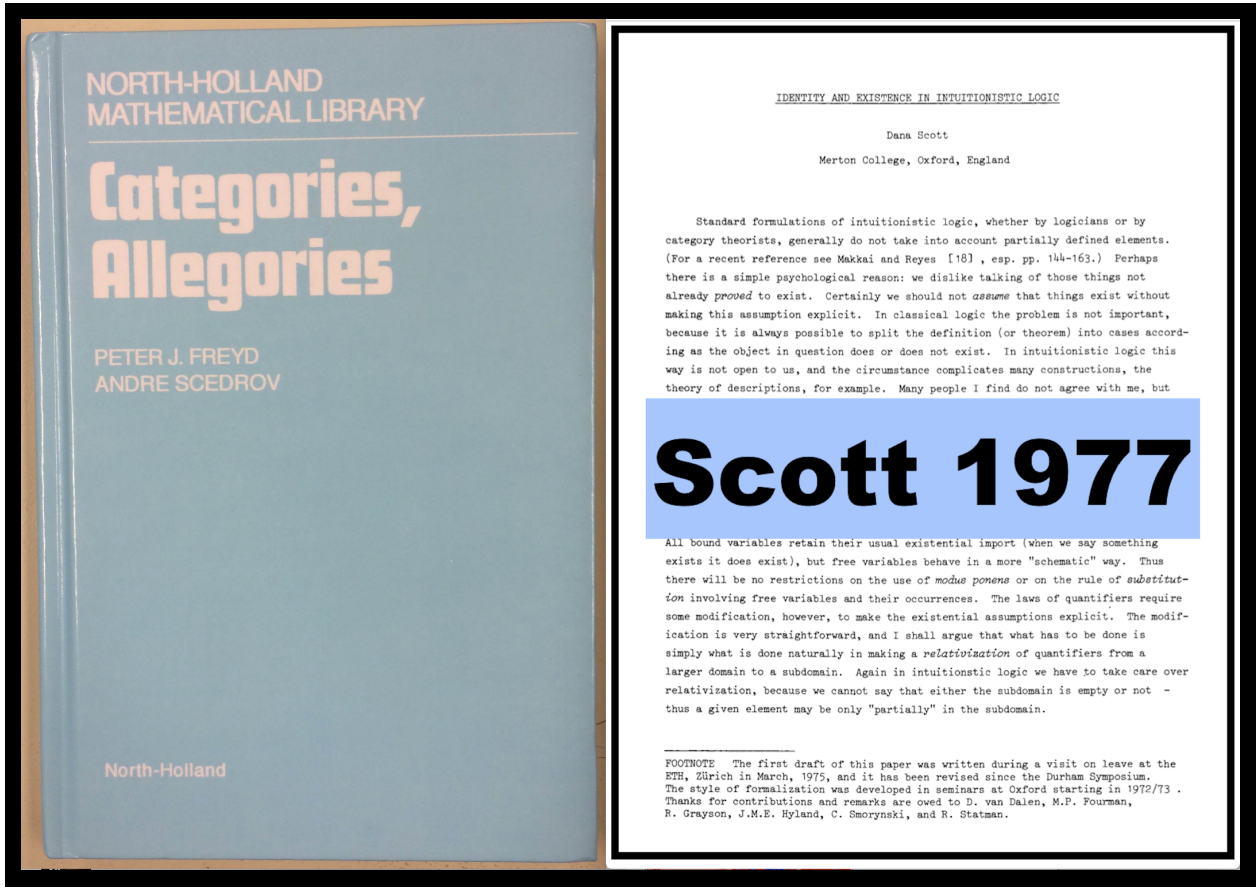


FreeFOLminimal.thy

```
abbreviation fImplies:: "bool⇒bool⇒bool" (infixr "→" 49)
 where "φ→ψ ≡ φ⟶ψ"
abbreviation fForall:: "(i⇒bool)⇒bool" ("∀")
 where "∀Φ ≡ ∀x. E(x)⟶Φ(x)"
abbreviation fForallBinder:: "(i⇒bool)⇒bool" (binder "∀" [8] 9)
 where "∀x. φ(x) ≡ ∀φ"
abbreviation fThat:: "(i⇒bool)⇒i" ("I")
 where "IΦ ≡ if ∃x. E(x) ∧ Φ(x) ∧ (∀y. (E(y) ∧ Φ(y)) ⟶ (y = x))
              then THE x. E(x) ∧ Φ(x)
              else ⋆"
abbreviation fThatBinder:: "(i⇒bool)⇒i"  (binder "I" [8] 9)
 where "Ix. φ(x) ≡ I(φ)"
abbreviation fOr (infixr "∨" 51) where "φ∨ψ ≡ (¬φ)→ψ"
abbreviation fAnd (infixr "∧" 52) where "φ∧ψ ≡ ¬(¬φ∨¬ψ)"
abbreviation fEquiv (infixr "↔" 50) where "φ↔ψ ≡ (φ→ψ)∧(ψ→φ)"
```

```
abbreviation fThat:: "(i⇒bool)⇒i" ("I")
 where "IΦ ≡ if ∃x. E(x) ∧ Φ(x) ∧ (∀y. (E(y) ∧ Φ(y)) ⟶ (y = x))
              then THE x. E(x) ∧ Φ(x)
              else ⋆"
abbreviation fThatBinder:: "(i⇒bool)⇒i"  (binder "I" [8] 9)
 where "Ix. φ(x) ≡ I(φ)"
```

17,24 (511/4534) (isabelle,isabelle,UTF-8-Isabelle) UG 548/781MB 1:36 AM



Part C — Application:
Exploration of Axioms Systems for Category Theory

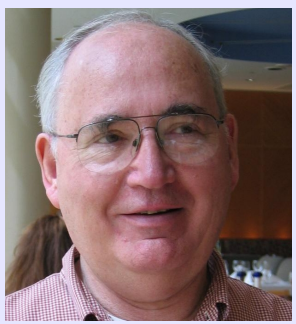
Exemplary Case Study: Exploration of Axioms Sets for Category Theory

Axioms Set I

———

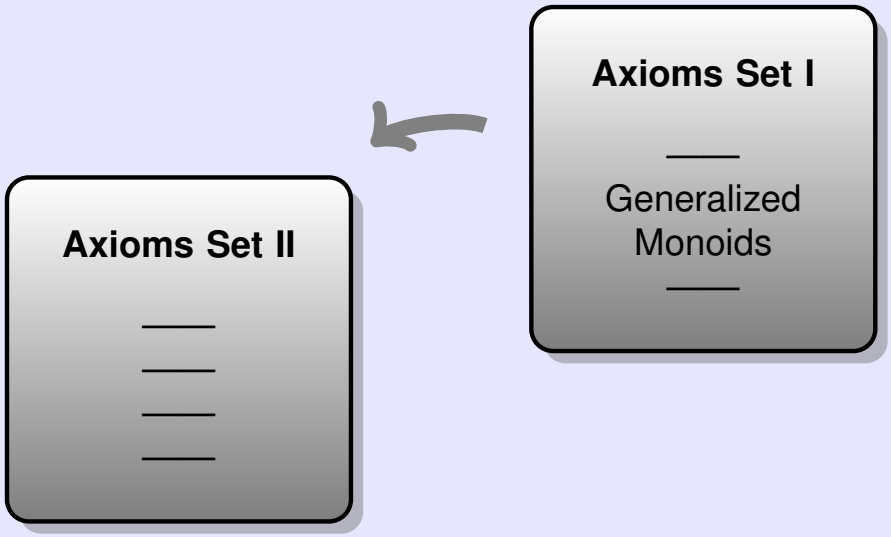
Generalized
Monoids

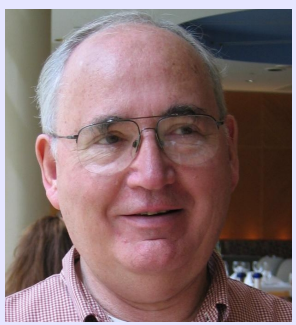
———



Dana Scott

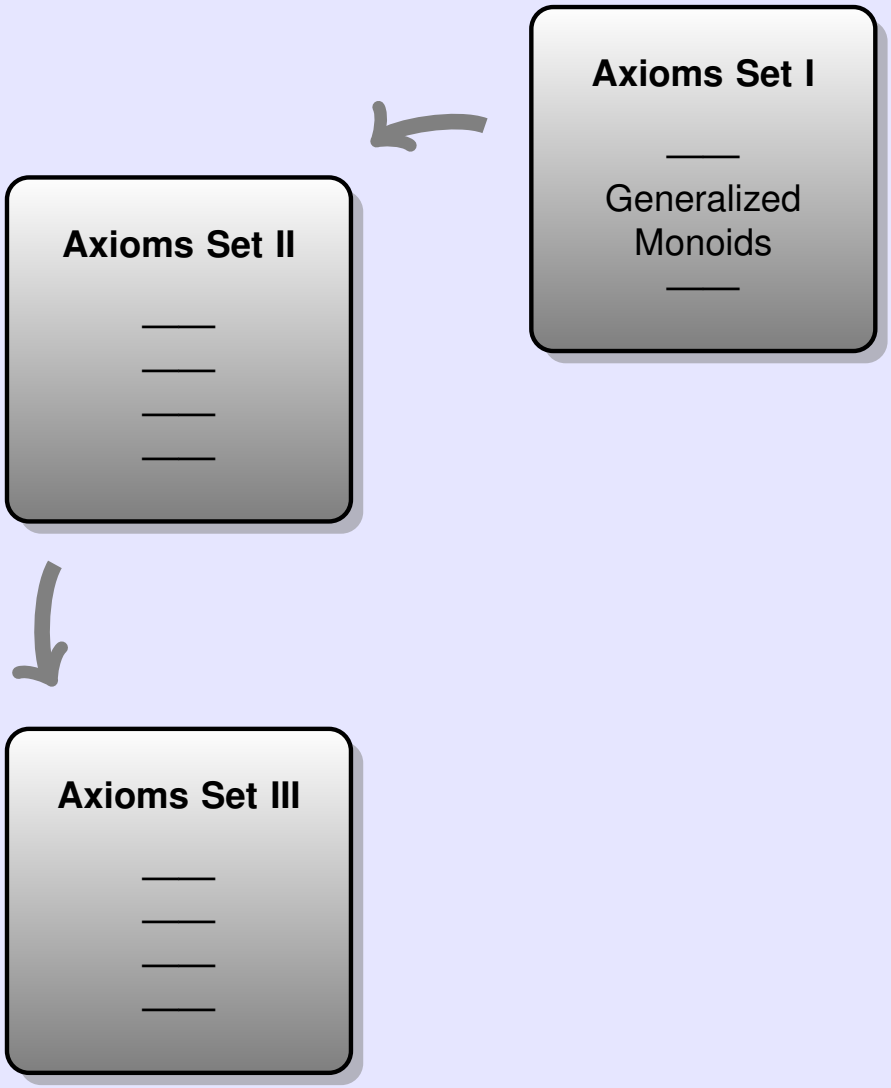
Exemplary Case Study: Exploration of Axioms Sets for Category Theory

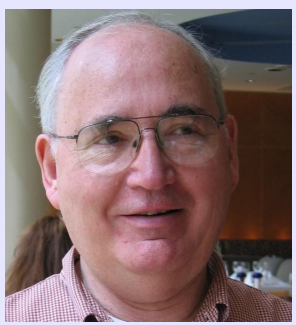




Dana Scott

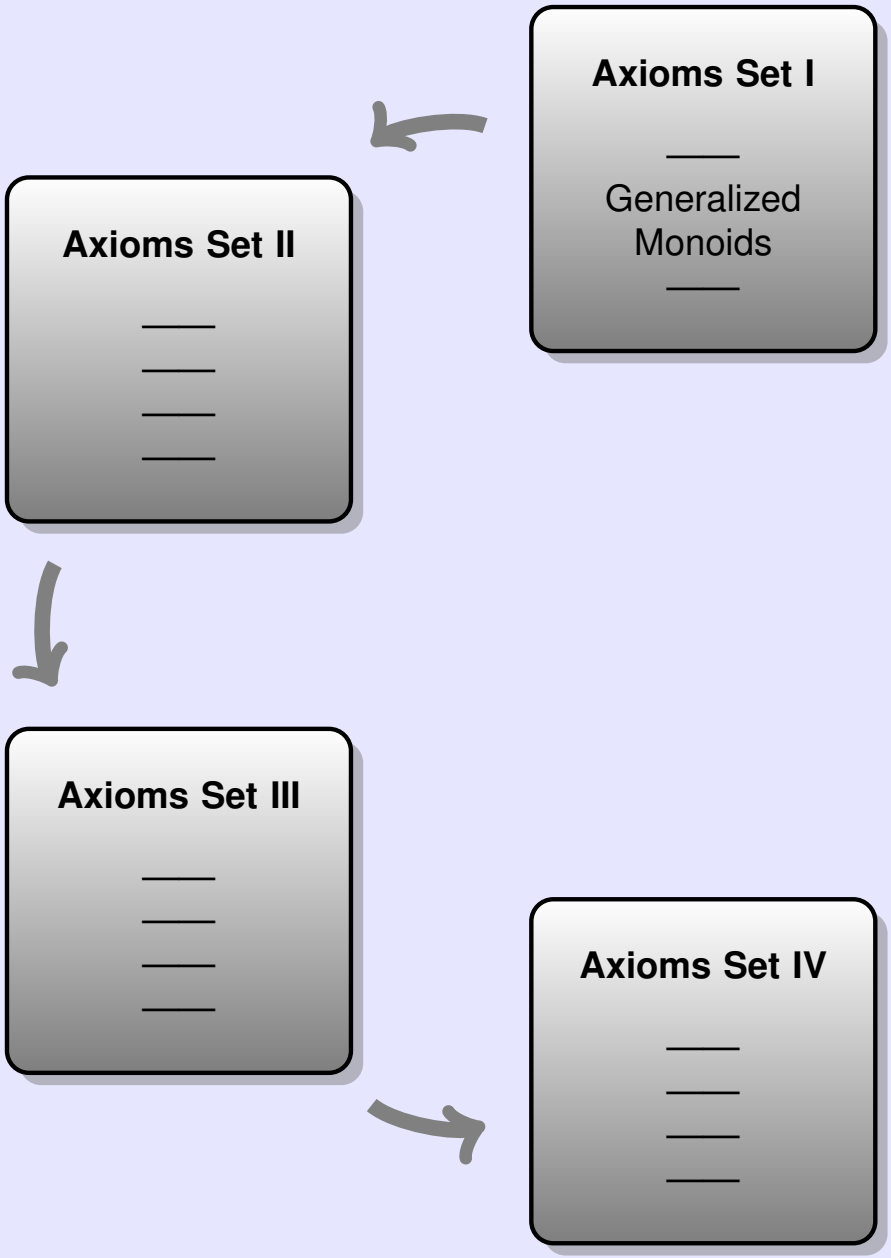
Exemplary Case Study: Exploration of Axioms Sets for Category Theory





Dana Scott

Exemplary Case Study: Exploration of Axioms Sets for Category Theory



Dana Scott

Exemplary Case Study: Exploration of Axioms Sets for Category Theory

Axioms Set I

Generalized Monoids

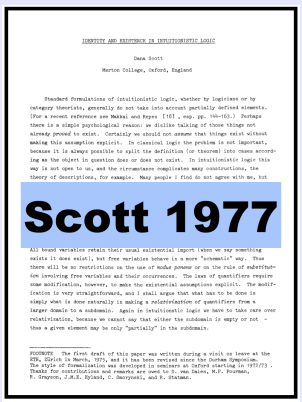
Axioms Set II

Axioms Set III

Axioms Set IV

Axioms Set V

Dana Scott's Axioms from 1977



IDENTITY AND REFERENCE IN INTUITIONISTIC LOGIC

Dana Scott

Merton College, Oxford, England

Standard formulations of intuitionistic logic, whether by logicians or by category theorists, generally do not take into account partially defined elements. (For a recent reference see Makkai and Reyes [1] , esp. pp. 116-163.) Perhaps there is a simple psychological reason: we dislike talking of those things not already proved to exist. Certainly we should not assume that things exist without making this assumption explicit. In classical logic the problem is not important, because it is always possible to split the definition (or theorem) into cases according as the object in question does or does not exist. In intuitionistic logic this way is not open to us, and the circumstance complicates many constructions, the theory of descriptions, for example. Many people I find do not agree with me, but

Scott 1977

All bound variables retain their usual existential import (when we say something exists it does exist), but free variables behave in a more "schematic" way. Thus there will be no restrictions on the use of modus ponens or on the rule of substitution involving free variables and their occurrences. The laws of quantifiers require some modification, however, to make the existential assumptions explicit. The modification is very straightforward, and I shall argue that what has to be done is simply what is done naturally in making a relativization of quantifiers from a larger domain to a subdomain. Again in intuitionistic logic we have to take care over relativizing, because we cannot say that either the subdomain is empty or not - thus a given element may be only "partially" in the subdomain.

FOOTNOTE The first draft of this paper was written during a visit on leave at the ETH, Zürich in March, 1973, and it has been revised since the Durham Symposium. The style of formalization was developed in seminars at Oxford starting in 1970/71 . Thanks for contributions and remarks are owed to D. van Dalen, M.P. Fourman, P. Grayson, J.M.E. Hyland, C. Smoryński, and R. Statman.

Dana Scott

Exemplary Case Study: Exploration of Axioms Sets for Category Theory

Axioms Set I

Generalized Monoids

Axioms Set II

Axioms Set III

Axioms Set IV

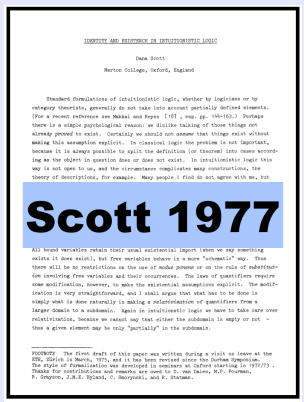
Axioms Set V

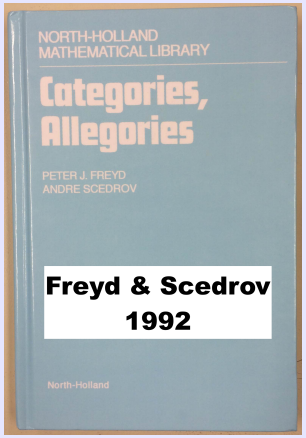
Dana Scott's Axioms from 1977

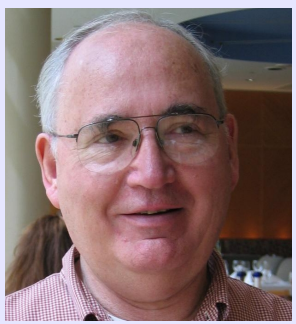
–?–

Axioms Set VI

Freyd & Scedrov's Axioms from 1992







Dana Scott

Exemplary Case Study: Exploration of Axioms Sets for Category Theory

Axioms Set I

Generalized Monoids

Axioms Set II

Axioms Set III

Axioms Set IV

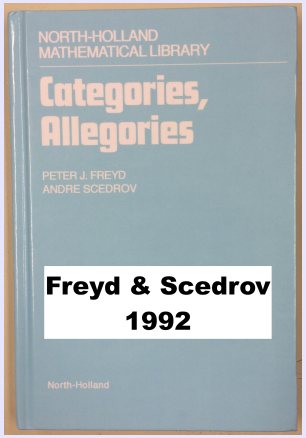
Axioms Set V

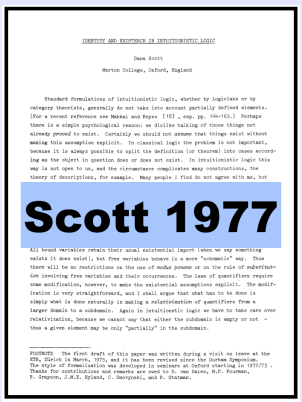
Dana Scott's Axioms from 1977

–?–

Axioms Set VI

Freyd & Scedrov's Axioms from 1992

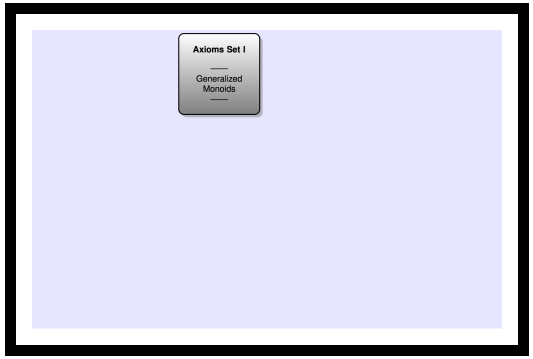




Dana Scott

all equivalent?

Preliminaries

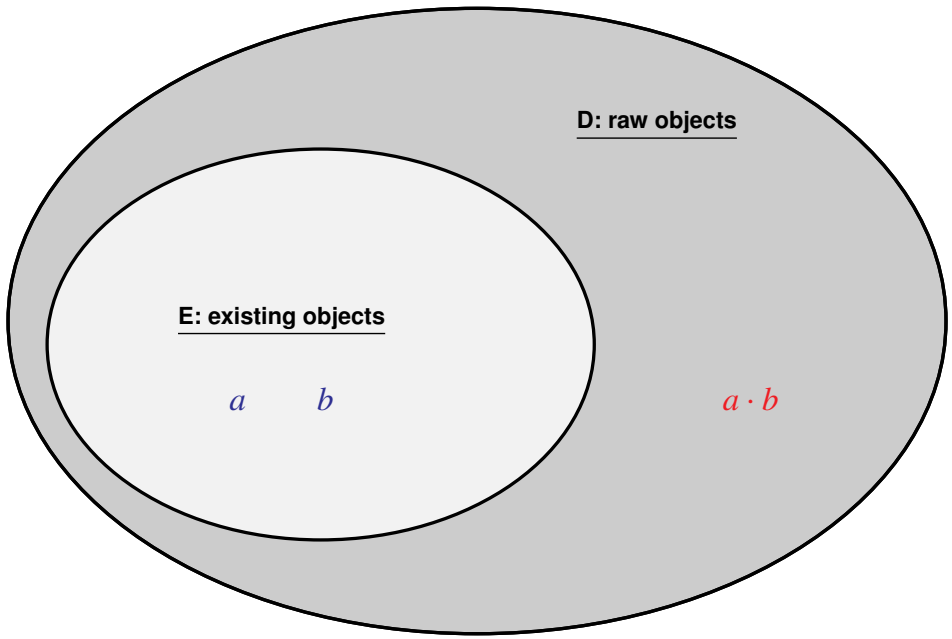


Morphisms: objects of type of i (raw domain D)

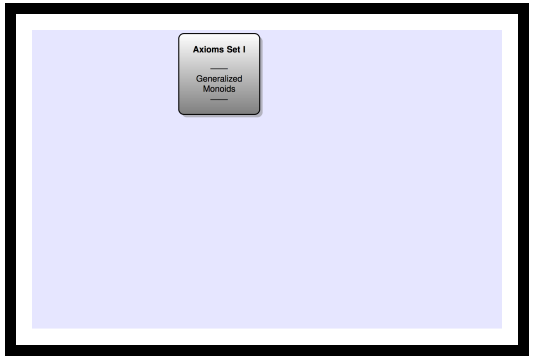
Partial functions:

domain	dom	of type $i \to i$
codomain	cod	of type $i \to i$
composition	$\cdot$	of type $i \to i \to i$ (resp. $i \times i \to i$)

Partiality of "$\cdot$" handled as expected:
$a \cdot b$ may be non-existing for some existing morphisms a and b.



Preliminaries

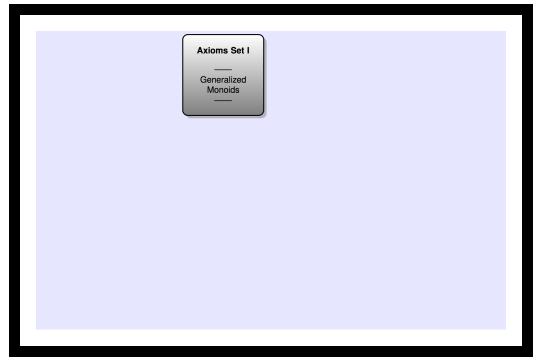


Morphisms: objects of type of i (raw domain D)

Partial functions:

domain	dom	of type $i \to i$
codomain	cod	of type $i \to i$
composition	$\cdot$	of type $i \to i \to i$ (resp. $i \times i \to i$)

Preliminaries



Morphisms: objects of type of i (raw domain D)

Partial functions:

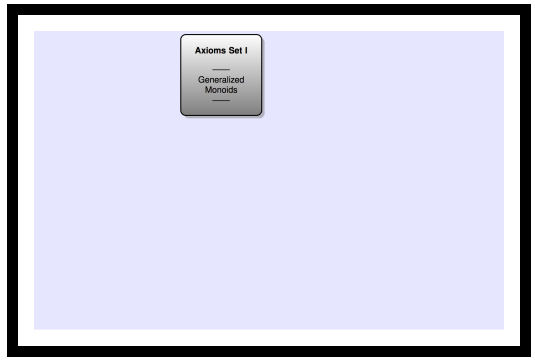
domain	dom	of type $i \to i$
codomain	cod	of type $i \to i$
composition	$\cdot$	of type $i \to i \to i$ (resp. $i \times i \to i$)

$\cong$ denotes Kleene equality: $x \cong y \equiv (Ex \vee Ey) \to x = y$

(where $=$ is identity on all objects of type i, existing or non-existing)

$\cong$ is an equivalence relation: **SLEDGEHAMMER**.

Preliminaries



Morphisms: objects of type of i (raw domain D)

Partial functions:

domain	dom	of type $i \to i$
codomain	cod	of type $i \to i$
composition	$\cdot$	of type $i \to i \to i$ (resp. $i \times i \to i$)

$\cong$ denotes Kleene equality: $x \cong y \equiv (Ex \vee Ey) \to x = y$

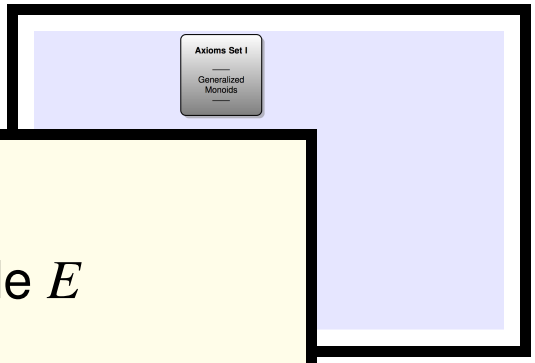
(where $=$ is identity on all objects of type i, existing or non-existing)

$\cong$ is an equivalence relation: **SLEDGEHAMMER**.

$\simeq$ denotes existing identity: $x \simeq y \equiv Ex \wedge Ey \wedge x = y$

$\simeq$ is symmetric and transitive, but lacks reflexivity: **SLEDGEHAMMER**, **NITPICK**.

Preliminaries



- $\simeq$ equivalence relation on E, empty relation outside E
- $1/0 \not\simeq 1/0 \quad 1/0 \not\simeq 2/0 \quad \ldots$
- $Ix.pkoFrance(x) \not\simeq Ix.pkoFrance(x)$
  $Ix.pkoFrance(x) \not\simeq Ix.pkoPoland(x)$

$\cong$ denotes Kleene equality: $x \cong y \equiv (Ex \vee Ey) \rightarrow x = y$

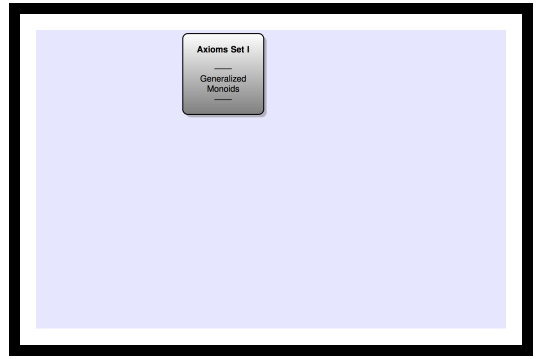
(where $=$ is identity on all objects of type i, existing or non-existing)

$\cong$ is an equivalence relation: **SLEDGEHAMMER**.

$\simeq$ denotes existing identity: $x \simeq y \equiv Ex \wedge Ey \wedge x = y$

$\simeq$ is symmetric and transitive, but lacks reflexivity: **SLEDGEHAMMER**, **NITPICK**.

From Monoids to Categories



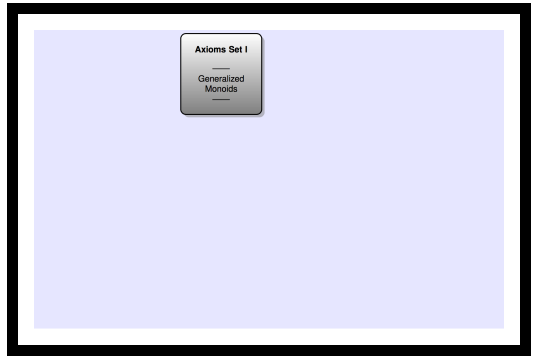
Monoid

A monoid is an algebraic structure $(S, \circ)$, where $\circ$ is a binary operator on set S, satisfying the following properties:

Closure:	$\forall a, b \in S.\ a \circ b \in S$
Associativity:	$\forall a, b, c \in S.\ a \circ (b \circ c) = (a \circ b) \circ c$
Identity:	$\exists id_S \in S.\ \forall a \in S.\ id_S \circ a = a = a \circ id_S$

That is, a monoid is a semigroup with a two-sided identity element.

From Monoids to Categories



We employ a partial, strict binary composition operation $\cdot$
Left and right identity elements are addressed in C_i, D_i, .

Categories: Axioms Set I

S_i	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey)$
E_i	Existence	$E(x \cdot y) \leftarrow (Ex \wedge Ey \wedge (\exists z.z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y))$
A_i	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_i	Codomain	$\forall y.\exists i.ID(i) \wedge i \cdot y \cong y$
D_i	Domain	$\forall x.\exists j.ID(j) \wedge x \cdot j \cong x$

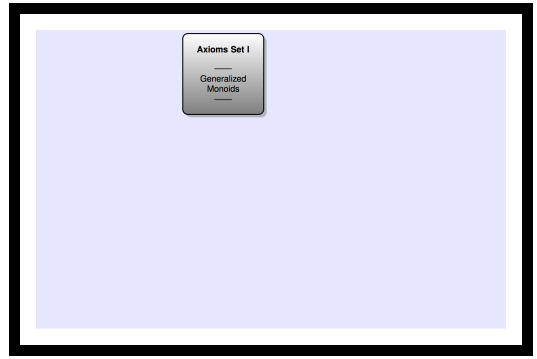
where I is an identity morphism predicate:

$$ID(i) \equiv (\forall x.\ E(i \cdot x) \rightarrow i \cdot x \cong x) \wedge (\forall x.\ E(x \cdot i) \rightarrow x \cdot i \cong x)$$

Monoid

Closure:	$\forall a, b \in S.\ a \circ b \in S$
Associativity:	$\forall a, b, c \in S.\ a \circ (b \circ c) = (a \circ b) \circ c$
Identity:	$\exists id_S \in S.\ \forall a \in S.\ id_S \circ a = a = a \circ id_S$

From Monoids to Categories



We employ a partial, strict binary composition operation $\cdot$
Left and right identity elements are addressed in C_i, D_i, .

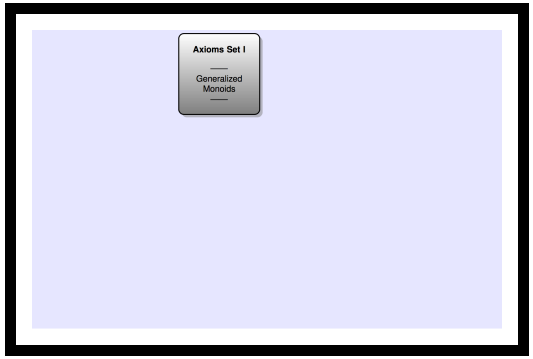
Categories: Axioms Set I

S_i	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey)$
E_i	Existence	$E(x \cdot y) \leftarrow (Ex \wedge Ey \wedge (\exists z.z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y))$
A_i	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_i	Codomain	$\forall y.\exists i.ID(i) \wedge i \cdot y \cong y$
D_i	Domain	$\forall x.\exists j.ID(j) \wedge x \cdot j \cong x$

where I is an identity morphism predicate:

$$ID(i) \equiv (\forall x.\ E(i \cdot x) \rightarrow i \cdot x \cong x) \wedge (\forall x.\ E(x \cdot i) \rightarrow x \cdot i \cong x)$$

From Monoids to Categories



We employ a partial, strict binary composition operation $\cdot$
Left and right identity elements are addressed in C_i, D_i, .

Categories: Axioms Set I

S_i	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey)$
E_i	Existence	$E(x \cdot y) \leftarrow (Ex \wedge Ey \wedge (\exists z.z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y))$
A_i	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_i	Codomain	$\forall y.\exists i.ID(i) \wedge i \cdot y \cong y$
D_i	Domain	$\forall x.\exists j.ID(j) \wedge x \cdot j \cong x$

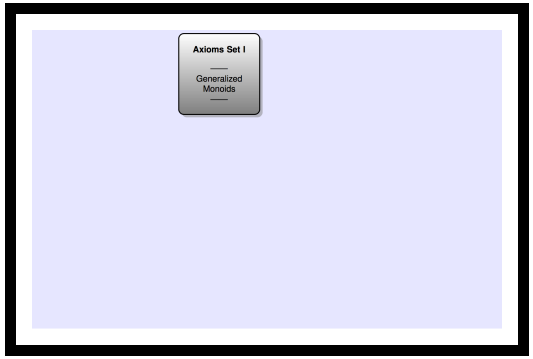
where I is an identity morphism predicate:

$$ID(i) \equiv (\forall x.\ E(i \cdot x) \rightarrow i \cdot x \cong x) \wedge (\forall x.\ E(x \cdot i) \rightarrow x \cdot i \cong x)$$

Experiments with Isabelle/HOL

- The i in axiom C is unique: **Sledgehammer**.
- The j in axiom D is unique: **Sledgehammer**.
- However, the i and j need not be equal: **Nitpick**

From Monoids to Categories



We employ a partial, strict binary composition operation $\cdot$
Left and right identity elements are addressed in C_i, D_i, .

Categories: Axioms Set I

S_i	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey)$
E_i	Existence	$E(x \cdot y) \leftarrow (Ex \wedge Ey \wedge (\exists z.z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y))$
A_i	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_i	Codomain	$\forall y.\exists i.ID(i) \wedge i \cdot y \cong y$
D_i	Domain	$\forall x.\exists j.ID(j) \wedge x \cdot j \cong x$

where I is an identity morphism predicate:

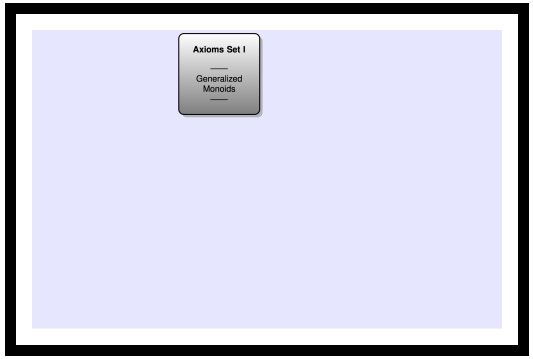
$$ID(i) \equiv (\forall x.\ E(i \cdot x) \rightarrow i \cdot x \cong x) \wedge (\forall x.\ E(x \cdot i) \rightarrow x \cdot i \cong x)$$

Experiments with Isabelle/HOL

- The left-to-right direction of E is implied: **SLEDGEHAMMER**.

$$E(x \cdot y) \rightarrow (Ex \wedge Ey \wedge (\exists z.z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y))$$

From Monoids to Categories



We employ a partial, strict binary composition operation $\cdot$
Left and right identity elements are addressed in C_i, D_i, .

Categories: Axioms Set I

S_i	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey)$
E_i	Existence	$E(x \cdot y) \leftarrow (Ex \wedge Ey \wedge (\exists z.z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y))$
A_i	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_i	Codomain	$\forall y.\exists i.ID(i) \wedge i \cdot y \cong y$
D_i	Domain	$\forall x.\exists j.ID(j) \wedge x \cdot j \cong x$

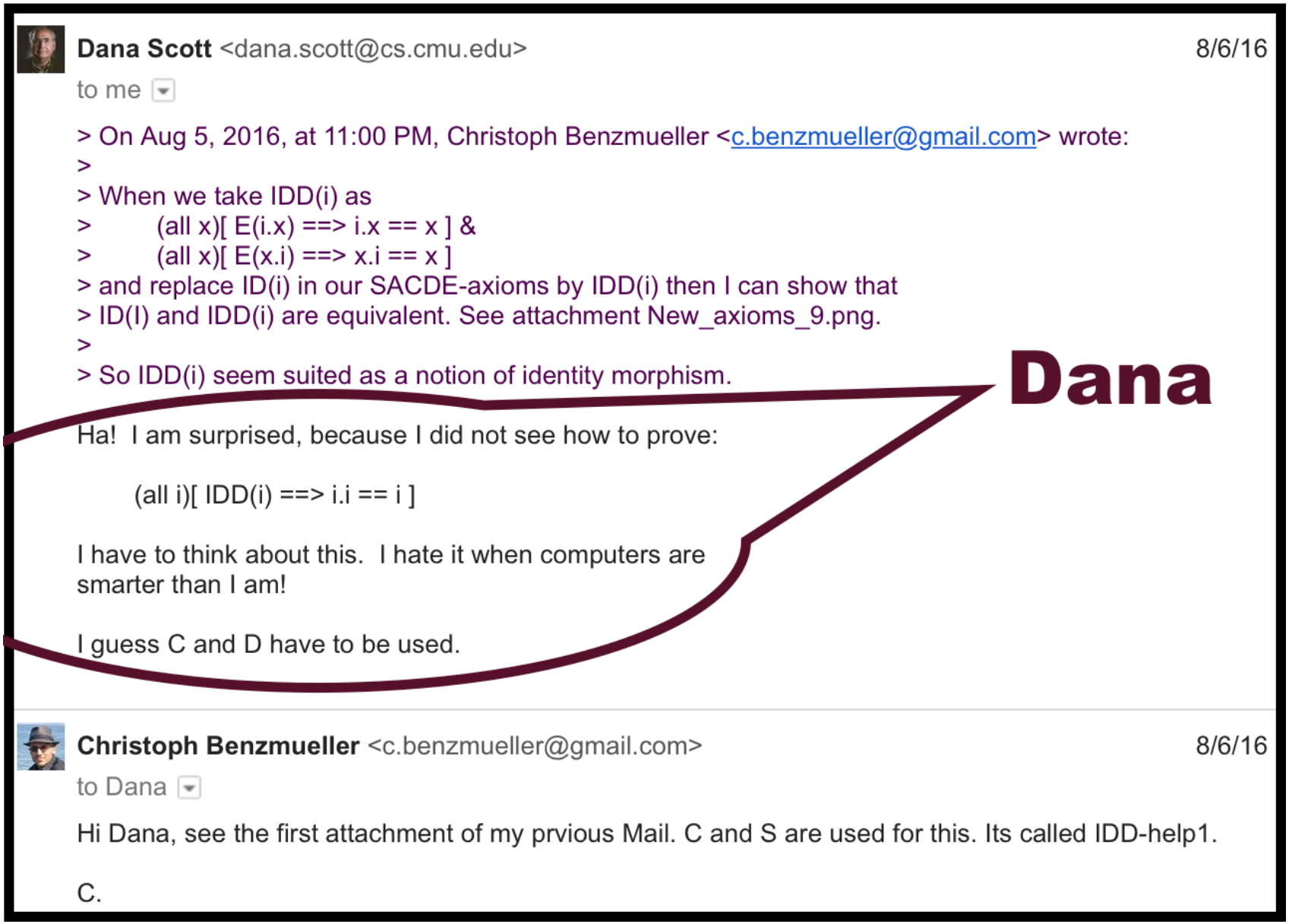
where I is an identity morphism predicate:

$$ID(i) \equiv (\forall x.\ E(i \cdot x) \rightarrow i \cdot x \cong x) \wedge (\forall x.\ E(x \cdot i) \rightarrow x \cdot i \cong x)$$

Experiments with Isabelle/HOL

- Model finder **Nitpick** confirms that this axiom set is consistent.
- Even if we assume there are non-existing objects ($\exists x.\neg(Ex)$) we get consistency.

Interaction: Dana – Christoph – Isabelle/HOL



Dana Scott <dana.scott@cs.cmu.edu> 8/6/16

to me

> On Aug 5, 2016, at 11:00 PM, Christoph Benzmueller <c.benzmueller@gmail.com> wrote:
>
> When we take IDD(i) as
> (all x)[E(i.x) ==> i.x == x] &
> (all x)[E(x.i) ==> x.i == x]
> and replace ID(i) in our SACDE-axioms by IDD(i) then I can show that
> ID(I) and IDD(i) are equivalent. See attachment New_axioms_9.png.
>
> So IDD(i) seem suited as a notion of identity morphism.

Dana

Ha! I am surprised, because I did not see how to prove:

(all i)[IDD(i) ==> i.i == i]

I have to think about this. I hate it when computers are smarter than I am!

I guess C and D have to be used.

Christoph Benzmueller <c.benzmueller@gmail.com> 8/6/16

to Dana

Hi Dana, see the first attachment of my prvious Mail. C and S are used for this. Its called IDD-help1.

C.

Interaction: Dana – Christoph – Isabelle/HOL

Christoph Benzmueller <c.benzmueller@gmail.com> 7/23/16

to Dana

Dana,

here are the results of the experiments; doesn't look too good.

On Fri, Jul 22, 2016 at 11:43 PM, Dana Scott <dana.scott@cs.cmu.edu> wrote:

> On Jul 21, 2016, at 9:32 AM, Christoph Benzmueller <c.benzmueller@gmail.com> wrote:
> >
> > The F-axioms are all provable from the old S-axioms.
> > But D2, D3 and E3 are not.
>
> I think I see the trouble with those D axioms. But E3 is very odd.
>
> E3: E(x.y) ==> (exist i)[Id(i) & x.(i.y) == x.y]
>
> You see, by the S-axioms, if you assume E(x.y), then E(x) & E(y) & E(cod(x))
> follows. So the "i" in the conclusion of E3 ought to be "cod(x)".
>
> Please check, therefore, whether this is provable from the S-axioms:
>
> (all x) Id(cod(x))

Apparently it isn't. See file Scott_new_axioms_4.png; the countermodel is presented in the lower window; he have:

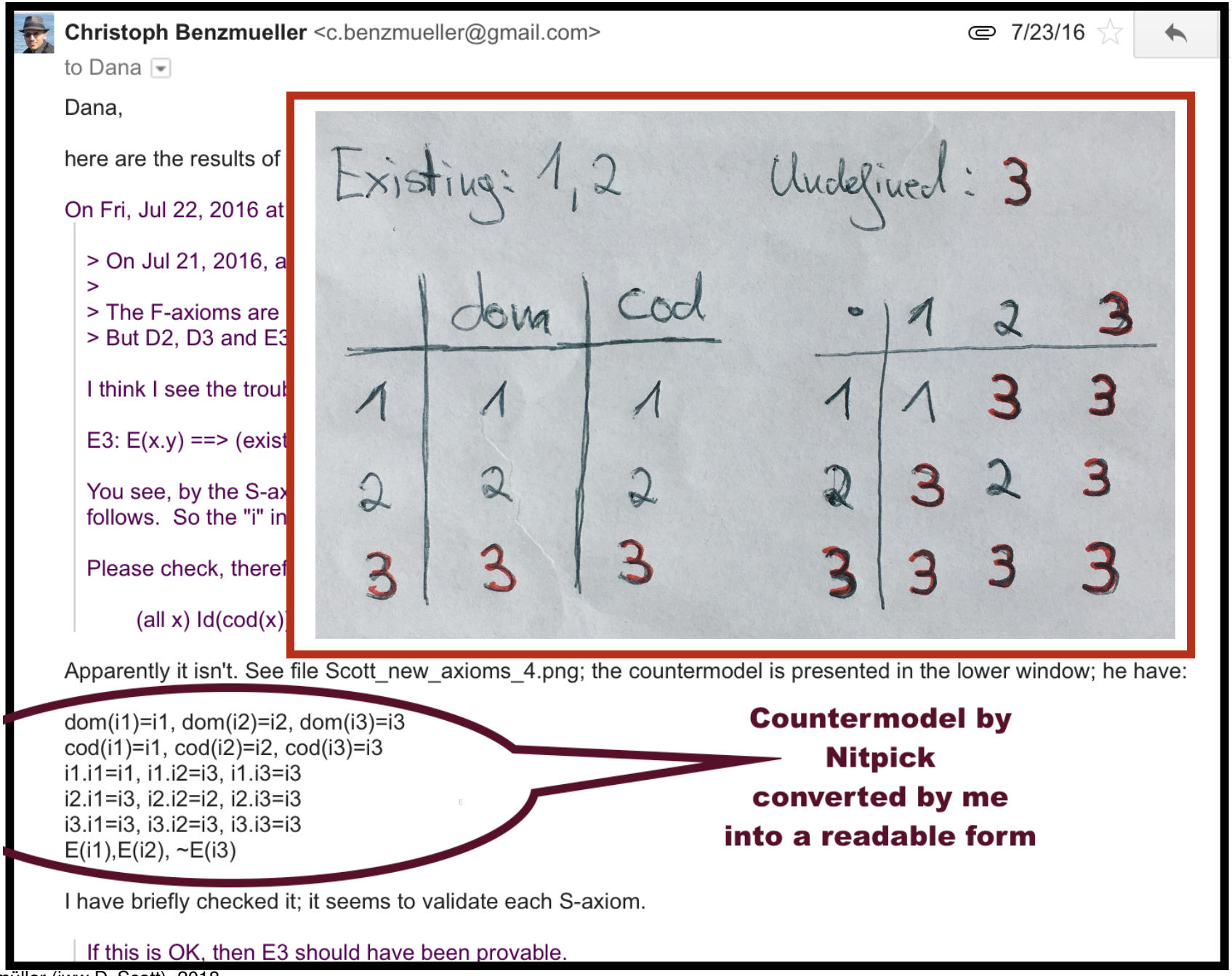
dom(i1)=i1, dom(i2)=i2, dom(i3)=i3
cod(i1)=i1, cod(i2)=i2, cod(i3)=i3
i1.i1=i1, i1.i2=i3, i1.i3=i3
i2.i1=i3, i2.i2=i2, i2.i3=i3
i3.i1=i3, i3.i2=i3, i3.i3=i3
E(i1),E(i2), ~E(i3)

Countermodel by Nitpick converted by me into a readable form

I have briefly checked it; it seems to validate each S-axiom.

> If this is OK, then E3 should have been provable.

Interaction: Dana – Christoph – Isabelle/HOL



Christoph Benzmueller <c.benzmueller@gmail.com> 7/23/16

to Dana

Dana,

here are the results of

On Fri, Jul 22, 2016 at

> On Jul 21, 2016, a
>
> The F-axioms are
> But D2, D3 and E3

I think I see the troub

E3: E(x.y) ==> (exist

You see, by the S-ax
follows. So the "i" in

Please check, theref

(all x) Id(cod(x)

Existing: 1, 2 Undefined: 3

	dom	cod
1	1	1
2	2	2
3	3	3

·	1	2	3
1	1	3	3
2	3	2	3
3	3	3	3

Apparently it isn't. See file Scott_new_axioms_4.png; the countermodel is presented in the lower window; he have:

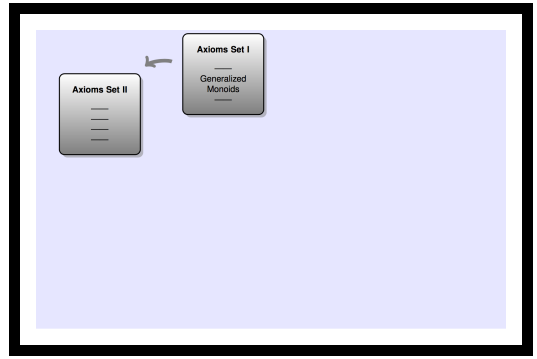
dom(i1)=i1, dom(i2)=i2, dom(i3)=i3
cod(i1)=i1, cod(i2)=i2, cod(i3)=i3
i1.i1=i1, i1.i2=i3, i1.i3=i3
i2.i1=i3, i2.i2=i2, i2.i3=i3
i3.i1=i3, i3.i2=i3, i3.i3=i3
E(i1),E(i2), ~E(i3)

Countermodel by Nitpick converted by me into a readable form

I have briefly checked it; it seems to validate each S-axiom.

If this is OK, then E3 should have been provable.

From Monoids to Categories



Axioms Set II is developed from Axioms Set I by Skolemization of i and j in axioms C and D. We can argue semantically that every model of Axioms Set I has such functions. The strictness axiom S is extended, so that strictness is now also postulated for the new Skolem functions dom and cod.

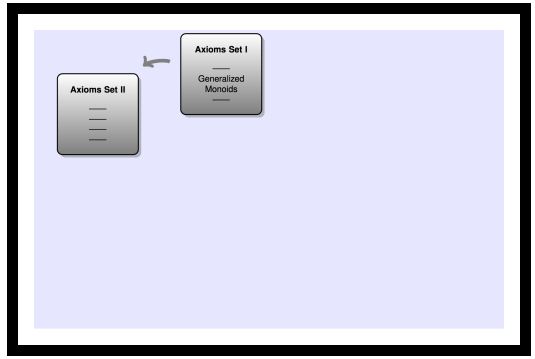
Categories: Axioms Set II

S_{ii}	Strictness	$E(x \cdot y) \to (Ex \wedge Ey) \wedge (E(dom\ x) \to Ex) \wedge (E(cod\ y) \to Ey)$
E_{ii}	Existence	$E(x \cdot y) \leftarrow (Ex \wedge Ey \wedge (\exists z.z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y))$
A_{ii}	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_{ii}	Codomain	$Ey \to (ID(cod\ y) \wedge (cod\ y) \cdot y \cong y)$
D_{ii}	Domain	$Ex \to (ID(dom\ x) \wedge x \cdot (dom\ x) \cong x)$

Categories: Axioms Set I

S_i	Strictness	$E(x \cdot y) \to (Ex \wedge Ey)$
E_i	Existence	$E(x \cdot y) \leftarrow (Ex \wedge Ey \wedge (\exists z.z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y))$
A_i	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_i	Codomain	$\forall y.\exists i.ID(i) \wedge i \cdot y \cong y$
D_i	Domain	$\forall x.\exists j.ID(j) \wedge x \cdot j \cong x$

From Monoids to Categories



Axioms Set II is developed from Axioms Set I by Skolemization of i and j in axioms C and D. We can argue semantically that every model of Axioms Set I has such functions. The strictness axiom S is extended, so that strictness is now also postulated for the new Skolem functions dom and cod.

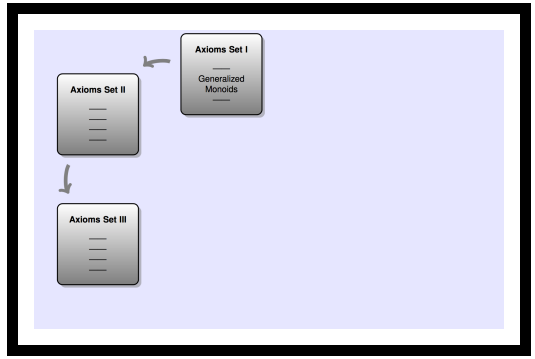
Categories: Axioms Set II

S_{ii}	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey) \wedge (E(dom\ x) \rightarrow Ex) \wedge (E(cod\ y) \rightarrow Ey)$
E_{ii}	Existence	$E(x \cdot y) \leftarrow (Ex \wedge Ey \wedge (\exists z.z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y))$
A_{ii}	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_{ii}	Codomain	$Ey \rightarrow (ID(cod\ y) \wedge (cod\ y) \cdot y \cong y)$
D_{ii}	Domain	$Ex \rightarrow (ID(dom\ x) \wedge x \cdot (dom\ x) \cong x)$

Experiments with Isabelle/HOL

- Consistency holds (also when $\exists x.\neg(Ex)$): confirmed by **Nitpick**.
- Axioms Set II implies Axioms Set I: easily proved by **Sledgehammer**.
- Axioms Set I also implies Axioms Set II (by semantical means on the meta-level)

From Monoids to Categories



In Axioms Set III the existence axiom E is simplified by taking advantage of the two new Skolem functions dom and cod.

Categories: Axioms Set III

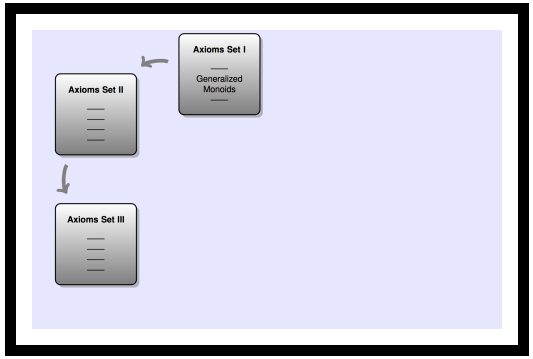
S_{iii}	Strictness	$E(x \cdot y) \to (Ex \wedge Ey) \wedge (E(dom\ x) \to Ex) \wedge (E(cod\ y) \to Ey)$
E_{iii}	Existence	$E(x \cdot y) \leftarrow (dom\ x \cong cod\ y \wedge E(cod\ y))$
A_{iii}	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_{iii}	Codomain	$Ey \to (ID(cod\ y) \wedge (cod\ y) \cdot y \cong y)$
D_{iii}	Domain	$Ex \to (ID(dom\ x) \wedge x \cdot (dom\ x) \cong x)$

Categories: Axioms Set II

S_{ii}	Strictness	$E(x \cdot y) \to (Ex \wedge Ey) \wedge (E(dom\ x) \to Ex) \wedge (E(cod\ y) \to Ey)$
E_{ii}	Existence	$E(x \cdot y) \leftarrow (Ex \wedge Ey \wedge (\exists z.z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y))$
A_{ii}	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_{ii}	Codomain	$Ey \to (ID(cod\ y) \wedge (cod\ y) \cdot y \cong y)$
D_{ii}	Domain	$Ex \to (ID(dom\ x) \wedge x \cdot (dom\ x) \cong x)$

From Monoids to Categories

In Axioms Set III the existence axiom E is simplified by taking advantage of the two new Skolem functions dom and cod.



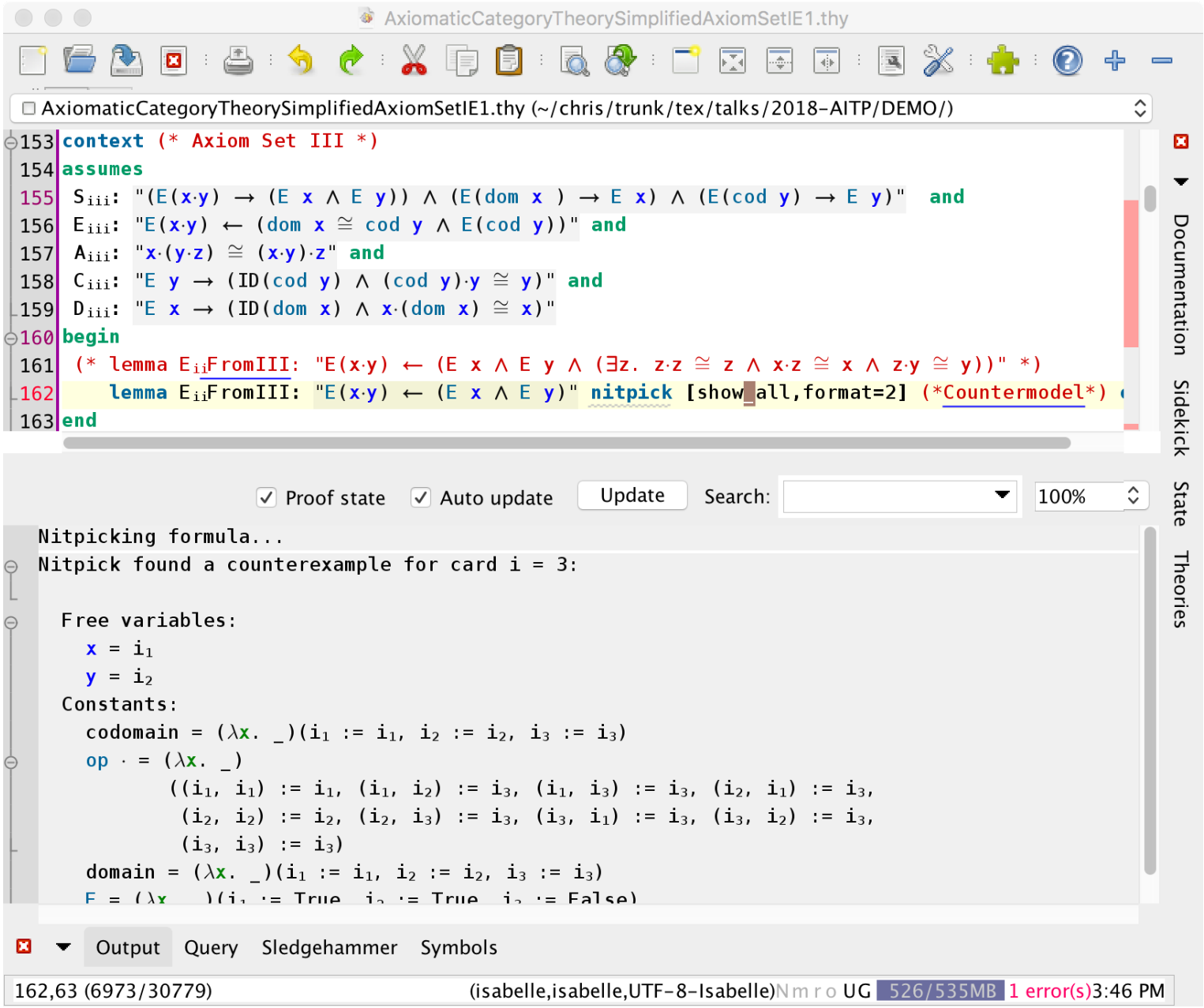
Categories: Axioms Set III

S_{iii}	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey) \wedge (E(dom\ x) \rightarrow Ex) \wedge (E(cod\ y) \rightarrow Ey)$
E_{iii}	Existence	$E(x \cdot y) \leftarrow (dom\ x \cong cod\ y \wedge E(cod\ y))$
A_{iii}	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_{iii}	Codomain	$Ey \rightarrow (ID(cod\ y) \wedge (cod\ y) \cdot y \cong y)$
D_{iii}	Domain	$Ex \rightarrow (ID(dom\ x) \wedge x \cdot (dom\ x) \cong x)$

Experiments with Isabelle/HOL

- Consistency holds (also when $\exists x.\neg(Ex)$): confirmed by **Nitpick**.
- The left-to-right direction of existence axiom E is implied: **Sledgehammer**.
- Axioms Set III implies Axioms Set II: **Sledgehammer**.
- Axioms Set II implies Axioms Set III: **Sledgehammer**.

Interesting Model (idempotents, but no left- & right-identities)



Interesting Model (idempotents, but no left- & right-identities)

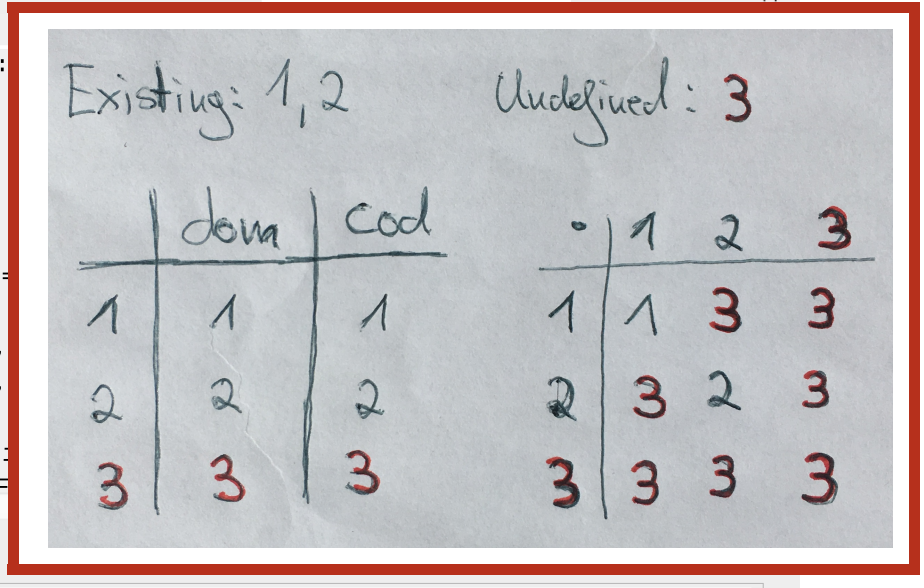
AxiomaticCategoryTheorySimplifiedAxiomSetIE1.thy

AxiomaticCategoryTheorySimplifiedAxiomSetIE1.thy (~/chris/trunk/tex/talks/2018-AITP/DEMO/)

```
153 context (* Axiom Set III *)
154 assumes
155  S_iii: "(E(x·y) → (E x ∧ E y)) ∧ (E(dom x ) → E x) ∧ (E(cod y) → E y)"  and
156  E_iii: "E(x·y) ← (dom x ≅ cod y ∧ E(cod y))" and
157  A_iii: "x·(y·z) ≅ (x·y)·z" and
158  C_iii: "E y → (ID(cod y) ∧ (cod y)·y ≅ y)" and
159  D_iii: "E x → (ID(dom x) ∧ x·(dom x) ≅ x)"
160 begin
161  (* lemma E_iiFromIII: "E(x·y) ← (E x ∧ E y ∧ (∃z. z·z ≅ z ∧ x·z ≅ x ∧ z·y ≅ y))" *)
162     lemma E_iiFromIII: "E(x·y) ← (E x ∧ E y)" nitpick [show_all,format=2] (*Countermodel*)
163 end
```

Proof state | Auto update | Update | Search: | 100%

```
Nitpicking formula...
Nitpick found a counterexample for card i = 3:

  Free variables:
    x = i₁
    y = i₂
  Constants:
    codomain = (λx. _)(i₁ := i₁, i₂ := i₂, i₃ :=
    op · = (λx. _)
           ((i₁, i₁) := i₁, (i₁, i₂) := i₃, (i₁,
            (i₂, i₂) := i₂, (i₂, i₃) := i₃, (i₃,
            (i₃, i₃) := i₃)
    domain = (λx. _)(i₁ := i₁, i₂ := i₂, i₃ :=
    E = (λx. _)(i₁ := True, i₂ := True, i₃ := F
```

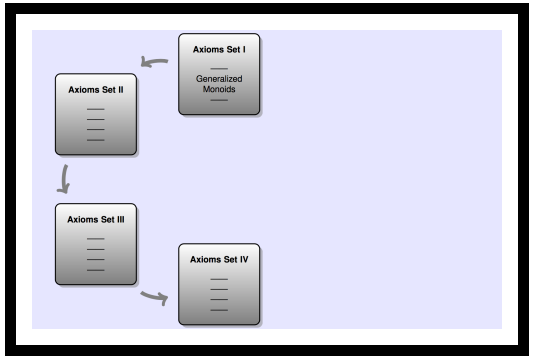


Output | Query | Sledgehammer | Symbols

162,63 (6973/30779) (isabelle,isabelle,UTF-8-Isabelle)Nmro UG 526/535MB 1 error(s)3:46 PM

From Monoids to Categories

Axioms Set IV simplifies the axioms C and D. However, as it turned out, these simplifications also require the existence axiom E to be strengthened into an equivalence.



Categories: Axioms Set IV

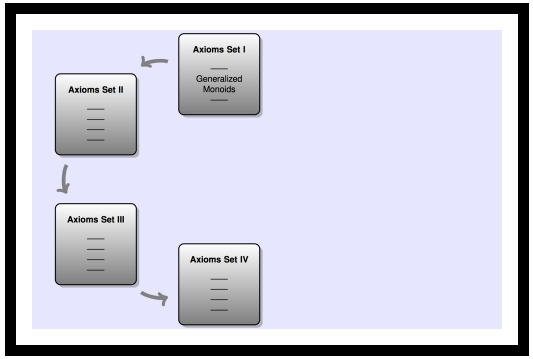
S_{iv}	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey) \wedge (E(dom\ x) \rightarrow Ex) \wedge (E(cod\ y) \rightarrow Ey)$
E_{iv}	Existence	$E(x \cdot y) \leftrightarrow (dom\ x \cong cod\ y \wedge E(cod\ y))$
A_{iv}	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_{iv}	Codomain	$(cod\ y) \cdot y \cong y$
D_{iv}	Domain	$x \cdot (dom\ x) \cong x$

Categories: Axioms Set III

S_{iii}	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey) \wedge (E(dom\ x) \rightarrow Ex) \wedge (E(cod\ y) \rightarrow Ey)$
E_{iii}	Existence	$E(x \cdot y) \leftarrow (dom\ x \cong cod\ y \wedge E(cod\ y))$
A_{iii}	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_{iii}	Codomain	$Ey \rightarrow (ID(cod\ y) \wedge (cod\ y) \cdot y \cong y)$
D_{iii}	Domain	$Ex \rightarrow (ID(dom\ x) \wedge x \cdot (dom\ x) \cong x)$

From Monoids to Categories

Axioms Set IV simplifies the axioms C and D. However, as it turned out, these simplifications also require the existence axiom E to be strengthened into an equivalence.



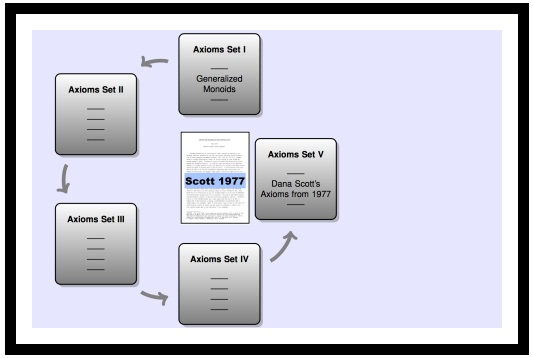
Categories: Axioms Set IV

S_{iv}	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey) \wedge (E(dom\ x) \rightarrow Ex) \wedge (E(cod\ y) \rightarrow Ey)$
E_{iv}	Existence	$E(x \cdot y) \leftrightarrow (dom\ x \cong cod\ y \wedge E(cod\ y))$
A_{iv}	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_{iv}	Codomain	$(cod\ y) \cdot y \cong y$
D_{iv}	Domain	$x \cdot (dom\ x) \cong x$

Experiments with Isabelle/HOL

- Consistency holds (also when $\exists x.\neg(Ex)$): confirmed by **Nitpick**.
- Axioms Set IV implies Axioms Set III: **Sledgehammer**.
- Axioms Set III implies Axioms Set IV: **Sledgehammer**.

From Monoids to Categories



Axioms Set V simplifies axiom E (and S).
Now, strictness of $\cdot$ is implied.

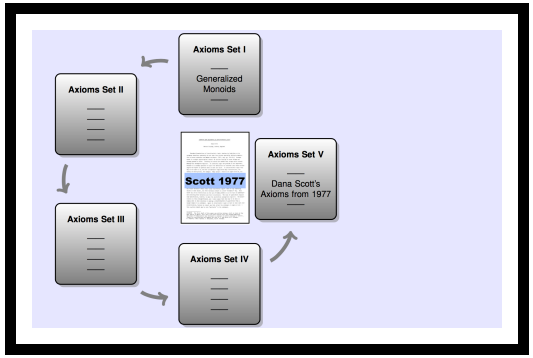
Categories: Axioms Set V (Scott, 1977)

$S1$	Strictness	$E(dom\ x) \rightarrow Ex$
$S2$	Strictness	$E(cod\ y) \rightarrow Ey$
$S3$	Existence	$E(x \cdot y) \leftrightarrow dom\ x \simeq cod\ y$
$S4$	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
$S5$	Codomain	$(cod\ y) \cdot y \cong y$
$S6$	Domain	$x \cdot (dom\ x) \cong x$

Categories: Axioms Set IV

S_{iv}	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey) \wedge (E(dom\ x) \rightarrow Ex) \wedge (E(cod\ y) \rightarrow Ey)$
E_{iv}	Existence	$E(x \cdot y) \leftrightarrow (dom\ x \cong cod\ y \wedge E(cod\ y))$
A_{iv}	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_{iv}	Codomain	$(cod\ y) \cdot y \cong y$
D_{iv}	Domain	$x \cdot (dom\ x) \cong x$

From Monoids to Categories



Axioms Set V simplifies axiom E (and S).
Now, strictness of $\cdot$ is implied.

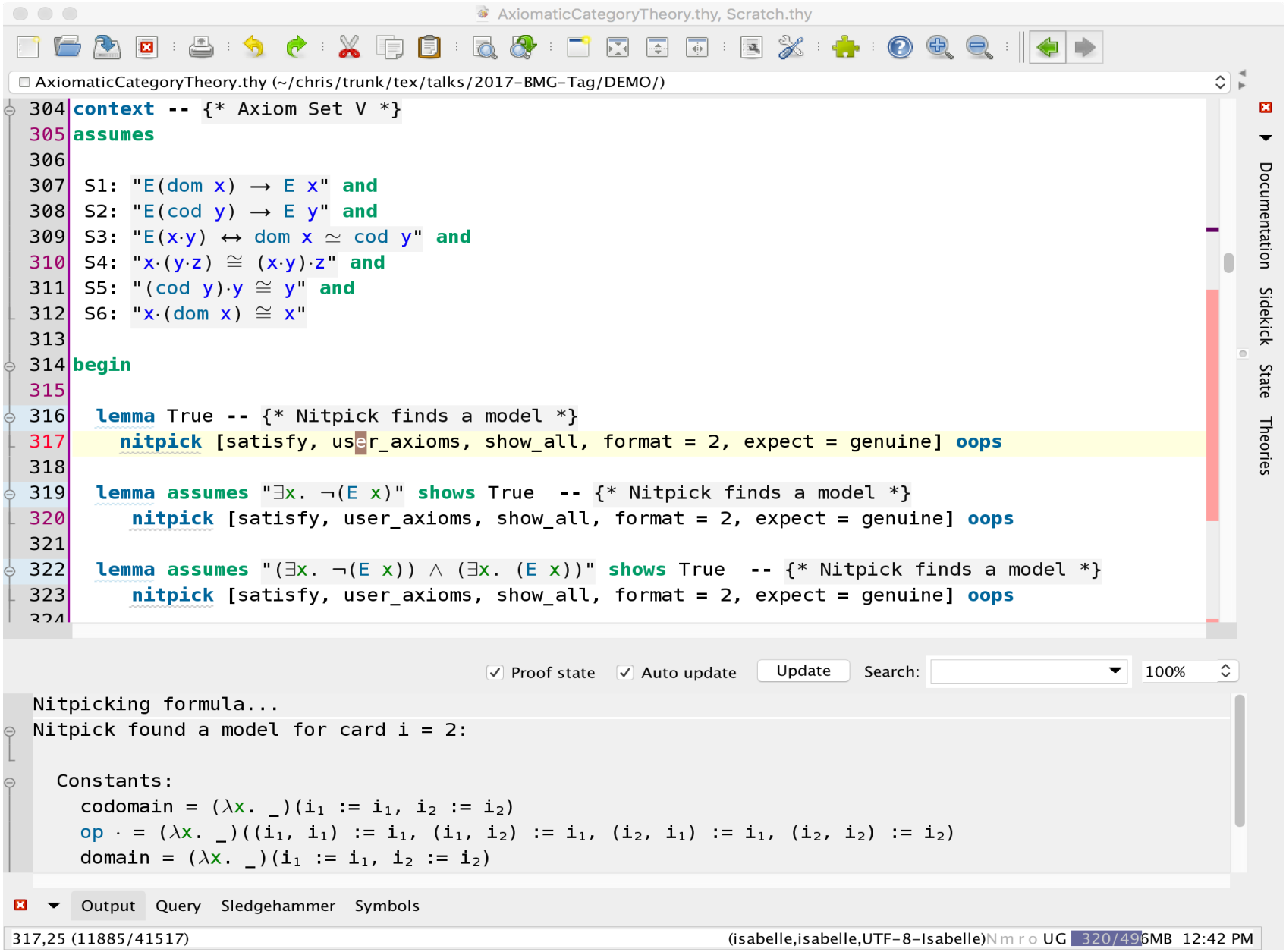
Categories: Axioms Set V (Scott, 1977)

$S1$	Strictness	$E(dom\ x) \rightarrow Ex$
$S2$	Strictness	$E(cod\ y) \rightarrow Ey$
$S3$	Existence	$E(x \cdot y) \leftrightarrow dom\ x \simeq cod\ y$
$S4$	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
$S5$	Codomain	$(cod\ y) \cdot y \cong y$
$S6$	Domain	$x \cdot (dom\ x) \cong x$

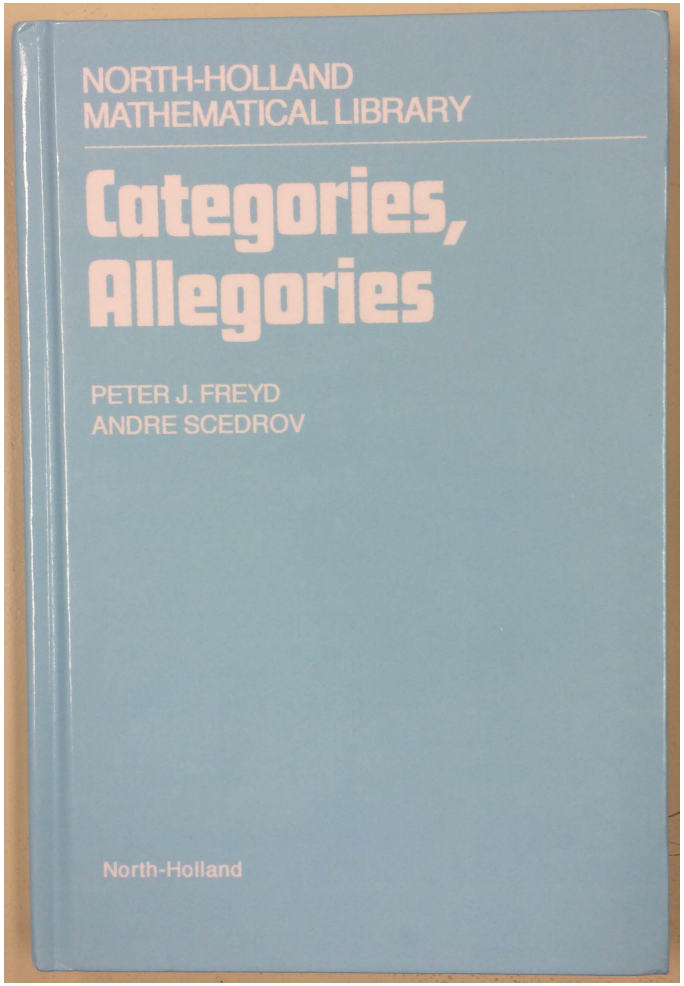
Experiments with Isabelle/HOL

- Consistency holds (also when $\exists x.\neg(Ex)$): confirmed by **NITPICK**.
- Axioms Set V implies Axioms Set IV: **SLEDGEHAMMER**.
- Axioms Set IV implies Axioms Set V: **SLEDGEHAMMER**.

Demo


AxiomaticCategoryTheory.thy, Scratch.thy
AxiomaticCategoryTheory.thy (~/chris/trunk/tex/talks/2017-BMG-Tag/DEMO/)
304 context -- {* Axiom Set V *}
305 assumes
306
307 S1: "E(dom x) ⟶ E x" and
308 S2: "E(cod y) ⟶ E y" and
309 S3: "E(x·y) ⟷ dom x ≃ cod y" and
310 S4: "x·(y·z) ≅ (x·y)·z" and
311 S5: "(cod y)·y ≅ y" and
312 S6: "x·(dom x) ≅ x"
313
314 begin
315
316 lemma True -- {* Nitpick finds a model *}
317 nitpick [satisfy, user_axioms, show_all, format = 2, expect = genuine] oops
318
319 lemma assumes "∃x. ¬(E x)" shows True -- {* Nitpick finds a model *}
320 nitpick [satisfy, user_axioms, show_all, format = 2, expect = genuine] oops
321
322 lemma assumes "(∃x. ¬(E x)) ∧ (∃x. (E x))" shows True -- {* Nitpick finds a model *}
323 nitpick [satisfy, user_axioms, show_all, format = 2, expect = genuine] oops
Documentation
Sidekick
State
Theories
Proof state
Auto update
Update
Search:
100%
Nitpicking formula...
Nitpick found a model for card i = 2:
Constants:
codomain = (λx. _)(i₁ := i₁, i₂ := i₂)
op · = (λx. _)((i₁, i₁) := i₁, (i₁, i₂) := i₁, (i₂, i₁) := i₁, (i₂, i₂) := i₂)
domain = (λx. _)(i₁ := i₁, i₂ := i₂)
Output
Query
Sledgehammer
Symbols
317,25 (11885/41517)
(isabelle,isabelle,UTF-8-Isabelle)
UG
320/496MB
12:42 PM

Cats & Alligators



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:

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

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

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

The axioms:

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

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

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

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

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

1.11. The ordinary equality sign = will be used only in the symmetric sense, to wit: if either side is defined then so is the other and they are equal. A theory, such as this, built on an ordered list of partial operations, the domain of definition of each given by equations in the previous, and with all other axioms equational, is called an ESSENTIALLY ALGEBRAIC THEORY.

1.12. We shall use a venturi-tube $\succcurlyeq$ for *directed equality* which means: if the left side is defined then so is the right and they are equal. The axiom that $\square(xy) = \square(x(\square y))$ is equivalent, in the presence of the earlier axioms, with $\square(xy) \succcurlyeq \square x$ as can be seen below.

1.13. $\square(\square x) = \square x$ because $\square(\square x) = \square((\square x)\square) = (\square x)\square = \square x$. Similarly $(x\square)\square = x\square$.

Cats & Alligators

Axioms Set I

Generalized Monoids

Axioms Set II

Axioms Set III

Axioms Set IV

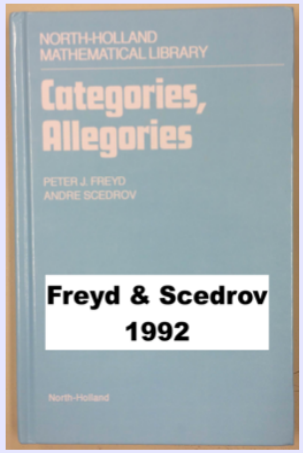
Axioms Set V

Dana Scott's Axioms from 1977

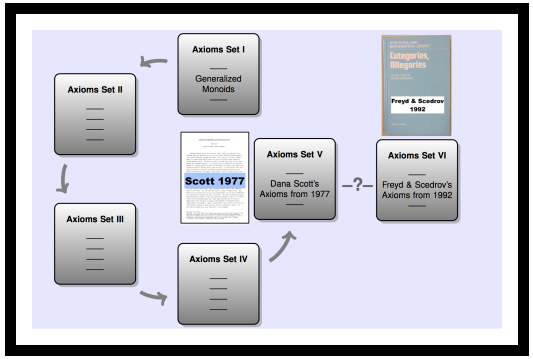
–?–

Axioms Set VI

Freyd & Scedrov's Axioms from 1992



Cats & Alligators



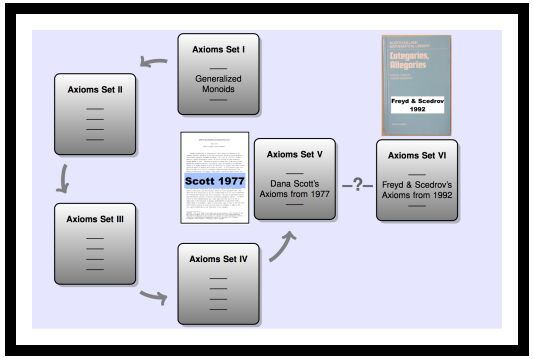
Categories: Original axiom set by Freyd and Scedrov (modulo notation)

A1	$E(x \cdot y) \leftrightarrow dom\ x \cong cod\ y$
A2a	$cod(dom\ x) \cong dom\ x$
A2b	$dom(cod\ y) \cong cod\ y$
A3a	$x \cdot (dom\ x) \cong x$
A3b	$(cod\ y) \cdot y \cong y$
A4a	$dom(x \cdot y) \cong dom((dom\ x) \cdot y)$
A4b	$cod(x \cdot y) \cong cod(x \cdot (cod\ y))$
A5	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$

Experiments with Isabelle/HOL

- Consistency? — Nitpick finds a model.
- Consistency when assuming $\exists x.\neg Ex$ — Nitpick does not find a model.
- lemma $(\exists x.\neg Ex) \rightarrow False$: **SLEDGEHAMMER**. (Problematic axioms: $A1, A2a, A3a$)

Cats & Alligators



Categories: Original axiom set by Freyd and Scedrov (modulo notation)

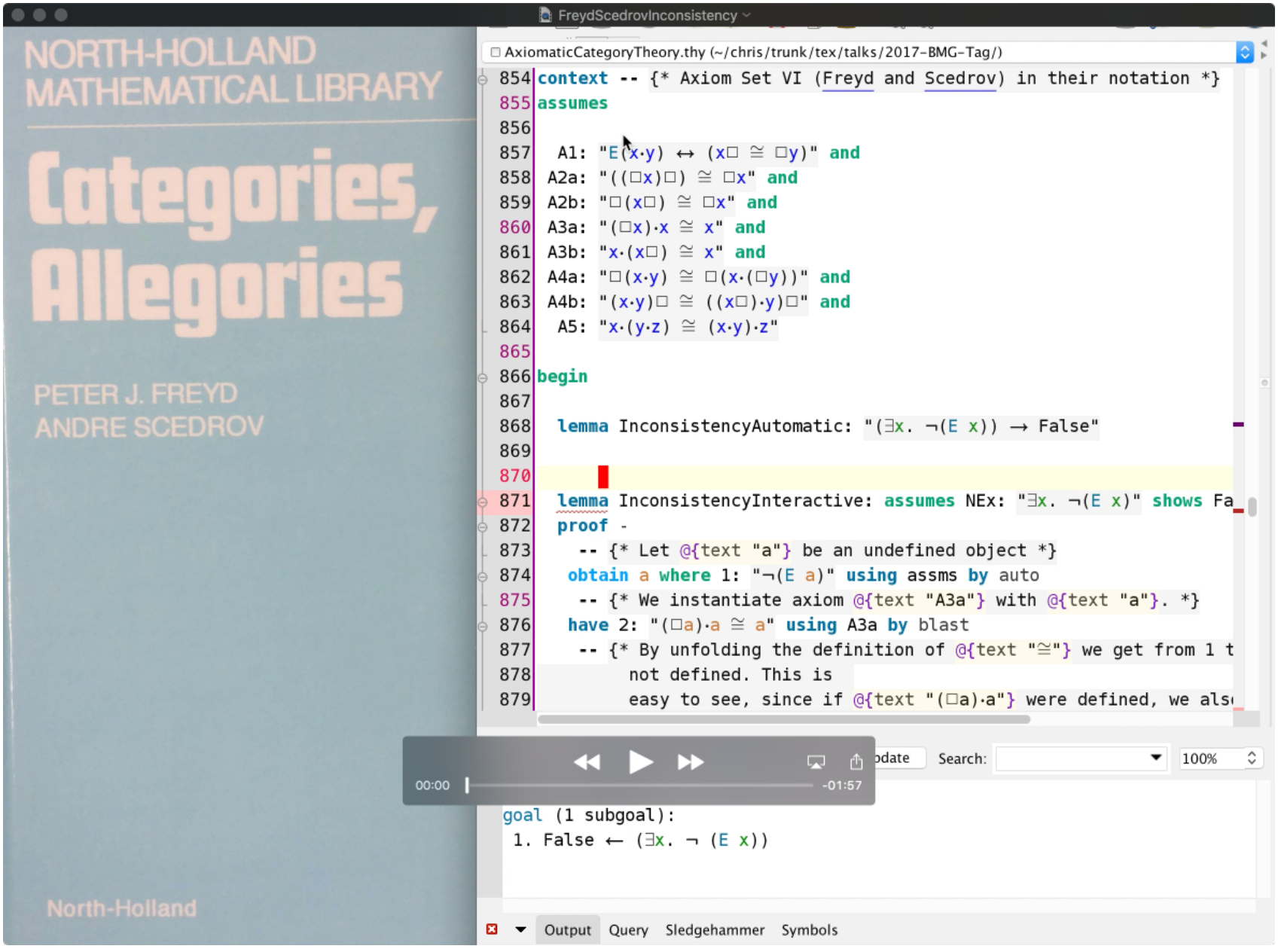
A1	$E(x \cdot y) \leftrightarrow dom\ x \cong cod\ y$
A2a	$cod(dom\ x) \cong dom\ x$
A2b	$dom(cod\ y) \cong cod\ y$
A3a	$x \cdot (dom\ x) \cong x$
A3b	$(cod\ y) \cdot y \cong y$
A4a	$dom(x \cdot y) \cong dom((dom\ x) \cdot y)$
A4b	$cod(x \cdot y) \cong cod(x \cdot (cod\ y))$
A5	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$

Experiments with Isabelle/HOL

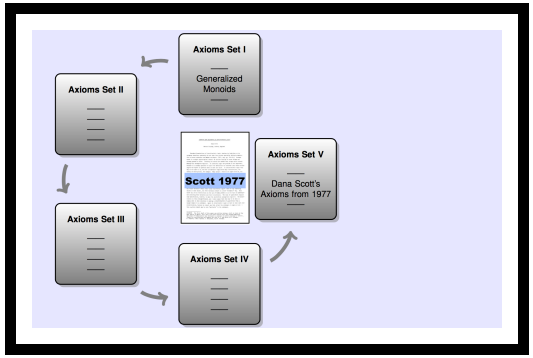
- Consistency? — Nitpick finds a model.
- Consistency when assuming $\exists x.\neg Ex$ — Nitpick does not find a model.
- lemma $(\exists x.\neg Ex) \rightarrow False$: **SLEDGEHAMMER**. (Problematic axioms: $A1, A2a, A3a$)

When interpreted in free logic, then the axioms of Freyd and Scedrov are flawed: Either all morphisms exist (i.e., $\cdot$ is total), or the axioms are inconsistent.

Demo


NORTH-HOLLAND
MATHEMATICAL LIBRARY
Categories, Allegories
PETER J. FREYD
ANDRE SCEDROV
North-Holland
FreydScedrovInconsistency
AxiomaticCategoryTheory.thy (~/chris/trunk/tex/talks/2017-BMG-Tag/)
context -- {* Axiom Set VI (Freyd and Scedrov) in their notation *}
assumes
A1: "E(x·y) ↔ (x□ ≅ □y)" and
A2a: "((□x)□) ≅ □x" and
A2b: "□(x□) ≅ □x" and
A3a: "(□x)·x ≅ x" and
A3b: "x·(x□) ≅ x" and
A4a: "□(x·y) ≅ □(x·(□y))" and
A4b: "(x·y)□ ≅ ((x□)·y)□" and
A5: "x·(y·z) ≅ (x·y)·z"
begin
lemma InconsistencyAutomatic: "(∃x. ¬(E x)) ⟶ False"
lemma InconsistencyInteractive: assumes NEx: "∃x. ¬(E x)" shows Fa
proof -
-- {* Let @{text "a"} be an undefined object *}
obtain a where 1: "¬(E a)" using assms by auto
-- {* We instantiate axiom @{text "A3a"} with @{text "a"}. *}
have 2: "(□a)·a ≅ a" using A3a by blast
-- {* By unfolding the definition of @{text "≅"} we get from 1 t
not defined. This is
easy to see, since if @{text "(□a)·a"} were defined, we als
00:00
-01:57
Search:
100%
goal (1 subgoal):
1. False ⟵ (∃x. ¬ (E x))
Output
Query
Sledgehammer
Symbols

Cats & Alligators



Categories: Axioms Set VI (Freyd and Scedrov, when corrected)

A1	$E(x \cdot y) \leftrightarrow dom\ x \simeq cod\ y$
A2a	$cod(dom\ x) \cong dom\ x$
A2b	$dom(cod\ y) \cong cod\ y$
A3a	$x \cdot (dom\ x) \cong x$
A3b	$(cod\ y) \cdot y \cong y$
A4a	$dom(x \cdot y) \cong dom((dom\ x) \cdot y)$
A4b	$cod(x \cdot y) \cong cod(x \cdot (cod\ y))$
A5	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$

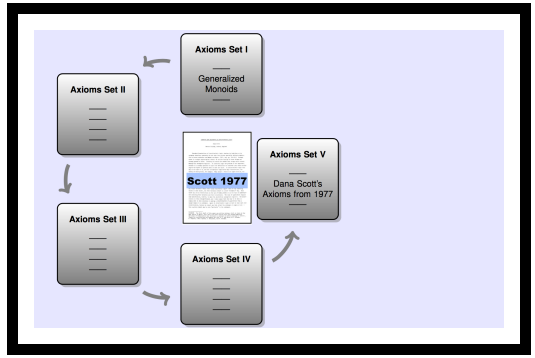
Experiments with Isabelle/HOL

- Consistency holds (also when $\exists x.\neg(Ex)$): confirmed by **Nitpick**.
- Axioms Set VI implies Axioms Set V: **Sledgehammer**.
- Axioms Set V implies Axioms Set VI: **Sledgehammer**.
- Redundancies:

— The $A4$-axioms are implied by the others: **Sledgehammer**.

— The $A2$-axioms are implied by the others: **Sledgehammer**.

Cats & Alligators



Categories: Axioms Set VI (Freyd and Scedrov, when corrected)

A1	$E(x \cdot y) \leftrightarrow dom\ x \simeq cod\ y$
A2a	$cod(dom\ x) \cong dom\ x$
A2b	$dom(cod\ y) \cong cod\ y$
A3a	$x \cdot (dom\ x) \cong x$
A3b	$(cod\ y) \cdot y \cong y$
A4a	$dom(x \cdot y) \cong dom((dom\ x) \cdot y)$
A4b	$cod(x \cdot y) \cong cod(x \cdot (cod\ y))$
A5	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$

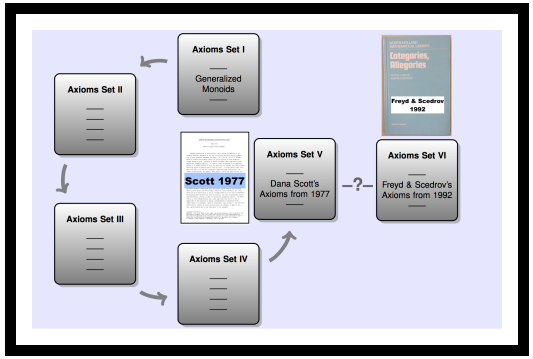
Experiments with Isabelle/HOL

- Consistency holds (also when $\exists x.\neg(Ex)$): confirmed by **Nitpick**.
- Axioms Set VI implies Axioms Set V: **Sledgehammer**.
- Axioms Set V implies Axioms Set VI: **Sledgehammer**.
- Redundancies:

— The $A4$-axioms are implied by the others: **Sledgehammer**.

— The $A2$-axioms are implied by the others: **Sledgehammer**.

Cats & Alligators



Maybe Freyd and Scedrov do not assume a free logic.
In algebraic theories free variables often range over existing objects only. However, we can formalise this as well:

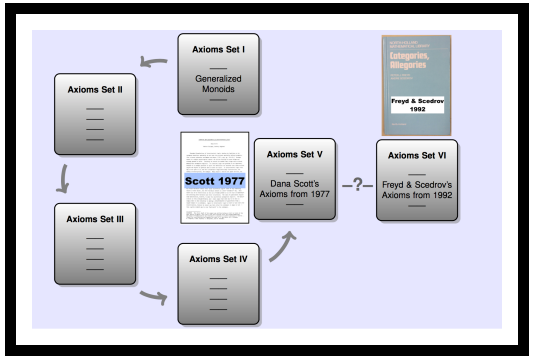
Categories: "Algebraic reading" of axiom set by Freyd and Scedrov.

A1	$\forall xy.\ E(x \cdot y) \leftrightarrow dom\ x \cong cod\ y$
A2a	$\forall x.\ cod(dom\ x) \cong dom\ x$
A2b	$\forall y.\ dom(cod\ y) \cong cod\ y$
A3a	$\forall x.\ x \cdot (dom\ x) \cong x$
A3b	$\forall y.\ (cod\ y) \cdot y \cong y$
A4a	$\forall xy.\ dom(x \cdot y) \cong dom((dom\ x) \cdot y)$
A4b	$\forall xy.\ cod(x \cdot y) \cong cod(x \cdot (cod\ y))$
A5	$\forall xyz.\ x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$

Experiments with Isabelle/HOL

- Consistency holds (also when $\exists x.\neg(Ex)$): confirmed by **Nitpick**.
- However, none of V-axioms are implied: **Nitpick**.
- For equivalence to V-axioms: add strictness of dom, cod, $\cdot$, **Sledgehammer**.

Cats & Alligators



Maybe Freyd and Scedrov do not assume a free logic.
In algebraic theories free variables often range over existing objects only. However, we can formalise this as well:

Categories: "Algebraic reading" of axiom set by Freyd and Scedrov.

A1	$\forall xy.\ E(x \cdot y) \leftrightarrow dom\ x \cong cod\ y$
A2a	$\forall x.\ cod(dom\ x) \cong dom\ x$
A2b	$\forall y.\ dom(cod\ y) \cong cod\ y$
A3a	$\forall x.\ x \cdot (dom\ x) \cong x$
A3b	$\forall y.\ (cod\ y) \cdot y \cong y$
A4a	$\forall xy.\ dom(x \cdot y) \cong dom((dom\ x) \cdot y)$
A4b	$\forall xy.\ cod(x \cdot y) \cong cod(x \cdot (cod\ y))$
A5	$\forall xyz.\ x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$

Experiments with Isabelle/HOL

But: Strictness is not mentioned in Freyd and Scedrov!
And it could not even be expressed axiomatically, when variables range over of existing objects only. This leaves us puzzled about their axiom system.

Hence, we better prefer the Axioms Set V by Scott (from 1977).

Very Recent Study: Axioms Set by Saunders Mac Lane (1948)

GROUPS, CATEGORIES AND DUALITY

By Saunders MacLane*

Department of Mathematics, University of Chicago

Communicated by Marshall Stone, May 1, 1948

It has long been recognized that the theorems of group theory display a certain duality. The concept of a lattice gives a partial expression for this duality, in that some of the theorems about groups which can be formulated in terms of the lattice of subgroups of a group display the customary lattice duality between meet (intersection) and join (union). The duality is not always present, in the sense that the lattice dual of a true theorem on groups need not be true; for example, a Jordan Holder theorem holds for certain ascending well-ordered infinite composition series, but not for the corresponding descending series.[1] Moreover, there are other striking group theoretic situations where a duality is present, but is not readily expressible in lattice-theoretic terms.

As an example, consider the direct product $D = G \times H$ of two groups

Very Recent Study: Axioms Set by Saunders Mac Lane (1948)

GROUPS, CATEGORIES AND DUALITY

By Saunders MacLane*

Department of Mathematics, University of Chicago

Communicated by Marshall Stone, May 1, 1948

It has long been recognized that the theorems of group theory display a certain duality. The concept of a lattice gives a partial expression for this duality, in that some of the theorems about groups which can be formulated in terms of the lattice of subgroups of a group display the

introduced the notion of a category.[6] A *category* is a class of "mappings" (say, homomorphisms) in which the product $\alpha\beta$ of certain pairs of mappings α and β is defined. A mapping e is called an *identity* if $\rho\alpha = \alpha$ and $\beta\rho = \beta$ whenever the products in question are defined. These products must satisfy the axioms:

(C-1). *If the products $\gamma\beta$ and $(\gamma\beta)\alpha$ are defined, so is $\beta\alpha$;*

(C-1′). *If the products $\beta\alpha$ and $\gamma(\beta\alpha)$ are defined, so is $\gamma\beta$;*

(C-2). *If the products $\gamma\beta$ and $\beta\alpha$ are defined, so are the products $(\gamma\beta)\alpha$ and $\gamma(\beta\alpha)$, and these products are equal.*

(C-3). *For each γ there is an identity e_D such that γe_D is defined;*

(C-4). *For each γ there is an identity e_R such that $e_R\gamma$ is defined.*

It follows that the identities e_D and e_R are unique; they may be called, respectively, the *domain* and the *range* of the given mapping γ. A mapping θ with a two-sided inverse is an *equivalence*.

These axioms are clearly self dual, and a dual theory of free and direct products may be constructed in any category in which such products exist.

Very Recent Study: Axioms Set by Saunders Mac Lane (1948)

GROUPS, CATEGORIES AND DUALITY

By Saunders MacLane*

Department of Mathematics, University of Chicago

Communicated by Marshall Stone, May 1, 1948

It has long been recognized that the theorems of group theory display a certain duality. The concept of a lattice gives a partial expression for this duality, in that some of the theorems about groups which can be formulated in terms of the lattice of subgroups of a group display the

introduced the notion of a category.[6] A *category* is a class of "mappings"

Equivalent to Axioms Set V (Scott, 1977)

— if strictness conditions are added —

(C-1). *If the products* $\gamma\beta$ *and* $(\gamma\beta)\alpha$ *are defined, so is* $\beta\alpha$;

(C-1′). *If the products* $\beta\alpha$ *and* $\gamma(\beta\alpha)$ *are defined, so is* $\gamma\beta$;

(C-2). *If the products* $\gamma\beta$ *and* $\beta\alpha$ *are defined, so are the products* $(\gamma\beta)\alpha$ *and* $\gamma(\beta\alpha)$, *and these products are equal.*

(C-3). *For each* γ *there is an identity* e_D *such that* γe_D *is defined;*

(C-4). *For each* γ *there is an identity* e_R *such that* $e_R\gamma$ *is defined.*

It follows that the identities e_D and e_R are unique; they may be called, respectively, the *domain* and the *range* of the given mapping γ. A mapping θ with a two-sided inverse is an *equivalence.*

These axioms are clearly self dual, and a dual theory of free and direct products may be constructed in any category in which such products exist.

Axioms Set by Saunders Mac Lane (1948)

As before, we adopt an algebraic reading and add an explicit strictness condition.

Categories: Axioms Set by Mac Lane

C0 $E(\gamma \cdot \beta) \rightarrow (E\gamma \wedge E\beta)$ **(added by us)**

C1 $\forall \gamma, \beta, \alpha.\ (E(\gamma \cdot \beta) \wedge E((\gamma \cdot \beta) \cdot \alpha)) \rightarrow E(\beta \cdot \alpha)$

C1' $\forall \gamma, \beta, \alpha.\ (E(\beta \cdot \alpha) \wedge E(\gamma \cdot (\beta \cdot \alpha)) \rightarrow E(\gamma \cdot \beta)$

C2 $\forall \gamma, \beta, \alpha.\ (E(\gamma \cdot \beta) \wedge E(\beta \cdot \alpha)) \rightarrow$
$(E((\gamma \cdot \beta) \cdot \alpha) \wedge E(\gamma \cdot (\beta \cdot \alpha)) \wedge ((\gamma \cdot \beta) \cdot \alpha) = (\gamma \cdot (\beta \cdot \alpha)))$

C3 $\forall \gamma.\ \exists eD.\ IDMcL(eD) \wedge E(\gamma \cdot eD)$

C4 $\forall \gamma.\ \exists eR.\ IDMcL(eR) \wedge E(eR \cdot \gamma)$

where $IDMcL(\rho) \equiv (\forall \alpha.\ E(\rho \cdot \alpha) \rightarrow \rho \cdot \alpha = \alpha) \wedge (\forall \beta.\ E(\beta \cdot \rho) \rightarrow \beta \cdot \rho = \beta)$

Consistency holds (also when $\exists x.\neg(Ex)$): confirmed by **Nitpick**.

Axioms Set by Saunders Mac Lane (1948)

As before, we adopt an algebraic reading and add an explicit strictness condition.

Categories: Axioms Set by Mac Lane

C0 $E(\gamma \cdot \beta) \to (E\gamma \wedge E\beta)$ **(added by us)**

C1 $\forall \gamma, \beta, \alpha.\ (E(\gamma \cdot \beta) \wedge E((\gamma \cdot \beta) \cdot \alpha)) \to E(\beta \cdot \alpha)$

C1' $\forall \gamma, \beta, \alpha.\ (E(\beta \cdot \alpha) \wedge E(\gamma \cdot (\beta \cdot \alpha)) \to E(\gamma \cdot \beta)$

C2 $\forall \gamma, \beta, \alpha.\ (E(\gamma \cdot \beta) \wedge E(\beta \cdot \alpha)) \to$
$(E((\gamma \cdot \beta) \cdot \alpha) \wedge E(\gamma \cdot (\beta \cdot \alpha)) \wedge ((\gamma \cdot \beta) \cdot \alpha) = (\gamma \cdot (\beta \cdot \alpha)))$

C3 $\forall \gamma.\ \exists eD.\ IDMcL(eD) \wedge E(\gamma \cdot eD)$

C4 $\forall \gamma.\ \exists eR.\ IDMcL(eR) \wedge E(eR \cdot \gamma)$

where $IDMcL(\rho) \equiv (\forall \alpha.\ E(\rho \cdot \alpha) \to \rho \cdot \alpha = \alpha) \wedge (\forall \beta.\ E(\beta \cdot \rho) \to \beta \cdot \rho = \beta)$

Consistency holds (also when $\exists x.\neg(Ex)$): confirmed by **Nitpick**.

This axioms set is equivalent to (as shown by Sledgehammer)

Categories: Axioms Set I

S_i	Strictness	$E(x \cdot y) \to (Ex \wedge Ey)$
E_i	Existence	$E(x \cdot y) \leftarrow (Ex \wedge Ey \wedge (\exists z.z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y))$
A_i	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_i	Codomain	$\forall y.\exists i.ID(i) \wedge i \cdot y \cong y$
D_i	Domain	$\forall x.\exists j.ID(j) \wedge x \cdot j \cong x$

Axioms Set by Saunders Mac Lane (1948)

How about the Skolemized variant?

Categories: Axioms Set by Mac Lane

C0 $(E(\gamma \cdot \beta) \rightarrow (E\gamma \wedge E\beta)) \wedge (E(dom\ \gamma) \rightarrow (E\gamma)) \wedge (E(cod\ \gamma) \rightarrow (E\gamma))$ **(added)**

C1 $\forall \gamma, \beta, \alpha.\ (E(\gamma \cdot \beta) \wedge E((\gamma \cdot \beta) \cdot \alpha)) \rightarrow E(\beta \cdot \alpha)$

C1' $\forall \gamma, \beta, \alpha.\ (E(\beta \cdot \alpha) \wedge E(\gamma \cdot (\beta \cdot \alpha)) \rightarrow E(\gamma \cdot \beta)$

C2 $\forall \gamma, \beta, \alpha.\ (E(\gamma \cdot \beta) \wedge E(\beta \cdot \alpha)) \rightarrow$
$(E((\gamma \cdot \beta) \cdot \alpha) \wedge E(\gamma \cdot (\beta \cdot \alpha)) \wedge ((\gamma \cdot \beta) \cdot \alpha) = (\gamma \cdot (\beta \cdot \alpha)))$

C3 $\forall \gamma.\ IDMcL(dom\ \gamma) \wedge E(\gamma \cdot (dom\ \gamma))$

C4 $\forall \gamma.\ IDMcL(cod\ \gamma) \wedge E((cod\ \gamma) \cdot \gamma)$

Consistency holds (also when $\exists x.\neg(Ex)$): confirmed by **NITPICK**.

Axioms Set by Saunders Mac Lane (1948)

How about the Skolemized variant?

Categories: Axioms Set by Mac Lane

C0	$(E(\gamma \cdot \beta) \rightarrow (E\gamma \wedge E\beta)) \wedge (E(dom\ \gamma) \rightarrow (E\gamma)) \wedge (E(cod\ \gamma) \rightarrow (E\gamma))$	**(added)**
C1	$\forall \gamma, \beta, \alpha.\ (E(\gamma \cdot \beta) \wedge E((\gamma \cdot \beta) \cdot \alpha)) \rightarrow E(\beta \cdot \alpha)$	
C1'	$\forall \gamma, \beta, \alpha.\ (E(\beta \cdot \alpha) \wedge E(\gamma \cdot (\beta \cdot \alpha)) \rightarrow E(\gamma \cdot \beta)$	
C2	$\forall \gamma, \beta, \alpha.\ (E(\gamma \cdot \beta) \wedge E(\beta \cdot \alpha)) \rightarrow$ $(E((\gamma \cdot \beta) \cdot \alpha) \wedge E(\gamma \cdot (\beta \cdot \alpha)) \wedge ((\gamma \cdot \beta) \cdot \alpha) = (\gamma \cdot (\beta \cdot \alpha)))$	
C3	$\forall \gamma.\ IDMcL(dom\ \gamma) \wedge E(\gamma \cdot (dom\ \gamma))$	
C4	$\forall \gamma.\ IDMcL(cod\ \gamma) \wedge E((cod\ \gamma) \cdot \gamma)$	

Consistency holds (also when $\exists x. \neg(Ex)$): confirmed by **Nitpick**.

This axioms set is equivalent to (as shown by Sledgehammer)

Categories: Axioms Set V (Scott, 1977)

$S1$	Strictness	$E(dom\ x) \rightarrow Ex$
$S2$	Strictness	$E(cod\ y) \rightarrow Ey$
$S3$	Existence	$E(x \cdot y) \leftrightarrow dom\ x \simeq cod\ y$
$S4$	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
$S5$	Codomain	$(cod\ y) \cdot y \cong y$
$S6$	Domain	$x \cdot (dom\ x) \cong x$

Axioms Set by Saunders Mac Lane (1948)

How about the Skolemized variant?

Categories: Axioms Set by Mac Lane

C0 $(E(\gamma \cdot \beta) \to (E\gamma \wedge E\beta)) \wedge (E(dom\ \gamma) \to (E\gamma)) \wedge (E(cod\ \gamma) \to (E\gamma))$ **(added)**

C1 $\forall \gamma, \beta, \alpha.\ (E(\gamma \cdot \beta) \wedge E((\gamma \cdot \beta) \cdot \alpha)) \to E(\beta \cdot \alpha)$

C1' $\forall \gamma, \beta, \alpha.\ (E(\beta \cdot \alpha) \wedge E(\gamma \cdot (\beta \cdot \alpha)) \to E(\gamma \cdot \beta)$

C2 $\forall \gamma, \beta, \alpha.\ (E(\gamma \cdot \beta) \wedge E(\beta \cdot \alpha)) \to$
$(E((\gamma \cdot \beta) \cdot \alpha) \wedge E(\gamma \cdot (\beta \cdot \alpha)) \wedge ((\gamma \cdot \beta) \cdot \alpha) = (\gamma \cdot (\beta \cdot \alpha)))$

C3 $\forall \gamma.\ IDMcL(dom\ \gamma) \wedge E(\gamma \cdot (dom\ \gamma))$

C4 $\forall \gamma.\ IDMcL(cod\ \gamma) \wedge E((cod\ \gamma) \cdot \gamma)$

Consistency holds (also when $\exists x.\neg(Ex)$): confirmed by **Nitpick**.

See also our "Archive of Formal Proofs" entry at:
`https://www.isa-afp.org/entries/AxiomaticCategoryTheory.html`

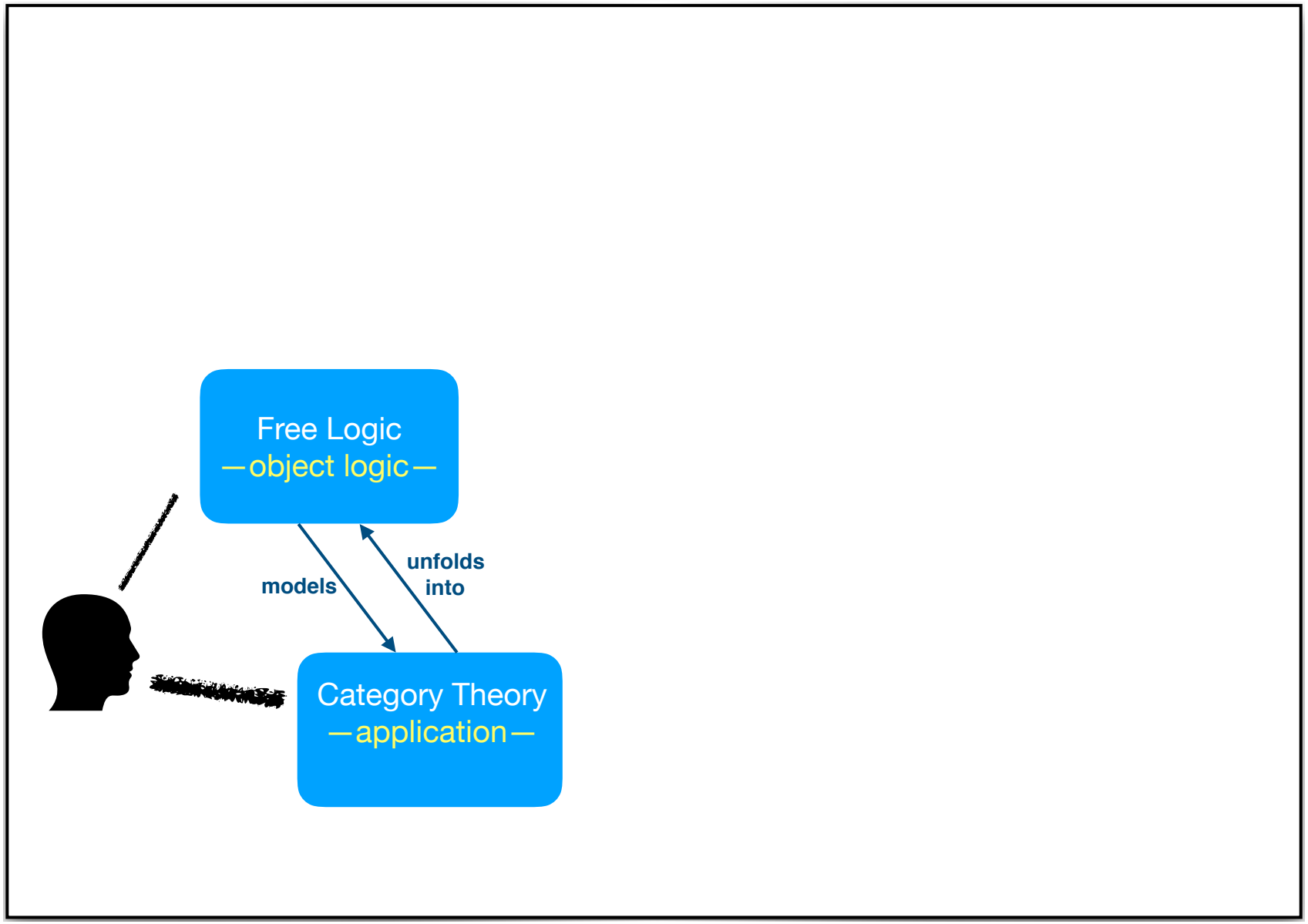
Part D — Discussion: Role of SMT solvers and ATPs

Discussion: Role of SMT in this work?

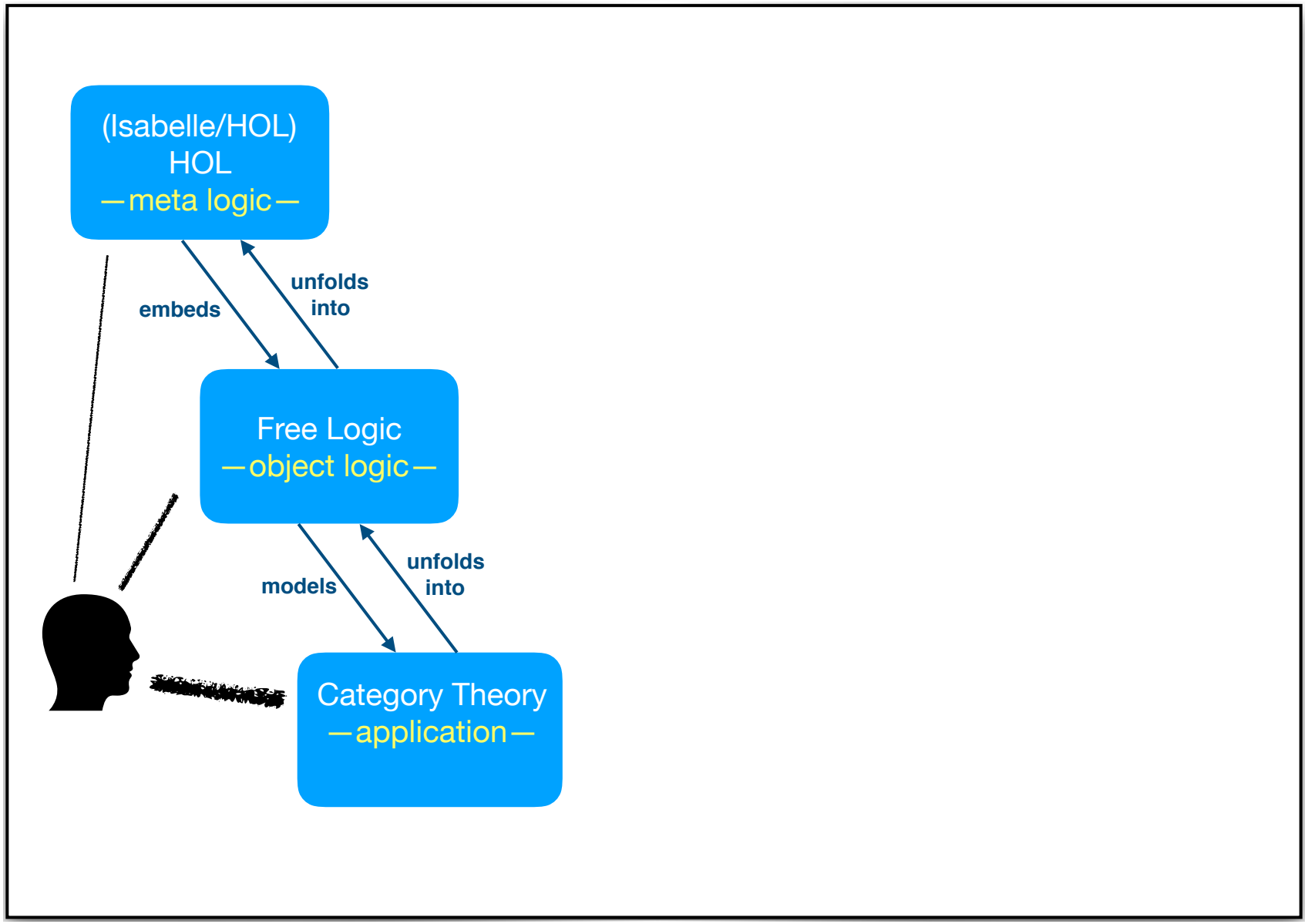
Discussion: Role of SMT in this work?

Category Theory
—application—

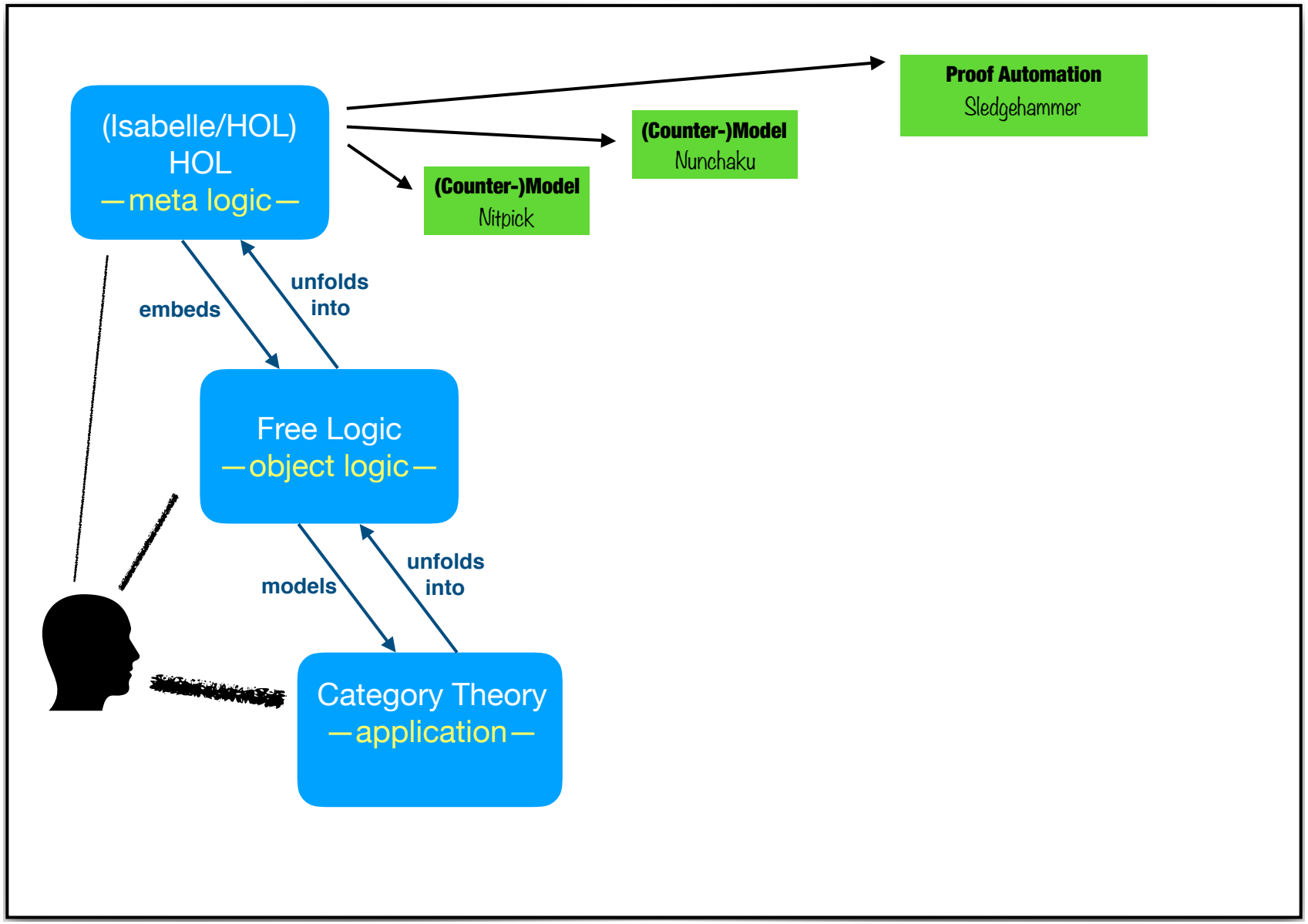
Discussion: Role of SMT in this work?



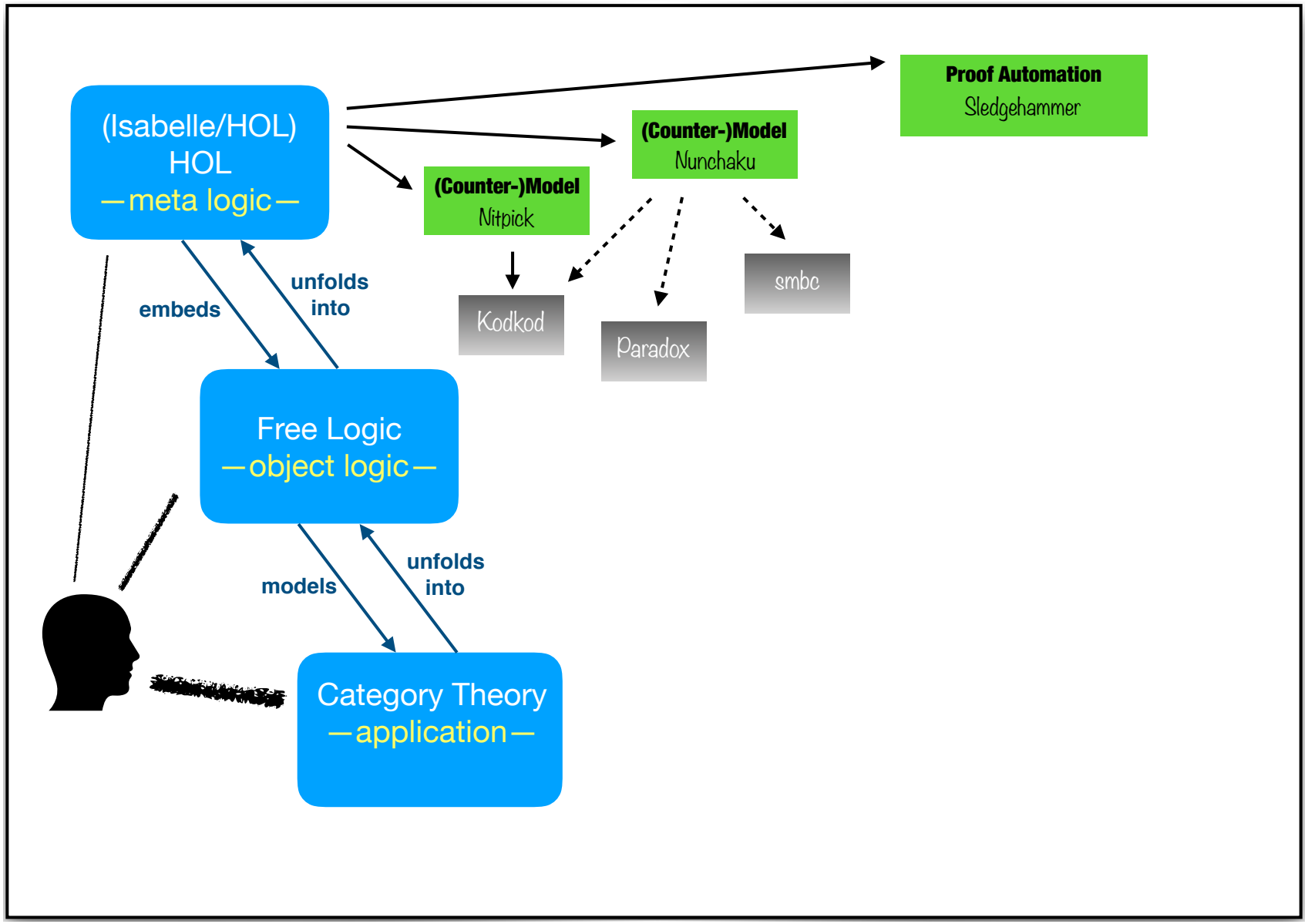
Discussion: Role of SMT in this work?



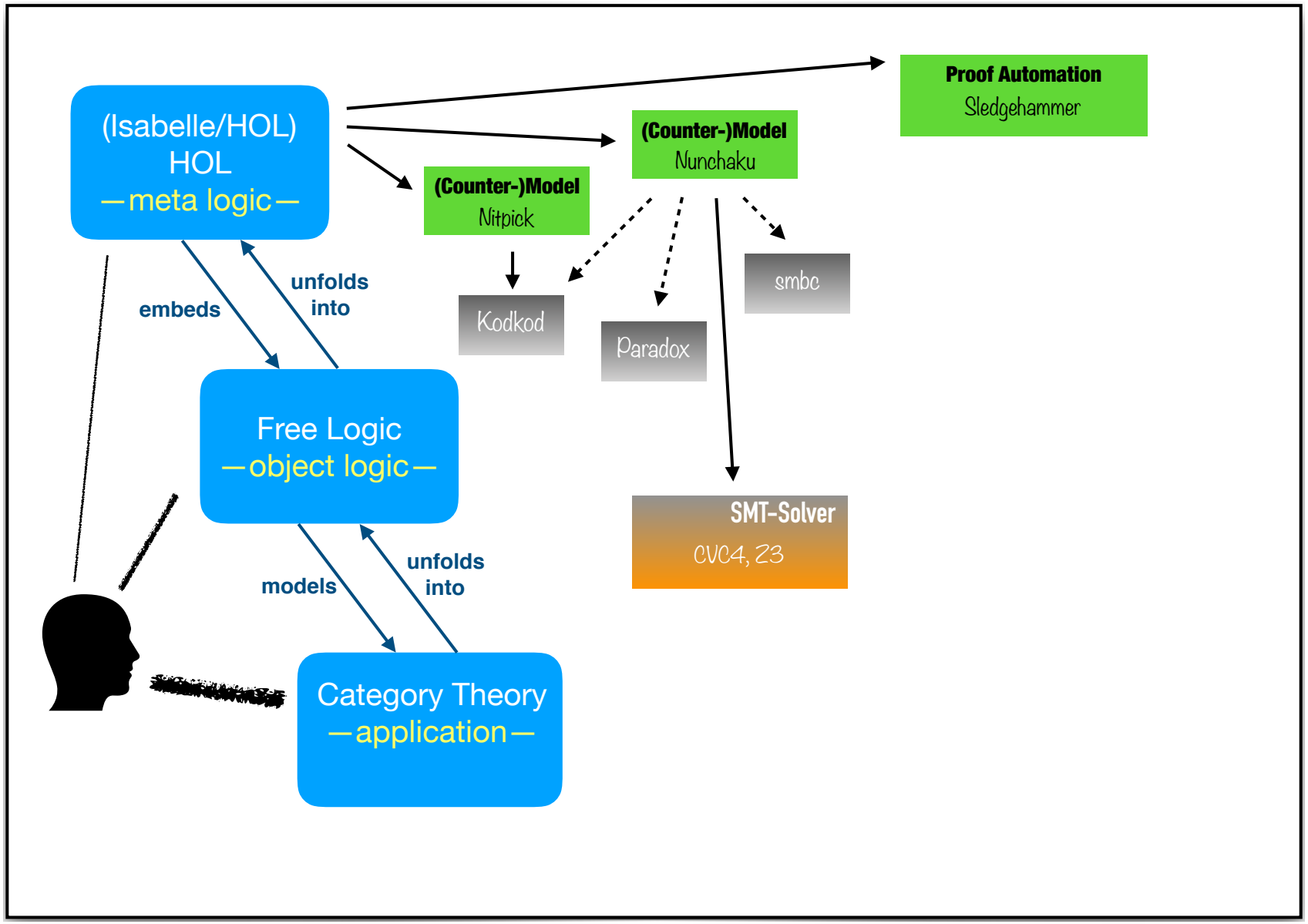
Discussion: Role of SMT in this work?



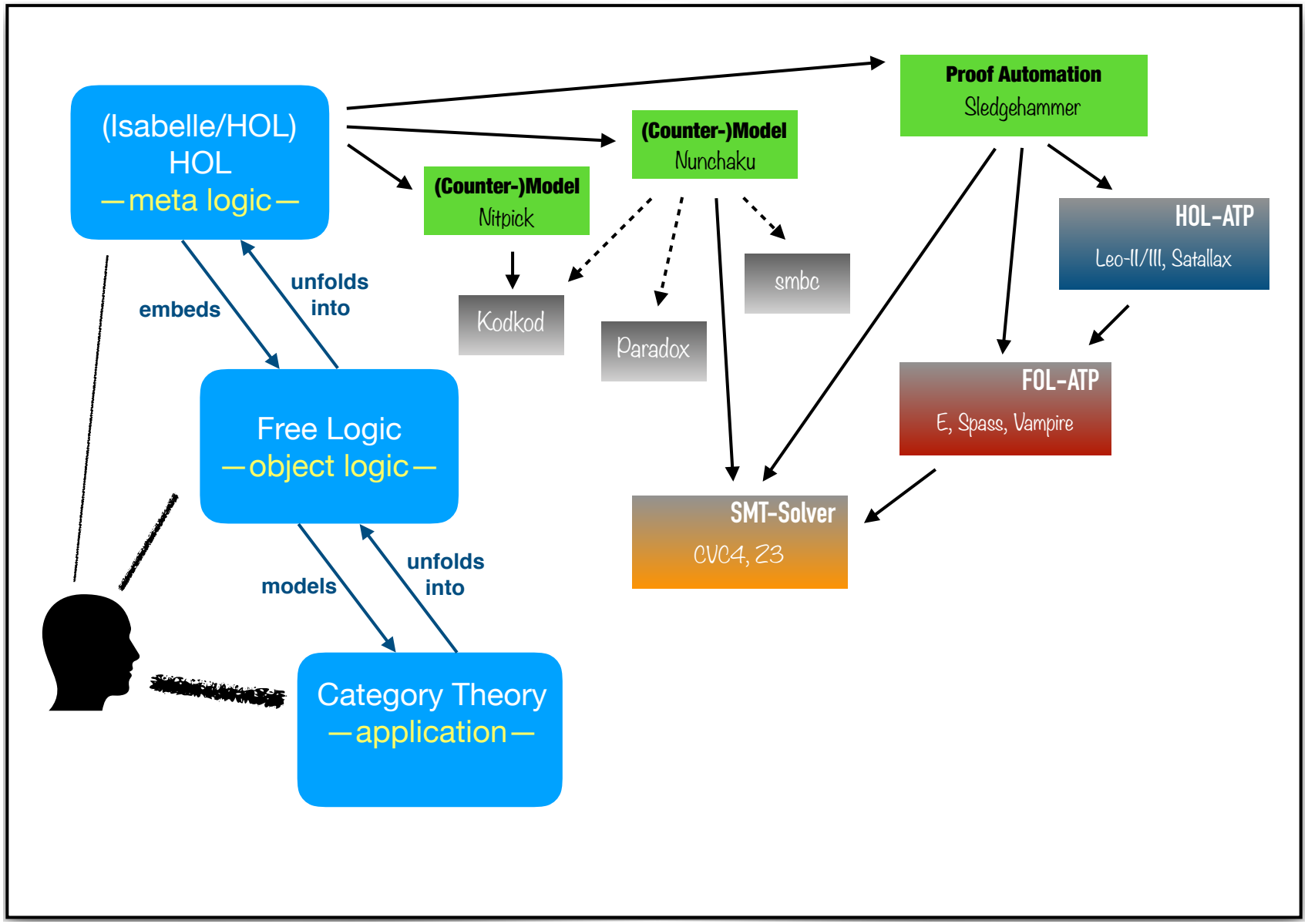
Discussion: Role of SMT in this work?



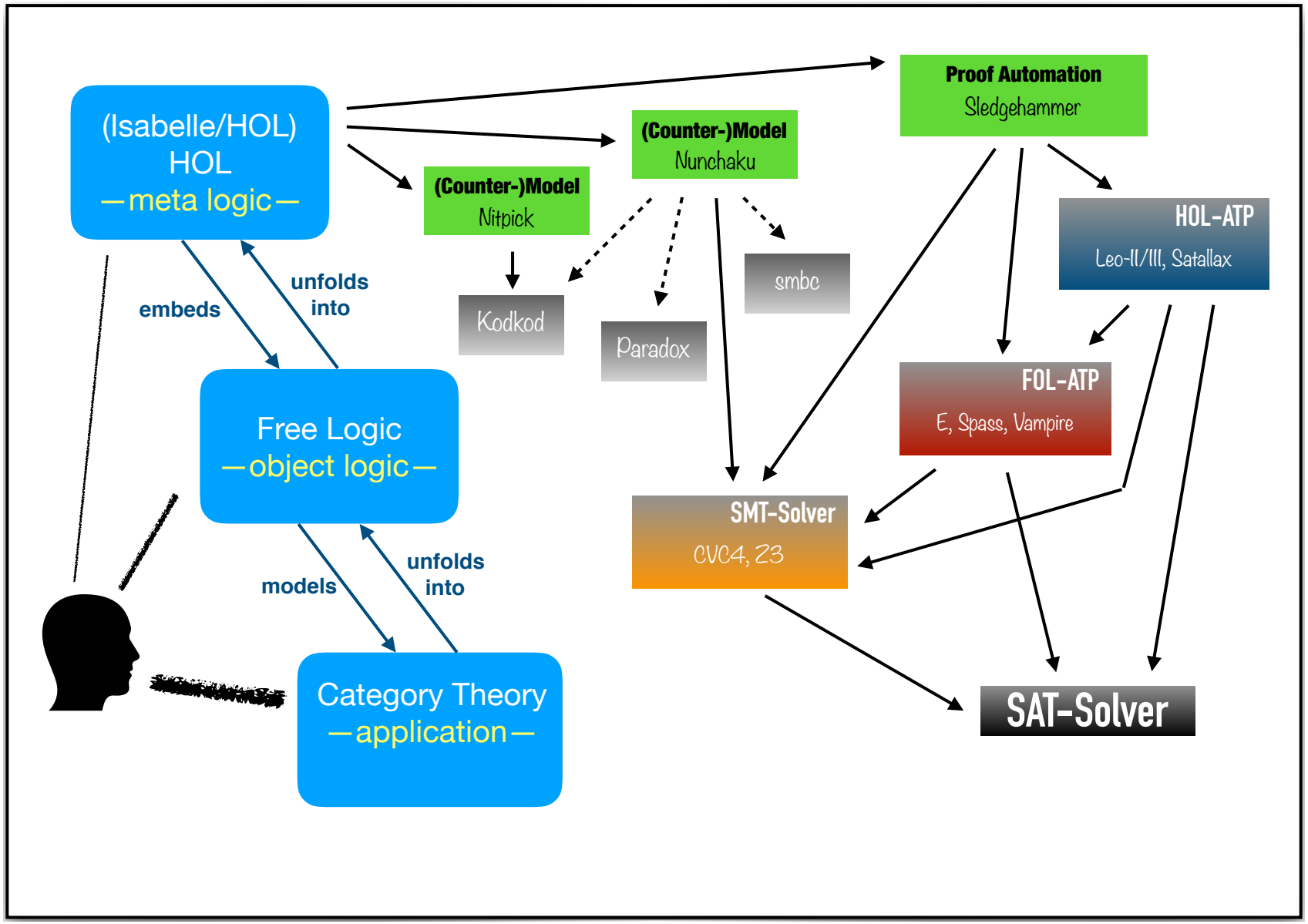
Discussion: Role of SMT in this work?



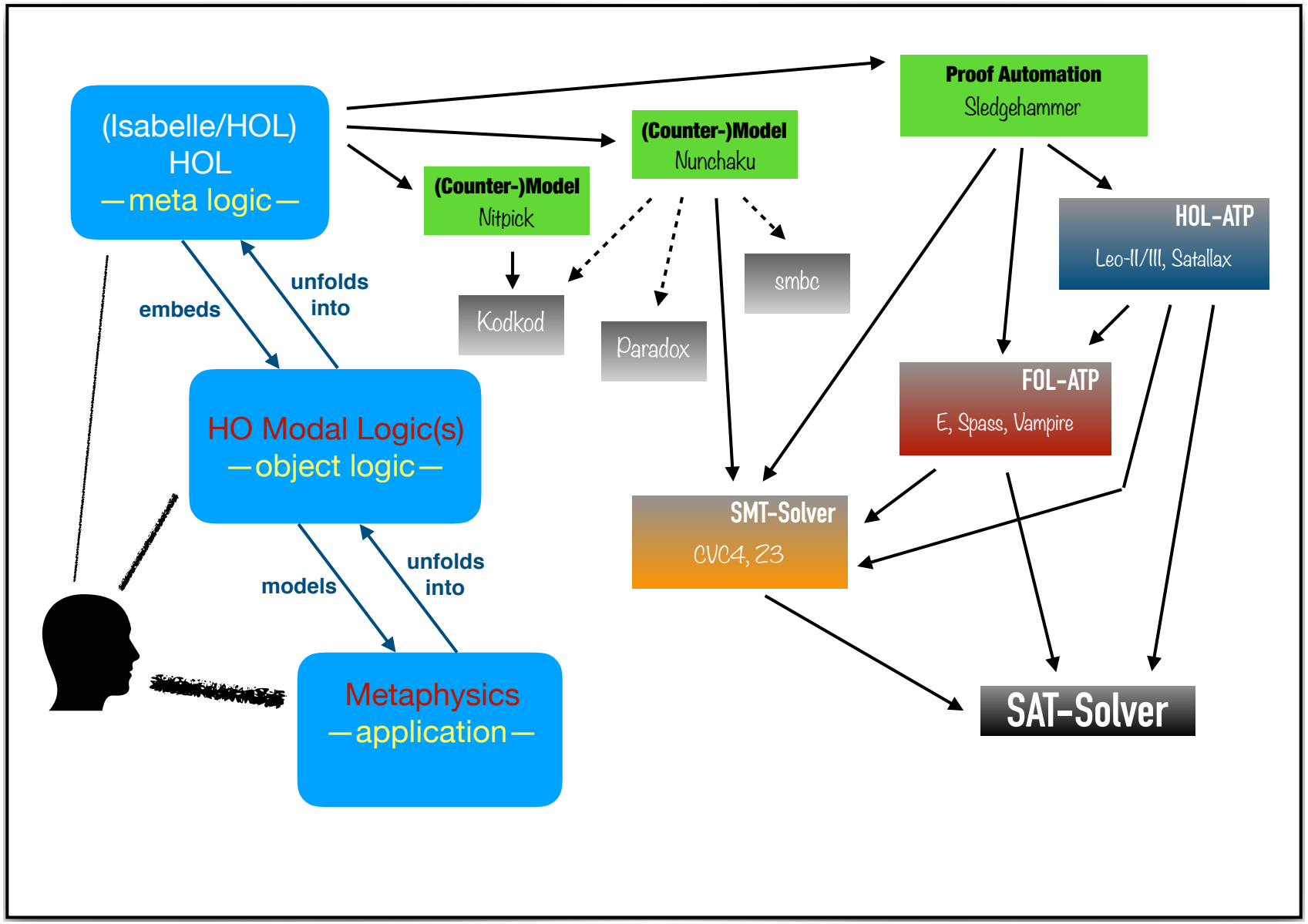
Discussion: Role of SMT in this work?



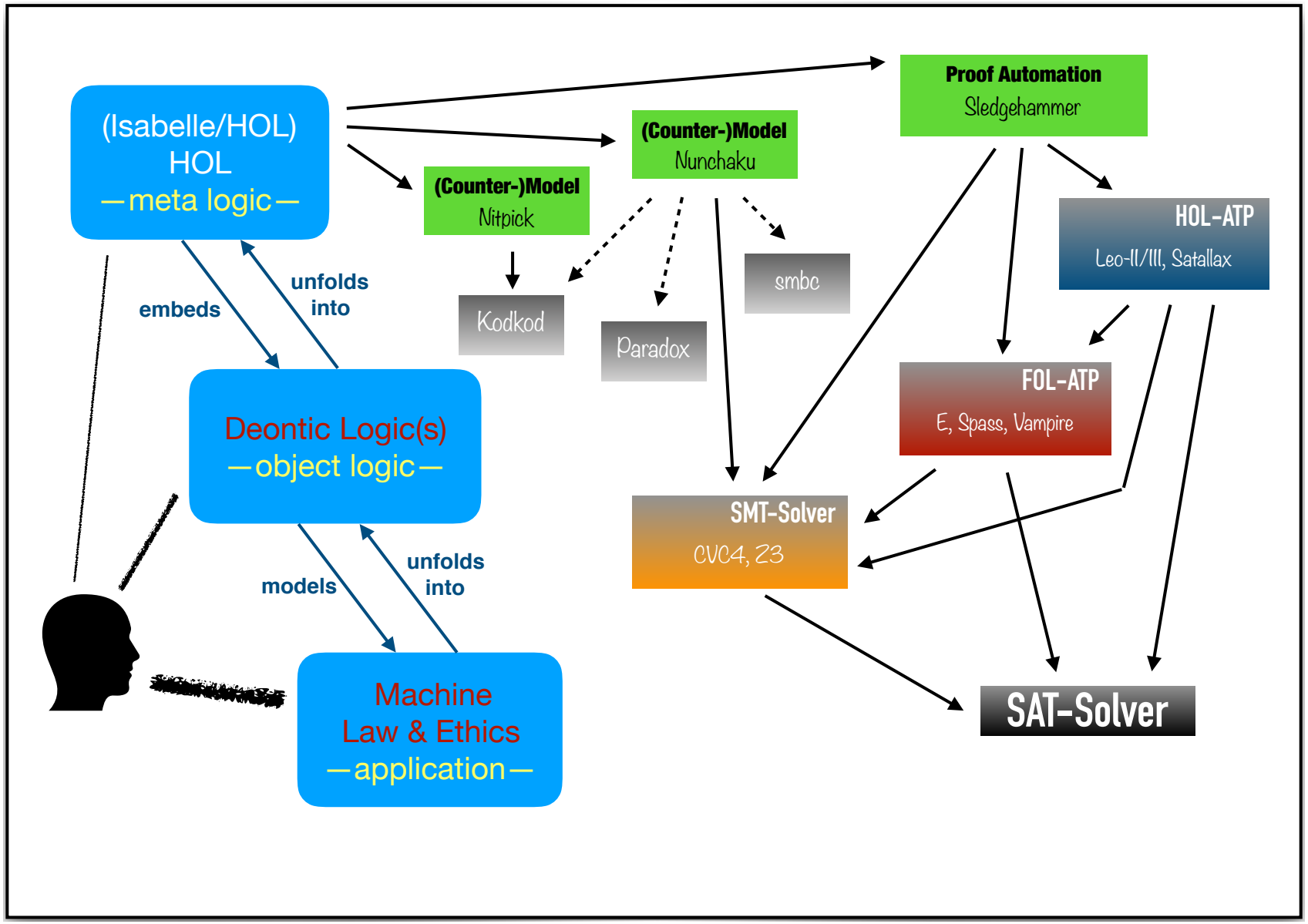
Discussion: Role of SMT in this work?



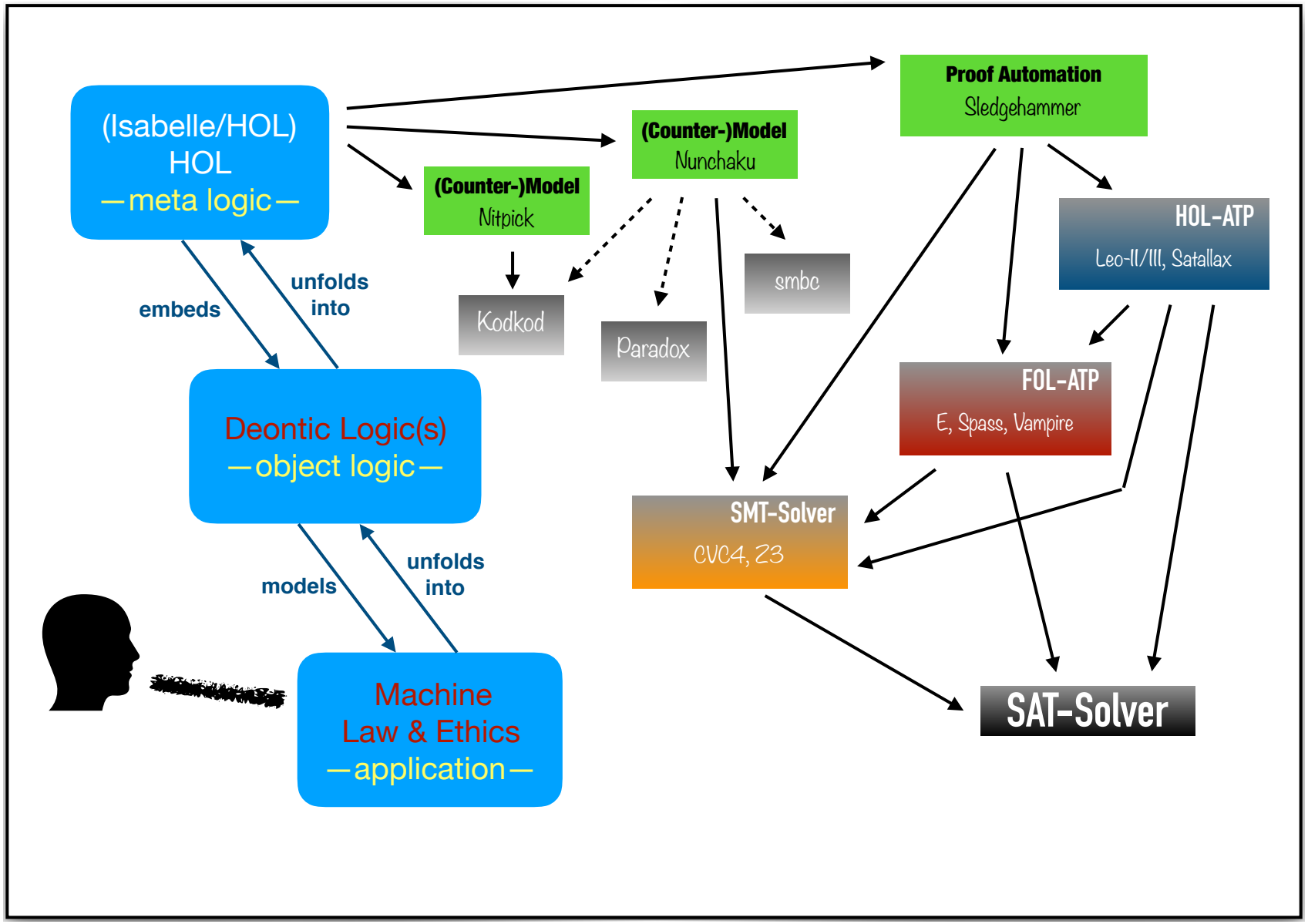
Discussion: Role of SMT in this work?



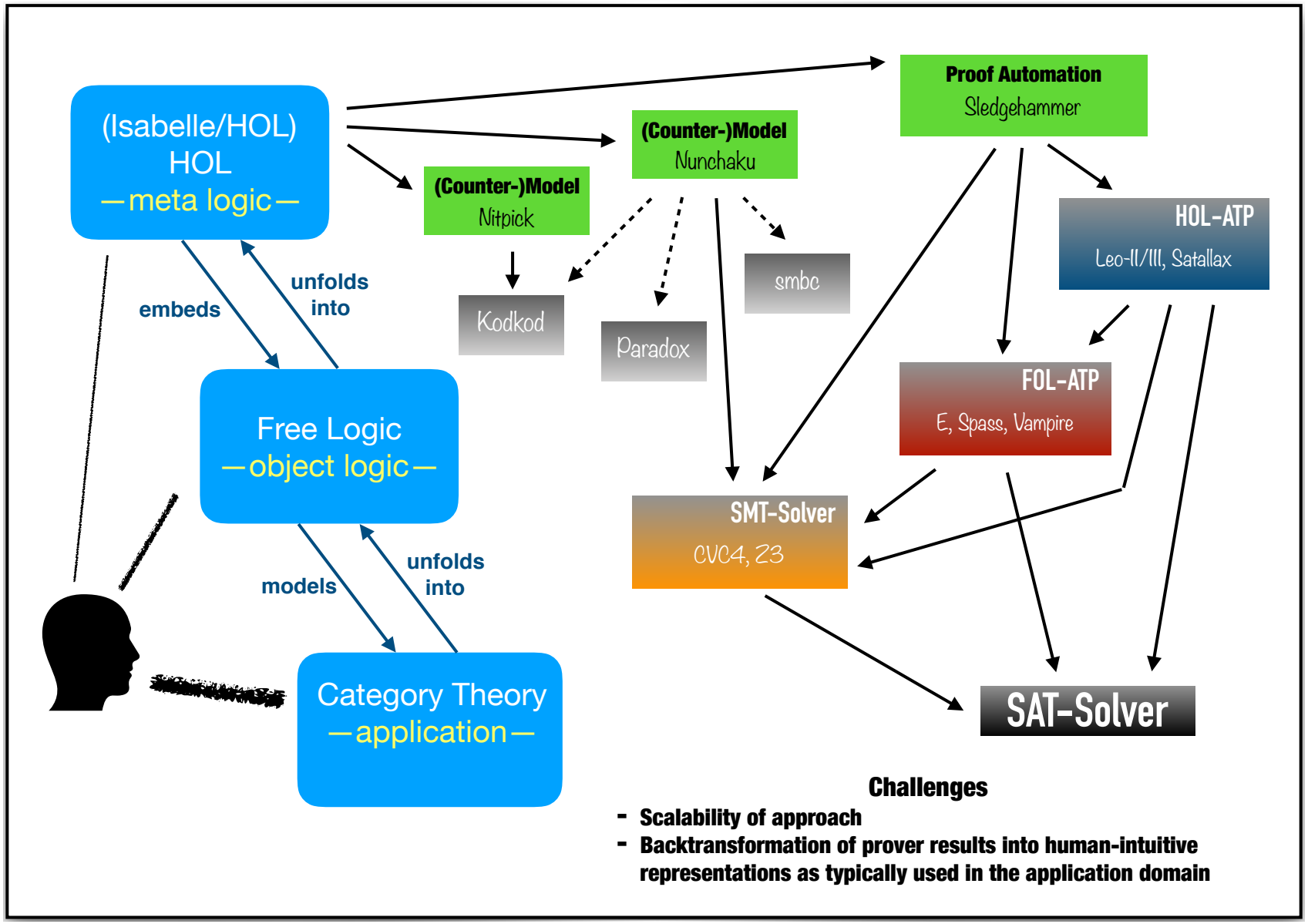
Discussion: Role of SMT in this work?



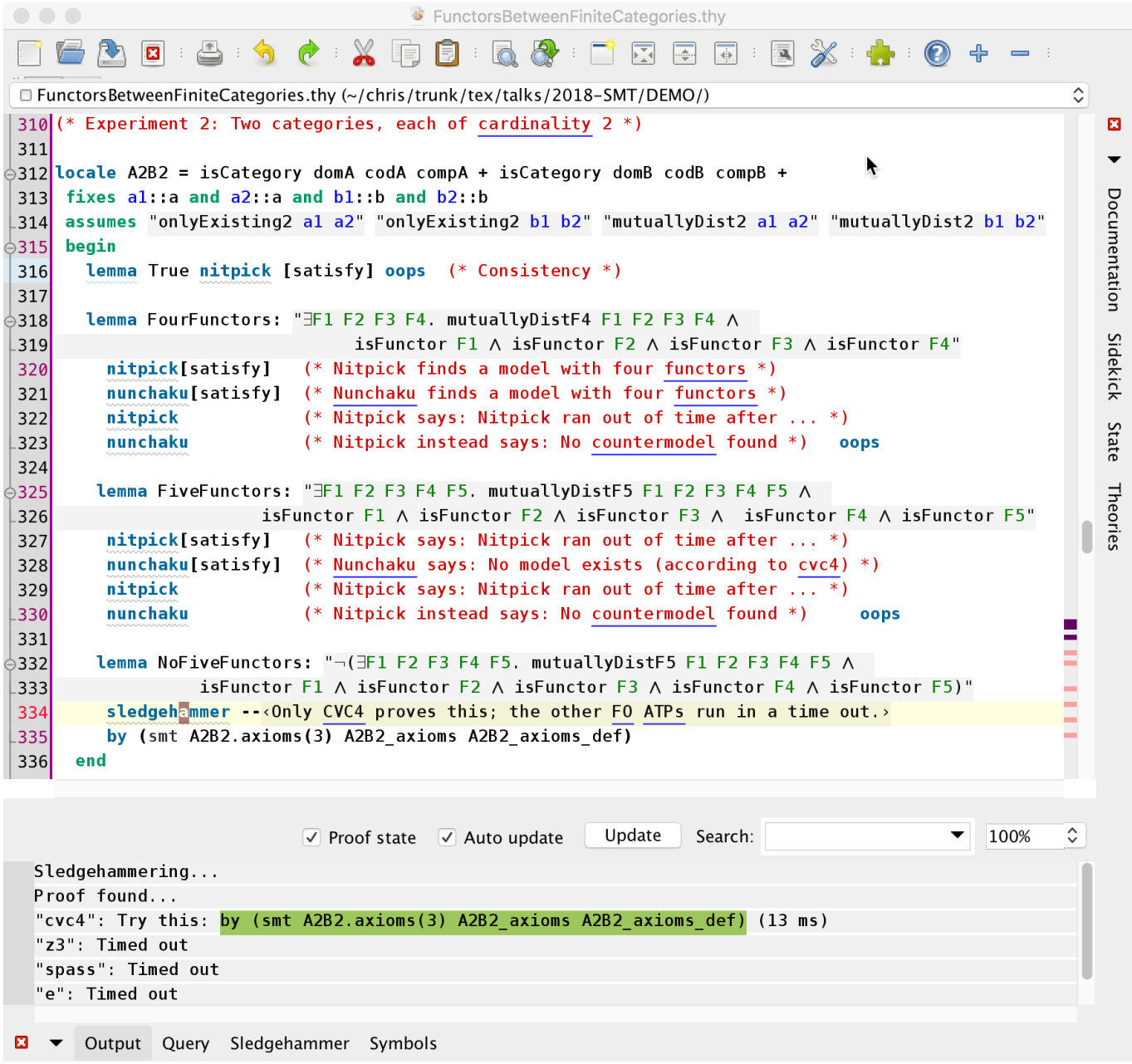
Discussion: Role of SMT in this work?



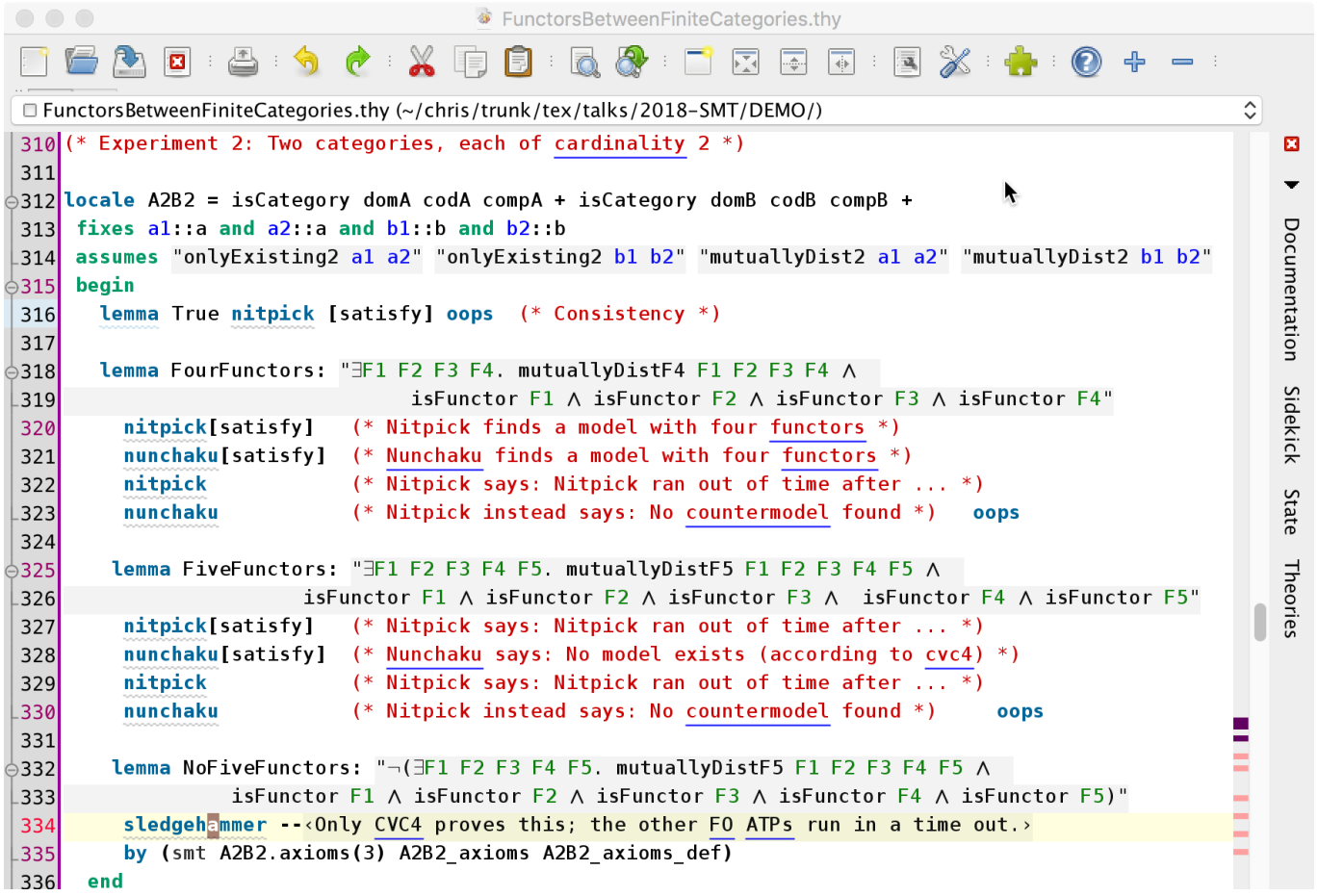
Discussion: Role of SMT in this work?



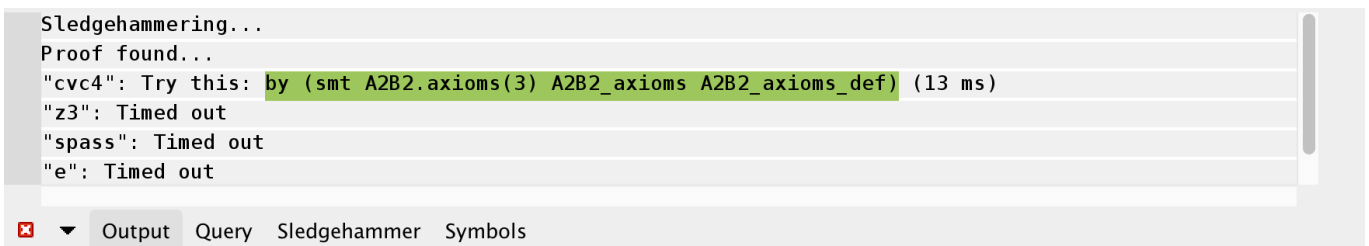
Discussion: Role of SMT solvers and ATPs



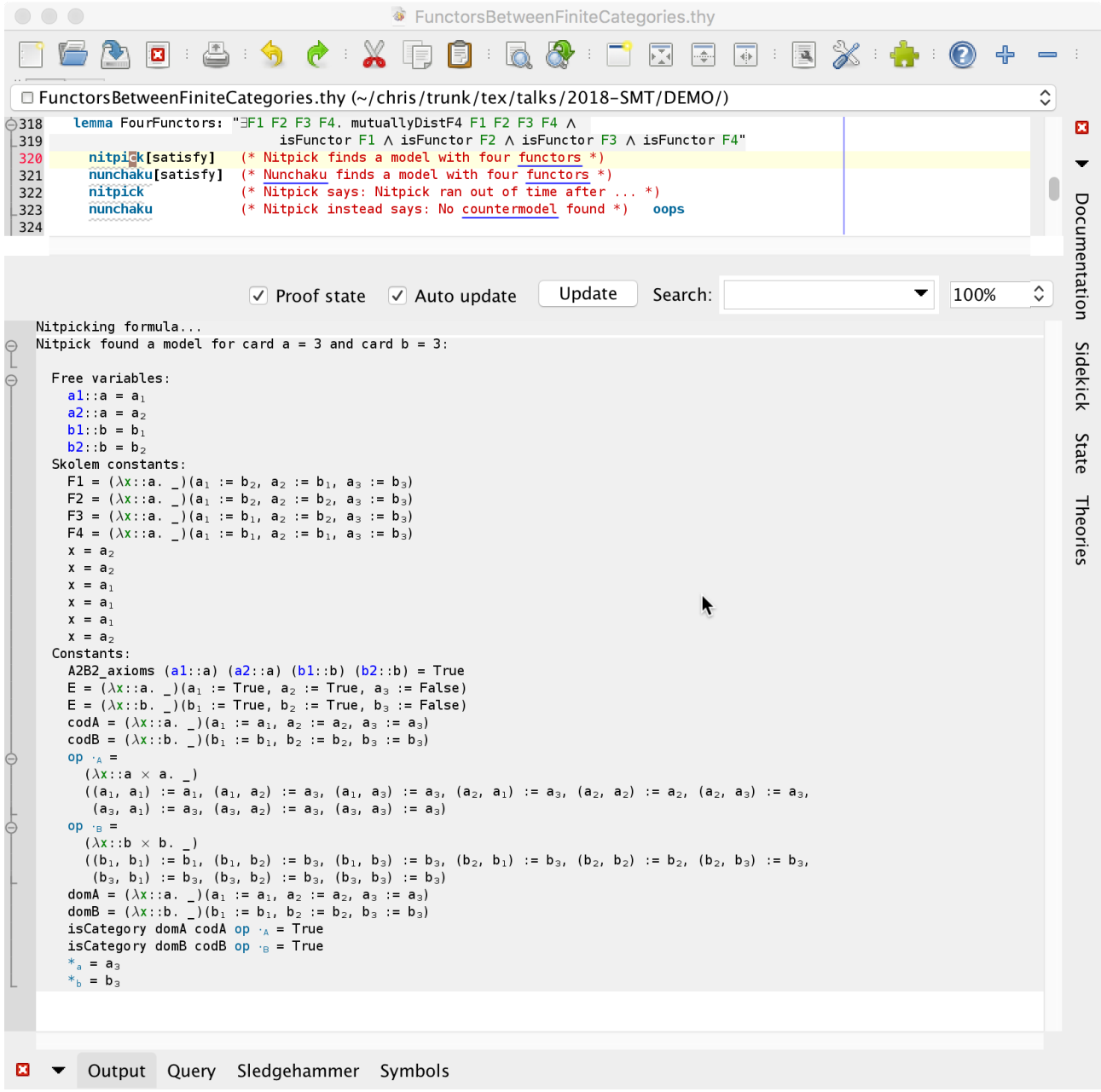
Discussion: Role of SMT solvers and ATPs



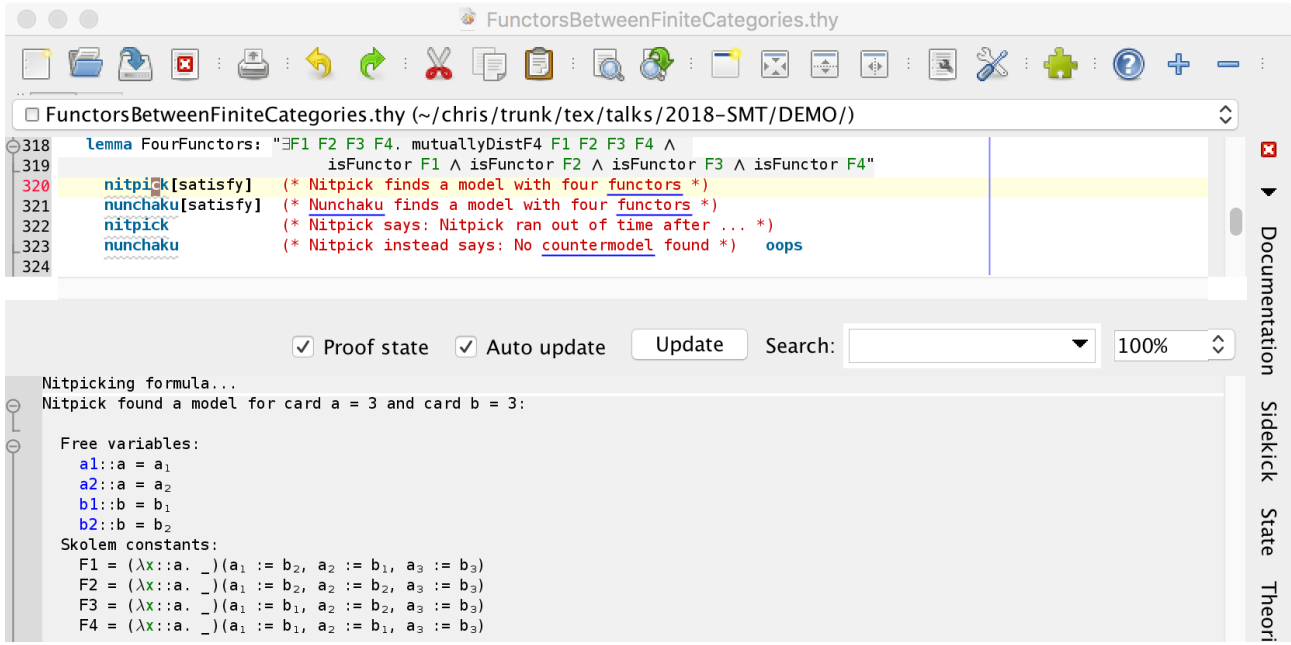
Occasionally only CVC4 answered (within default timeout)



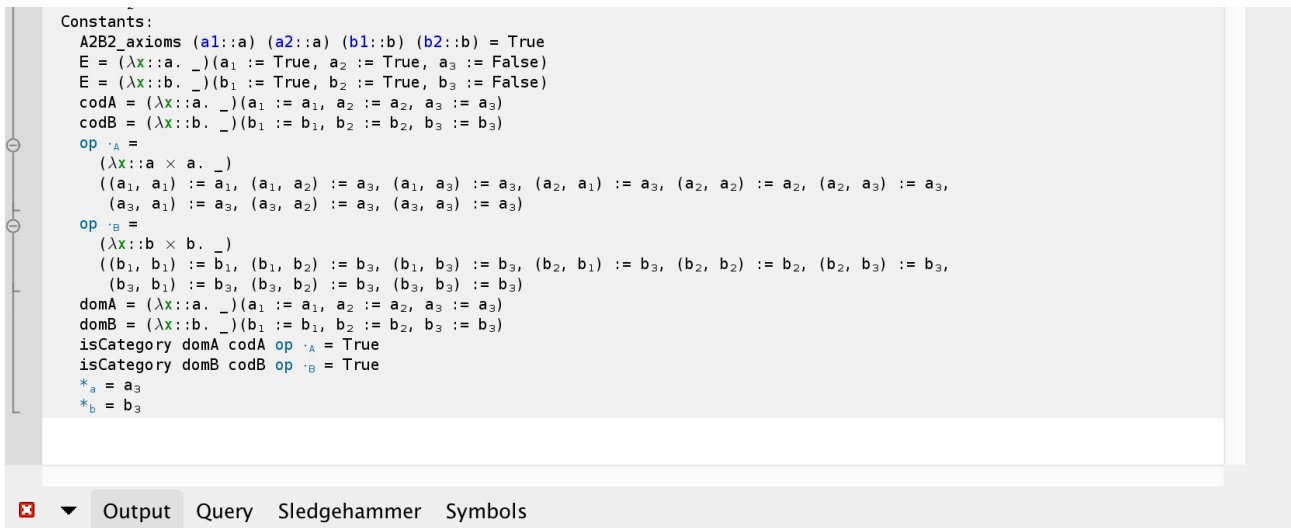
Discussion: Role of SMT solvers and ATPs



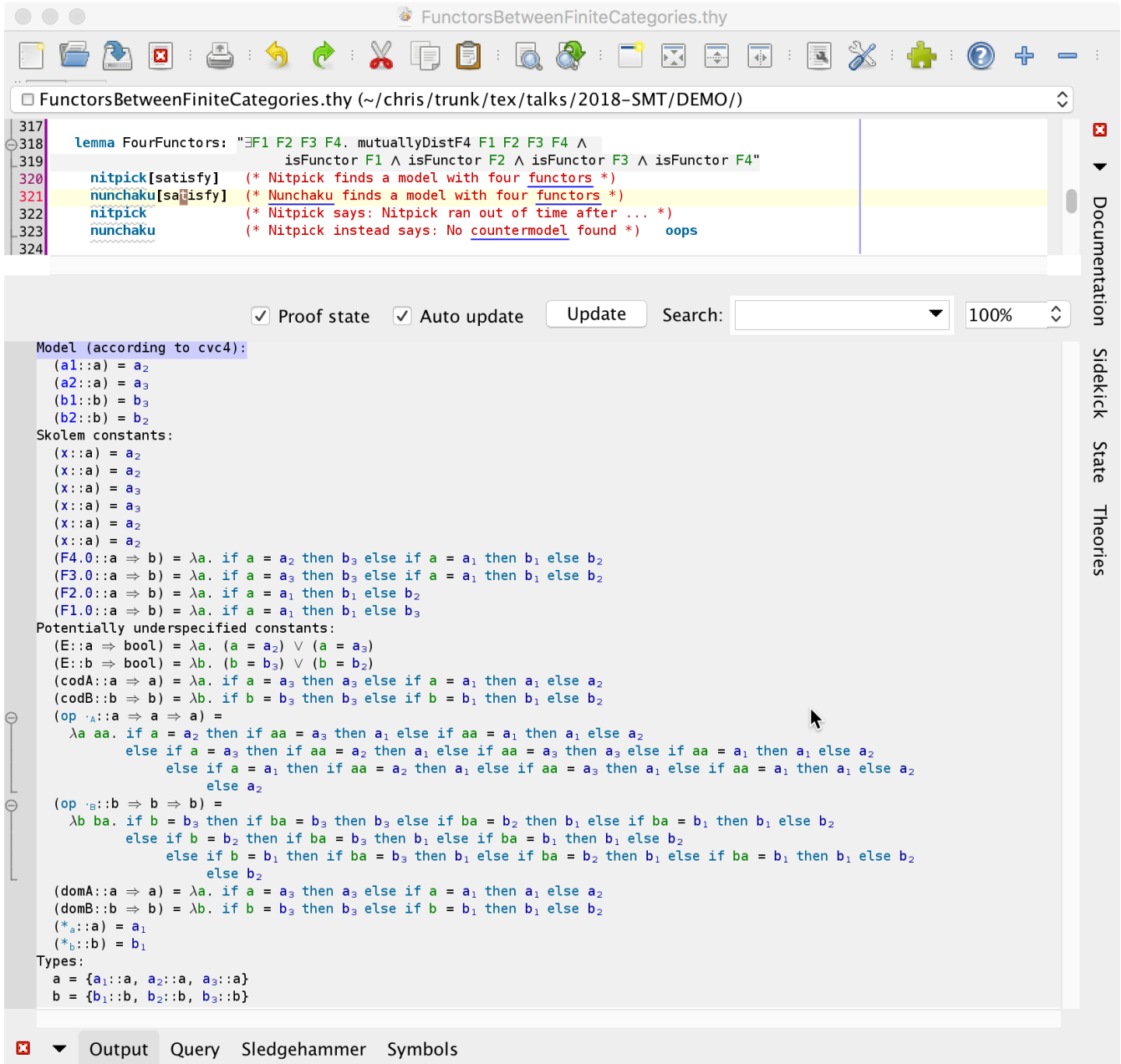
Discussion: Role of SMT solvers and ATPs



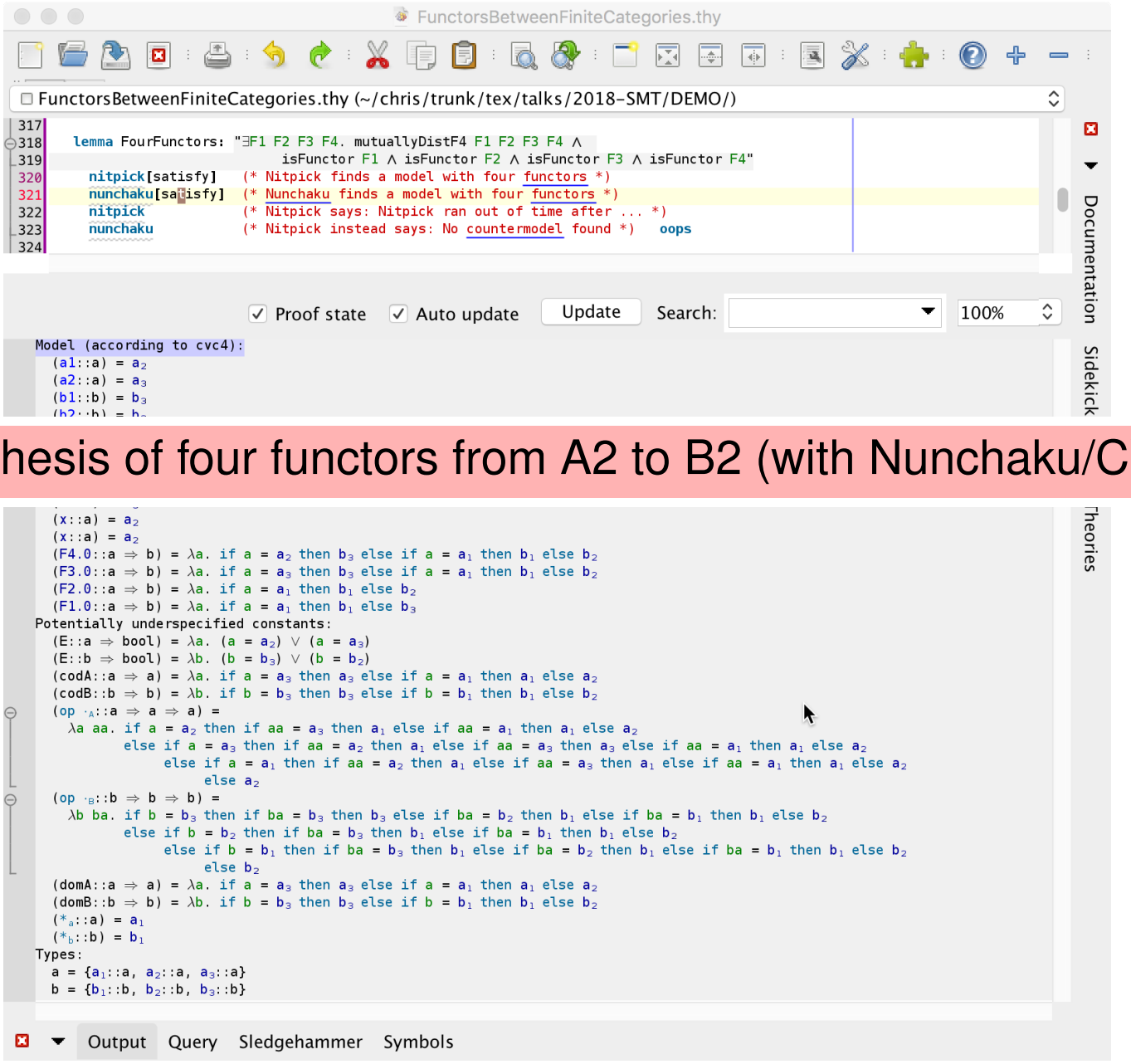
Synthesis of four different functors from A2 to B2 (with Nitpick)



Discussion: Role of SMT solvers and ATPs

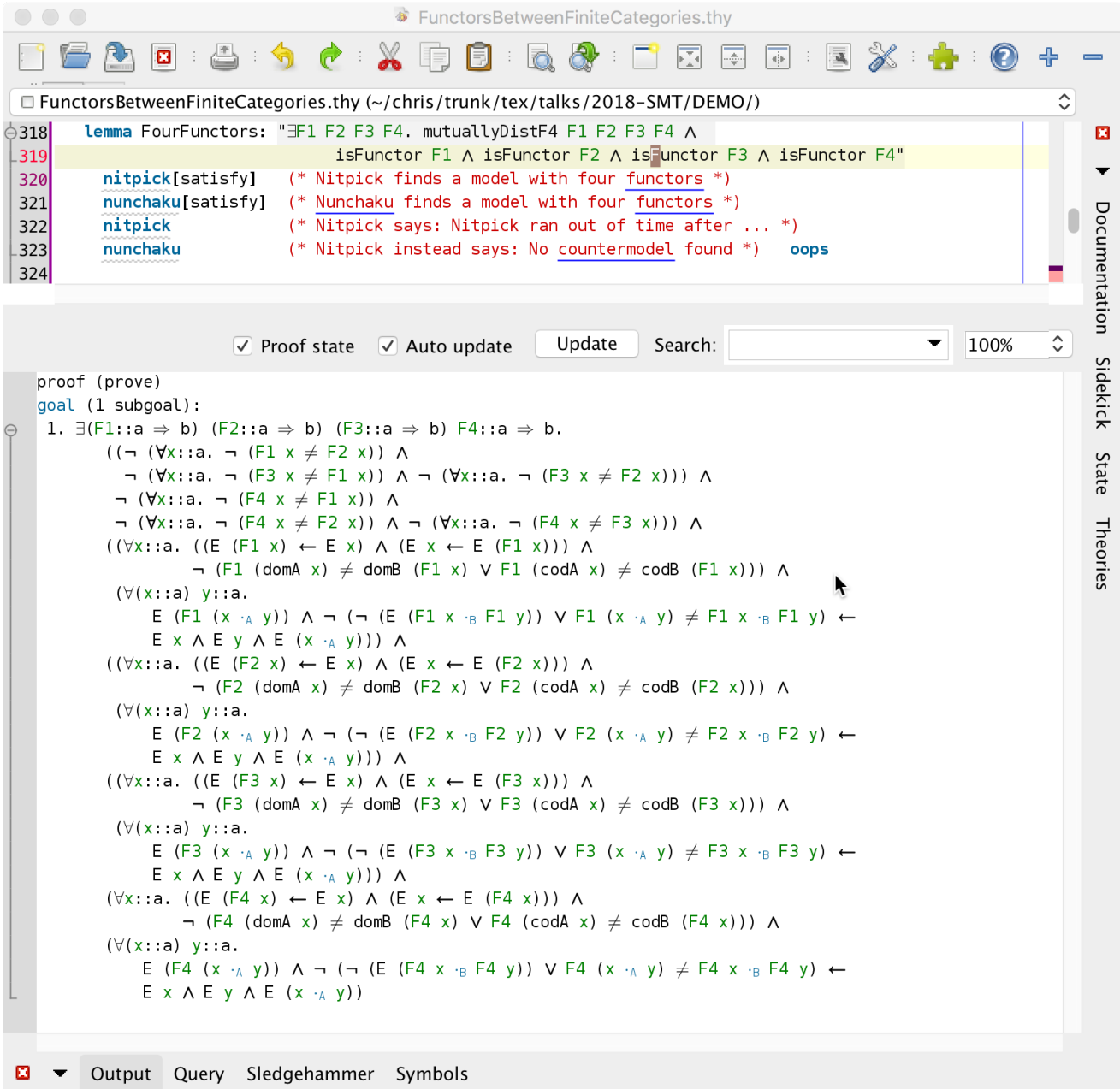


Discussion: Role of SMT solvers and ATPs

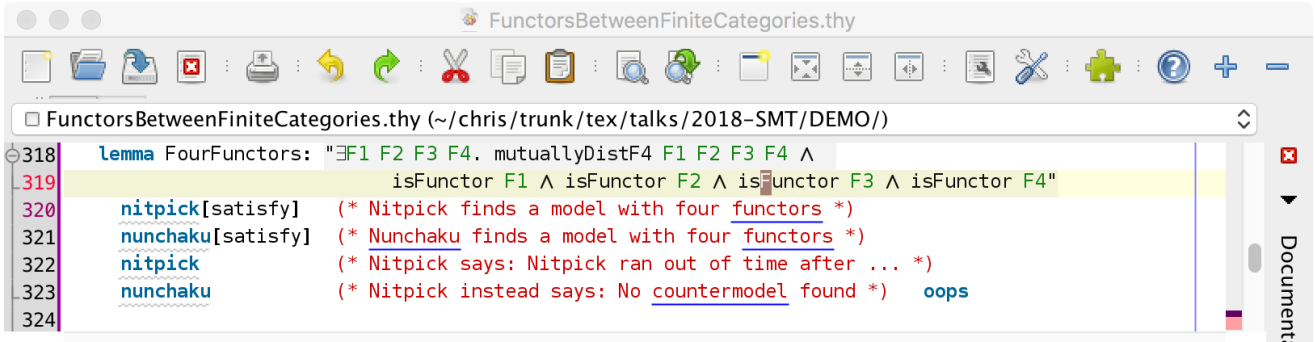


Synthesis of four functors from A2 to B2 (with Nunchaku/CVC4)

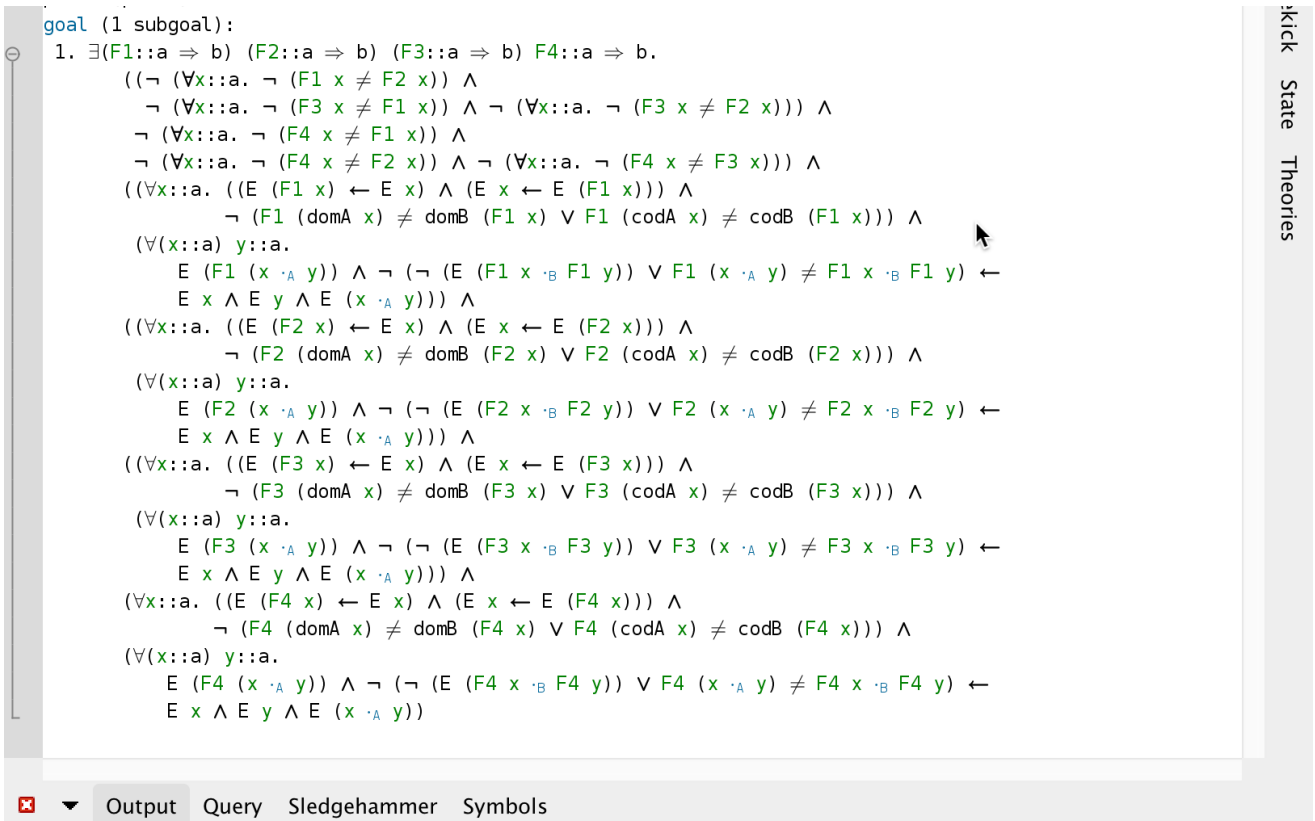
Discussion: Role of SMT solvers and ATPs



Discussion: Role of SMT solvers and ATPs



Unfolding in HOL – Intuition vanishes



Discussion: Role of SMT solvers and ATPs

media — -bash — 202×64

```
leopard:media cbenzmueller$ more ~/.isabelle/Isabelle2017/nunchaku.nun
# PATH="$CVC4_HOME:$KODKODI/bin:$SMBC_HOME:$PATH" "$NUNCHAKU_HOME"/nunchaku --skolems-in-model --no-color --solvers "cvc4,kodkod,paradox,smbc" --timeout 30 --kodkod-min-bound 4 --kodkod-bound-increment
4 /Users/cbenzmueller/.isabelle/Isabelle2017/nunchaku.nun
# This file was generated by Isabelle (most likely Nunchaku)
# 2018-07-09 11:50:35.761

# complete
# sound
val a_ : type.
val b_ : type.

val a1_ : a_.
val a2_ : a_.
val b1_ : b_.
val b2_ : b_.
val codA_ : a_ -> a_.
val codB_ : b_ -> b_.
val domA_ : a_ -> a_.
val domB_ : b_ -> b_.
val compA_ : a_ -> a_ -> a_.
val compB_ : b_ -> b_ -> b_.
val starA_ : a_.
val starB_ : b_.
val E_ : a_ -> prop.

rec isCategory_ : (a_ -> a_) -> (a_ -> a_) -> (a_ -> a_ -> a_) -> prop :=
  forall (dom_ : a_ -> a_) (cod_ : a_ -> a_) (comp_ : a_ -> a_ -> a_). isCategory_ dom_ cod_ comp_ = (((forall (x_ : a_). E_ (dom_ x_) => E_ x_) && (forall (y_ : a_). E_ (cod_ y_) => E_ y_) && (forall (
x_ : a_) (y_ : a_). ~ ((~ (~ (E_ (comp_ x_ y_) => (~ ((~ (~ (E_ (dom_ x_)))) => (~ (~ ((~ (~ (E_ (cod_ y_)))) => (~ (dom_ x_ = cod_ y_)))))))))) => (~ ((~ ((~ (~ (E_ (dom_ x_)))) => (~ (~ ((~ (~ (E_ (co
d_ y_)))) => (~ (dom_ x_ = cod_ y_))))))) => E_ (comp_ x_ y_)))))) && (forall (x_ : a_) (y_ : a_) (z_ : a_). ((~ (E_ (comp_ x_ (comp_ y_ z_)))) => E_ (comp_ (comp_ x_ y_) z_)) => (comp_ x_ (comp_ y_ z_)
 = comp_ (comp_ x_ y_) z_)) && (forall (y_ : a_). ((~ (E_ (comp_ (cod_ y_) y_))) => E_ y_) => (comp_ (cod_ y_) y_ = y_)) && (forall (x_ : a_). ((~ (E_ (comp_ x_ (dom_ x_)))) => E_ x_) => (comp_ x_ (dom_
 x_) = x_))).

val E1_ : b_ -> prop.

rec A2B2__axioms_ : a_ -> a_ -> b_ -> b_ -> prop :=
  forall (a11_ : a_) (a21_ : a_) (b11_ : b_) (b21_ : b_). A2B2__axioms_ a11_ a21_ b11_ b21_ = (((~ ((~ (E_ a11_))) => (~ (~ ((~ (~ (E_ a21_))) => (~ (forall (x_ : a_). (E_ x_ && (~ (x_ = a11_))) => (
x_ = a21_)))))))) && (~ ((~ (~ (E1_ b11_))) => (~ (~ ((~ (~ (E1_ b21_))) => (~ (forall (x_ : b_). (E1_ x_ && (~ (x_ = b11_))) => (x_ = b21_)))))))) && (~ (a11_ = a21_)) && (~ (b11_ = b21_))).
rec isCategory1_ : (b_ -> b_) -> (b_ -> b_) -> (b_ -> b_ -> b_) -> prop :=
  forall (dom_ : b_ -> b_) (cod_ : b_ -> b_) (comp_ : b_ -> b_ -> b_). isCategory1_ dom_ cod_ comp_ = (((forall (x_ : b_). E1_ (dom_ x_) => E1_ x_) && (forall (y_ : b_). E1_ (cod_ y_) => E1_ y_) && (for
all (x_ : b_) (y_ : b_). ~ ((~ (~ (E1_ (comp_ x_ y_) => (~ ((~ (~ (E1_ (dom_ x_)))) => (~ (~ ((~ (~ (E1_ (cod_ y_)))) => (~ (dom_ x_ = cod_ y_))))))))) => (~ ((~ ((~ (~ (E1_ (dom_ x_)))) => (~ (~ ((~
 ~ (E1_ (cod_ y_)))) => (~ (dom_ x_ = cod_ y_))))))) => E1_ (comp_ x_ y_)))))) && (forall (x_ : b_) (y_ : b_) (z_ : b_). ((~ (E1_ (comp_ x_ (comp_ y_ z_)))) => E1_ (comp_ (comp_ x_ y_) z_)) => (comp_ x_
(comp_ y_ z_) = comp_ (comp_ x_ y_) z_)) && (forall (y_ : b_). ((~ (E1_ (comp_ (cod_ y_) y_))) => E1_ y_) => (comp_ (cod_ y_) y_ = y_)) && (forall (x_ : b_). ((~ (E1_ (comp_ x_ (dom_ x_)))) => E1_ x_) =
> (comp_ x_ (dom_ x_) = x_))).
rec A2B2_ : a_ -> a_ -> b_ -> b_ -> prop :=
  forall (a11_ : a_) (a21_ : a_) (b11_ : b_) (b21_ : b_). A2B2_ a11_ a21_ b11_ b21_ = (isCategory_ domA_ codA_ compA_ && isCategory1_ domB_ codB_ compB_ && A2B2__axioms_ a11_ a21_ b11_ b21_).

axiom A2B2_ a1_ a2_ b1_ b2_.
axiom (~ (E1_ starB_)) && (forall (y_ : b_). (~ (E1_ y_)) => (y_ = starB_)).
axiom (~ (E_ starA_)) && (forall (y_ : a_). (~ (E_ y_)) => (y_ = starA_)).

goal exists (F1_ : a_ -> b_) (F2_ : a_ -> b_) (F3_ : a_ -> b_) (F4_ : a_ -> b_). ~ ((~ (~ (~ ((~ (~ ((~ (~ (~ (forall (x_ : a_). E_ x_ => (~ (~ (F1_ x_ = F2_ x_)))))))) => (~ (~ ((~ (~ (forall (x_
 : a_). E_ x_ => (~ (~ (F3_ x_ = F1_ x_)))))))) => (~ (~ (forall (x_ : a_). E_ x_ => (~ (~ (F3_ x_ = F2_ x_))))))))))))) => (~ (~ ((~ (~ (forall (x_ : a_). E_ x_ => (~ (~ (F4_ x_ = F1_ x_)))))))) => (~
 (~ ((~ (~ (~ (forall (x_ : a_). E_ x_ => (~ (~ (F4_ x_ = F2_ x_)))))))) => (~ (~ (forall (x_ : a_). E_ x_ => (~ (~ (F4_ x_ = F3_ x_))))))))))))) => (~ (~ ((~ (~ ((~ (~ (forall (x_ : a_). (~ ((~ (
~ ((~ (~ (E_ x_ => E1_ (F1_ x_)))) => (~ (E1_ (F1_ x_) => E_ x_)))))) => (~ (~ ((~ (~ (F1_ (domA_ x_) = domB_ (F1_ x_)))) => (~ (F1_ (codA_ x_) = codB_ (F1_ x_))))))))) => (~ (forall (x_ : a_) (y_ : a_
). (~ ((~ (~ (E_ x_))) => (~ (~ ((~ (~ (E_ y_))) => (~ (E_ (compA_ x_ y_)))))))) => (~ ((~ (~ (E1_ (F1_ (compA_ x_ y_))))) => (~ (~ ((~ (~ (E1_ (compB_ (F1_ x_) (F1_ y_))))) => (~ (F1_ (compA_ x_ y_) =
compB_ (F1_ x_) (F1_ y_)))))))))))) => (~ (~ ((~ (~ ((~ (~ (forall (x_ : a_). ~ ((~ (~ (~ ((~ (~ (E_ x_ => E1_ (F2_ x_)))) => (~ (E1_ (F2_ x_) => E_ x_))))))) => (~ (~ ((~ (~ (F2_ (domA_ x_) = domB_
 (F2_ x_)))) => (~ (F2_ (codA_ x_) = codB_ (F2_ x_))))))))))) => (~ (forall (x_ : a_) (y_ : a_). (~ ((~ (~ (E_ x_))) => (~ (~ ((~ (~ (E_ y_))) => (~ (E_ (compA_ x_ y_)))))))) => (~ ((~ (~ (E1_ (F2_ (comp
A_ x_ y_))))) => (~ (~ ((~ (~ (E1_ (compB_ (F2_ x_) (F2_ y_))))) => (~ (F2_ (compA_ x_ y_) = compB_ (F2_ x_) (F2_ y_)))))))))))))) => (~ (~ ((~ (~ ((~ (~ (forall (x_ : a_). ~ ((~ (~ ((~ (~ (E_ x_
 => E1_ (F3_ x_)))) => (~ (E1_ (F3_ x_) => E_ x_))))))) => (~ (~ ((~ (~ (F3_ (domA_ x_) = domB_ (F3_ x_)))) => (~ (F3_ (codA_ x_) = codB_ (F3_ x_))))))))))) => (~ (forall (x_ : a_) (y_ : a_). (~ ((~ (~ (E_
 x_))) => (~ (~ ((~ (~ (E_ y_))) => (~ (E_ (compA_ x_ y_)))))))) => (~ ((~ (~ (E1_ (F3_ (compA_ x_ y_))))) => (~ (~ ((~ (~ (E1_ (compB_ (F3_ x_) (F3_ y_))))) => (~ (F3_ (compA_ x_ y_) = compB_ (F3_ x_)
(F3_ y_)))))))))))))) => (~ (~ ((~ (~ (forall (x_ : a_). ~ ((~ (~ (~ ((~ (~ (E_ x_ => E1_ (F4_ x_)))) => (~ (E1_ (F4_ x_) => E_ x_))))))) => (~ (~ ((~ (~ (F4_ (domA_ x_) = domB_ (F4_ x_)))) => (~ (F4_ (c
odA_ x_) = codB_ (F4_ x_)))))))))) => (~ (forall (x_ : a_) (y_ : a_). (~ ((~ (~ (E_ x_))) => (~ (~ ((~ (~ (E_ y_))) => (~ (E_ (compA_ x_ y_)))))))) => (~ ((~ (~ (E1_ (F4_ (compA_ x_ y_))))) => (~ (~ ((~
 (~ (E1_ (compB_ (F4_ x_) (F4_ y_))))) => (~ (F4_ (compA_ x_ y_) = compB_ (F4_ x_) (F4_ y_)))))))))))))))))))))))).
leopard:media cbenzmueller$
```

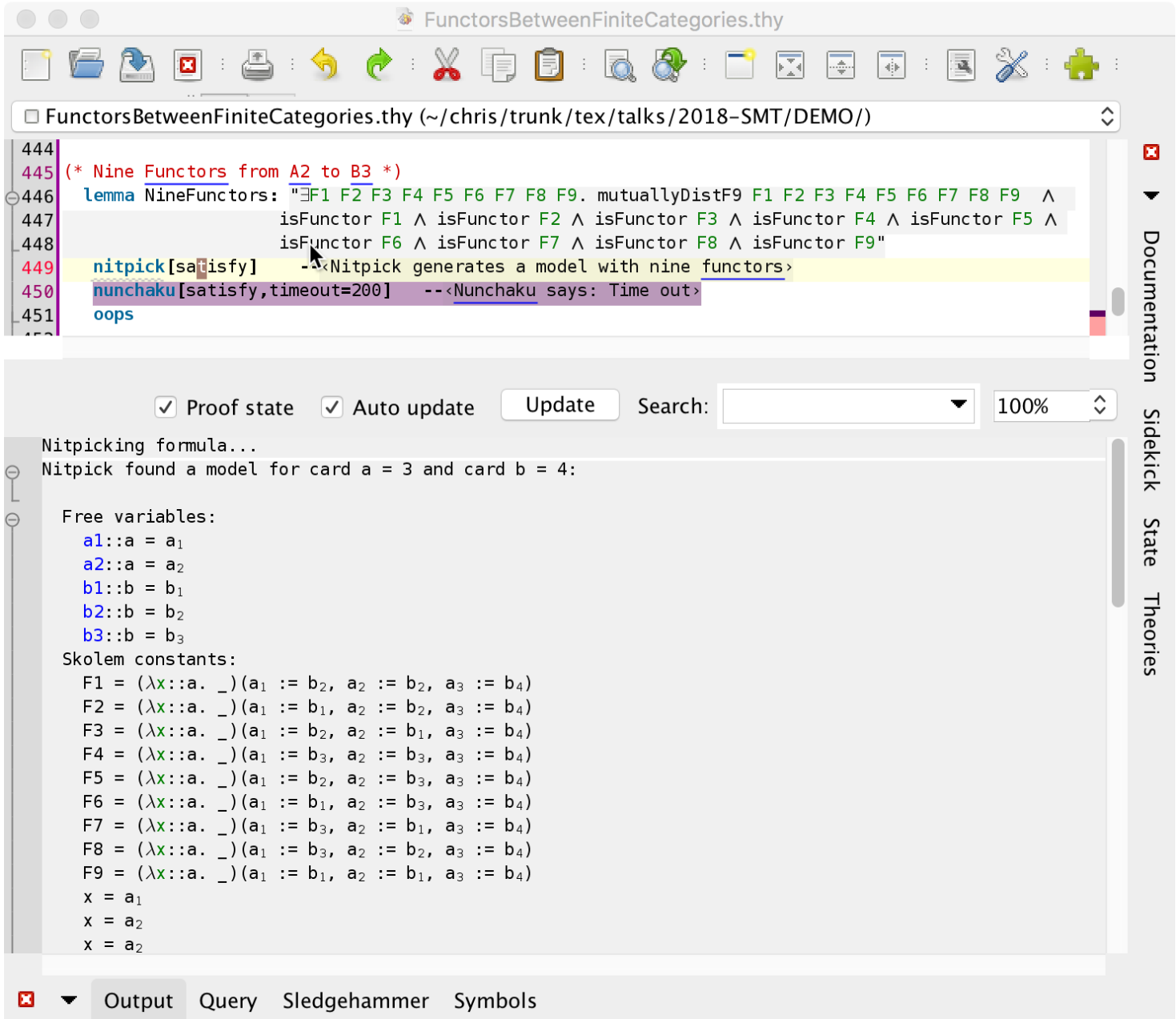
Discussion: Role of SMT solvers and ATPs

media — -bash — 202×64

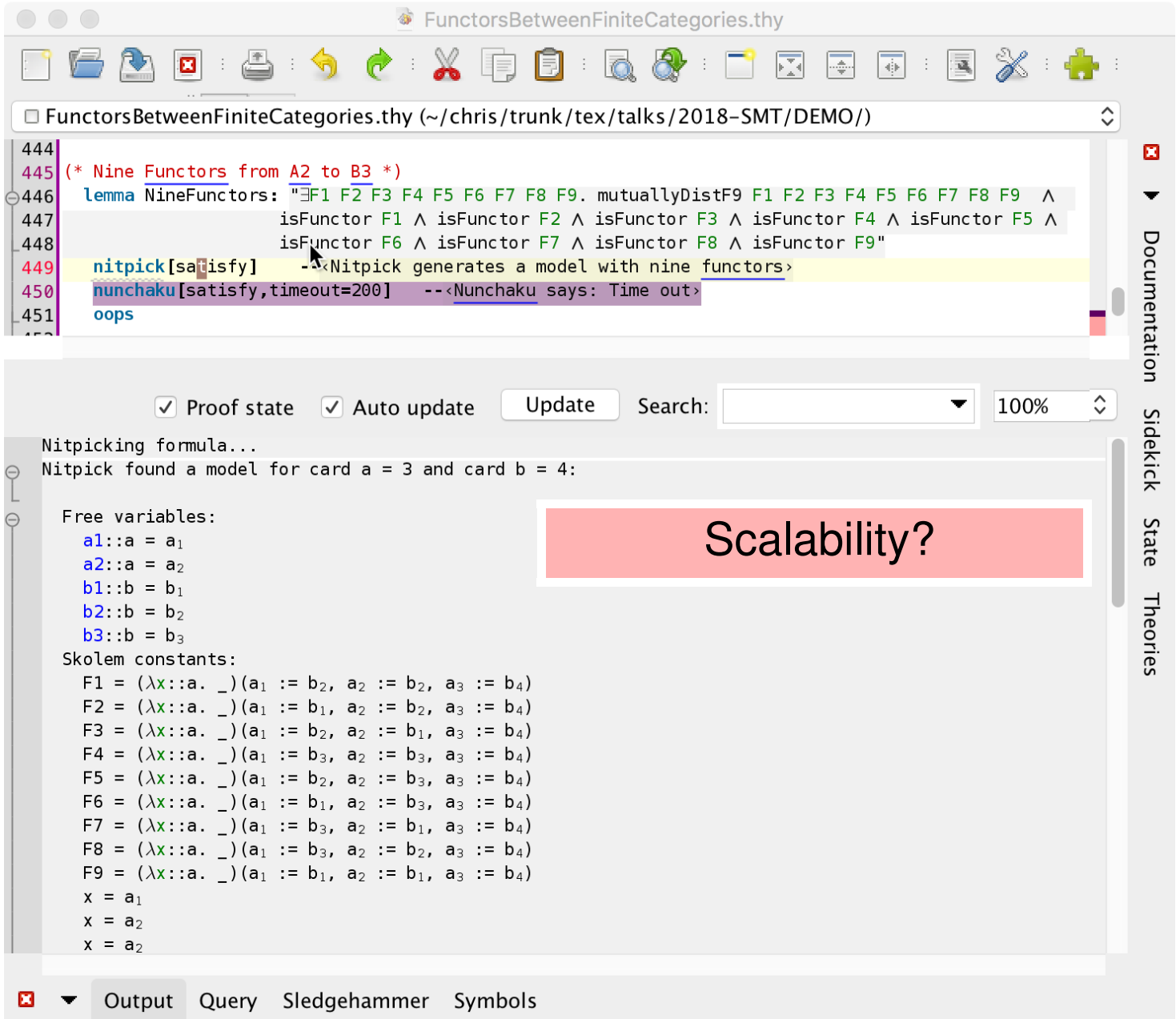
```
leopard:media cbenzmueller$ more ~/.isabelle/Isabelle2017/nunchaku.nun
# PATH="$CVC4_HOME:$KODKODI/bin:$SMBC_HOME:$PATH" "$NUNCHAKU_HOME"/nunchaku --skolems-in-model --no-color --solvers "cvc4,kodkod,paradox,smbc" --timeout 30 --kodkod-min-bound 4 --kodkod-bound-increment
4 /Users/cbenzmueller/.isabelle/Isabelle2017/nunchaku.nun
# This file was generated by Isabelle (most likely Nunchaku)
# 2018-07-09 11:50:35.761

# complete
# sound
val a_ : type.
val b_ : type.

val a1_ : a_.
val a2_ : a_.
val b1_ : b_.
val b2_ : b_.
val codA_ : a_ -> a_.
val codB_ : b_ -> b_.
val domA_ : a_ -> a_.
val domB_ : b_ -> b_.
val compA_ : a_ -> a_ -> a_.
val compB_ : b_ -> b_ -> b_.
val starA_ : a_.
val starB_ : b_.
val E_ : a_ -> prop.

rec isCategory_ : (a_ -> a_) -> (a_ -> a_) -> (a_ -> a_ -> a_) -> prop :=
  forall (dom_ : a_ -> a_) (cod_ : a_ -> a_) (comp_ : a_ -> a_ -> a_). isCategory_ dom_ cod_ comp_ = (((forall (x_ : a_). E_ (dom_ x_) => E_ x_) && (forall (y_ : a_). E_ (cod_ y_) => E_ y_) && (forall (
x_ : a_) (y_ : a_). ~ ((~ (~ (E_ (comp_ x_ y_) => (~ ((~ (~ (E_ (dom_ x_)))) => (~ (~ ((~ (~ (E_ (cod_ y_)))) => (~ (dom_ x_ = cod_ y_)))))))))) => (~ ((~ ((~ (~ (E_ (dom_ x_)))) => (~ (~ ((~ (~ (E_ (co
d_ y_)))) => (~ (dom_ x_ = cod_ y_))))))) => E_ (comp_ x_ y_)))))) && (forall (x_ : a_) (y_ : a_) (z_ : a_). ((~ (E_ (comp_ x_ (comp_ y_ z_)))) => E_ (comp_ (comp_ x_ y_) z_)) => (comp_ x_ (comp_ y_ z_)
 = comp_ (comp_ x_ y_) z_)) && (forall (y_ : a_). ((~ (E_ (comp_ (cod_ y_) y_))) => E_ y_) => (comp_ (cod_ y_) y_ = y_)) && (forall (x_ : a_). ((~ (E_ (comp_ x_ (dom_ x_)))) => E_ x_) => (comp_ x_ (dom_
 x_) = x_))).

val E1_ : b_ -> prop.

rec A2B2__axioms_ : a_ -> a_ -> b_ -> b_ -> prop :=
  forall (a11_ : a_) (a21_ : a_) (b11_ : b_) (b21_ : b_). A2B2__axioms_ a11_ a21_ b11_ b21_ = (((~ ((~ (~ (E_ a11_))) => (~ (~ ((~ (~ (E_ a21_))) => (~ (forall (x_ : a_). (E_ x_ && (~ (x_ = a11_))) => (
x_ = a21_)))))))) && (~ ((~ (~ (E1_ b11_))) => (~ (~ ((~ (~ (E1_ b21_))) => (~ (forall (x_ : b_). (E1_ x_ && (~ (x_ = b11_))) => (x_ = b21_)))))))) && (~ (a11_ = a21_)) && (~ (b11_ = b21_))).
rec isCategory1_ : (b_ -> b_) -> (b_ -> b_) -> (b_ -> b_ -> b_) -> prop :=
  forall (dom_ : b_ -> b_) (cod_ : b_ -> b_) (comp_ : b_ -> b_ -> b_). isCategory1_ dom_ cod_ comp_ = (((forall (x_ : b_). E1_ (dom_ x_) => E1_ x_) && (forall (y_ : b_). E1_ (cod_ y_) => E1_ y_) && (for
all (x_ : b_) (y_ : b_). ~ ((~ (~ (E1_ (comp_ x_ y_) => (~ ((~ (~ (E1_ (dom_ x_)))) => (~ (~ ((~ (~ (E1_ (cod_ y_)))) => (~ (dom_ x_ = cod_ y_)))))))))) => (~ ((~ ((~ (~ (E1_ (dom_ x_)))) => (~ (~ ((~ (
~ (E1_ (cod_ y_)))) => (~ (dom_ x_ = cod_ y_))))))) => E1_ (comp_ x_ y_)))))) && (forall (x_ : b_) (y_ : b_) (z_ : b_). ((~ (E1_ (comp_ x_ (comp_ y_ z_)))) => E1_ (comp_ (comp_ x_ y_) z_)) => (comp_ x_
(comp_ y_ z_) = comp_ (comp_ x_ y_) z_)) && (forall (y_ : b_). ((~ (E1_ (comp_ (cod_ y_) y_))) => E1_ y_) => (comp_ (cod_ y_) y_ = y_)) && (forall (x_ : b_). ((~ (E1_ (comp_ x_ (dom_ x_)))) => E1_ x_) =
> (comp_ x_ (dom_ x_) = x_))).
rec A2B2_ : a_ -> a_ -> b_ -> b_ -> prop :=
  forall (a11_ : a_) (a21_ : a_) (b11_ : b_) (b21_ : b_). A2B2_ a11_ a21_ b11_ b21_ = (isCategory_ domA_ codA_ compA_ && isCategory1_ domB_ codB_ compB_ && A2B2__axioms_ a11_ a21_ b11_ b21_).

axiom A2B2_ a1_ a2_ b1_ b2_.
axiom (~ (E1_ starB_)) && (forall (y_ : b_). (~ (E1_ y_)) => (y_ = starB_)).
axiom (~ (E_ starA_)) && (forall (y_ : a_). (~ (E_ y_)) => (y_ = starA_)).

goal exists (F1_ : a_ -> b_) (F2_ : a_ -> b_) (F3_ : a_ -> b_) (F4_ : a_ -> b_). ~ ((~ (~ (~ ((~ (~ (~ ((~ (~ (~ (forall (x_ : a_). E_ x_ => (~ (~ (F1_ x_ = F2_ x_)))))))) => (~ (~ ((~ (~ (forall (x_
: a_). E_ x_ => (~ (~ (F3_ x_ = F1_ x_)))))))) => (~ (~ (forall (x_ : a_). E_ x_ => (~ (~ (F3_ x_ = F2_ x_))))))))))))) => (~ (~ ((~ (~ (~ (forall (x_ : a_). E_ x_ => (~ (~ ((~ (~ (F4_ x_ = F1_ x_)))))))))))) => (~
(~ ((~ (~ (~ (forall (x_ : a_). E_ x_ => (~ (~ (F4_ x_ = F2_ x_)))))))) => (~ (~ (forall (x_ : a_). E_ x_ => (~ (~ (F4_ x_ = F3_ x_))))))))))))))) => (~ (~ ((~ (~ ((~ (~ (forall (x_ : a_). ~ ((~ (~
~ ((~ (~ (E_ x_ => E1_ (F1_ x_)))) => (~ (E1_ (F1_ x_) => E_ x_)))))) => (~ (~ ((~ (~ (F1_ (domA_ x_) = domB_ (F1_ x_)))) => (~ (F1_ (codA_ x_) = codB_ (F1_ x_))))))))) => (~ (forall (x_ : a_) (y_ : a_
). (~ ((~ (~ (E_ x_))) => (~ (~ ((~ (~ (E_ y_))) => (~ (E_ (compA_ x_ y_)))))))) => (~ ((~ (~ (E1_ (F1_ (compA_ x_ y_))))) => (~ (~ ((~ (~ (E1_ (compB_ (F1_ x_) (F1_ y_))))) => (~ (F1_ (compA_ x_ y_) =
compB_ (F1_ x_) (F1_ y_)))))))))))) => (~ (~ ((~ (~ ((~ (~ (forall (x_ : a_). ~ ((~ (~ (~ ((~ (~ (E_ x_ => E1_ (F2_ x_)))) => (~ (E1_ (F2_ x_) => E_ x_)))))) => (~ (~ ((~ (~ (F2_ (domA_ x_) = domB_
 (F2_ x_)))) => (~ (F2_ (codA_ x_) = codB_ (F2_ x_))))))))))) => (~ (forall (x_ : a_) (y_ : a_). (~ ((~ (~ (E_ x_))) => (~ (~ ((~ (~ (E_ y_))) => (~ (E_ (compA_ x_ y_)))))))) => (~ ((~ (~ (E1_ (F2_ (comp
A_ x_ y_))))) => (~ (~ ((~ (~ (E1_ (compB_ (F2_ x_) (F2_ y_))))) => (~ (F2_ (compA_ x_ y_) = compB_ (F2_ x_) (F2_ y_))))))))))))))) => (~ (~ ((~ (~ ((~ (~ (forall (x_ : a_). ~ ((~ (~ (~ ((~ (~ (E_ x_
=> E1_ (F3_ x_)))) => (~ (E1_ (F3_ x_) => E_ x_)))))) => (~ (~ ((~ (~ (F3_ (domA_ x_) = domB_ (F3_ x_)))) => (~ (F3_ (codA_ x_) = codB_ (F3_ x_))))))))))) => (~ (forall (x_ : a_) (y_ : a_). (~ ((~ (~ (E_
 x_))) => (~ (~ ((~ (~ (E_ y_))) => (~ (E_ (compA_ x_ y_)))))))) => (~ ((~ (~ (E1_ (F3_ (compA_ x_ y_))))) => (~ (~ ((~ (~ (E1_ (compB_ (F3_ x_) (F3_ y_))))) => (~ (F3_ (compA_ x_ y_) = compB_ (F3_ x_)
(F3_ y_))))))))))))))) => (~ (~ ((~ (~ (forall (x_ : a_). ~ ((~ (~ (~ ((~ (~ (E_ x_ => E1_ (F4_ x_)))) => (~ (E1_ (F4_ x_) => E_ x_)))))) => (~ (~ ((~ (~ (F4_ (domA_ x_) = domB_ (F4_ x_)))) => (~ (F4_ (c
odA_ x_) = codB_ (F4_ x_))))))))))) => (~ (forall (x_ : a_) (y_ : a_). (~ ((~ (~ (E_ x_))) => (~ (~ ((~ (~ (E_ y_))) => (~ (E_ (compA_ x_ y_)))))))) => (~ ((~ (~ (E1_ (F4_ (compA_ x_ y_))))) => (~ (~ ((~
 (~ (E1_ (compB_ (F4_ x_) (F4_ y_))))) => (~ (F4_ (compA_ x_ y_) = compB_ (F4_ x_) (F4_ y_)))))))))))))))))))))))).
leopard:media cbenzmueller$
```

Intuitive user-interaction?
Probably not at this level of representation!

Discussion: Role of SMT solvers and ATPs



Discussion: Role of SMT solvers and ATPs



Pipeline from Nunchaku to CVC4

Pipeline components

type inference

infer implicit type parameters, and check that the problem is well-typed

skolemization

replace existential variables in the goal by fresh constant/function symbols

monomorphization

specialize polymorphic functions and types, and perform dependency analysis to keep only the relevant types and definitions

elim matching

encode pattern matching into `DataSelect` / `DataTest` constructs that can be used by CVC4

recursion elimination

encoding of recursive functions that are more suitable for CVC4's finite model finding.

conversion to FO

check that the problem is now monomorphic first-order, and translate it to a specialized format

elim multiple equations

(**wip**) convert Haskell-like definitions, with multiple definitions for a single function, into a single-equation + pattern matching (à la Coq).

Pipeline from Nunchaku to CVC4

specialization

(**todo**) specialize higher-order functions (e.g., `map` or `fold`) to their functional arguments, to obtain a first-order function. For instance, `map (fun x → x+1)` should become `map_42 : list nat → list nat` where `map_42 l = match l with nil → nil | cons x l → cons (x+1) (map_42 l) end`

inlining/unfolding

(**todo**) related to specialization, this replaces some non-recursive functions by their definition

encoding HO functions to arrays

(**todo**) to have CVC4 synthesize some function `τ1 → τ2` passed as argument to another function, where `τ1` is a finite type, encode the function as an `array τ1 τ2`.

encoding dependent types

(**todo**) once we have dependent types, remove term arguments to a type and instead use some predicate to enforce typing invariant. For instance, `vec : nat → type → type` becomes `vec_ : type → type`, and `forall v:(vec n τ). p[v]` becomes `forall v:(vec τ). well-formed-vec n v ⇒ p[v]` where `well-formed-vec n v` is an inductive predicate basically enforcing that `n = length v`.

Backends:

CVC4

send the first-order problem to CVC4 in SMTlib syntax, and read a model back if it indicates `SAT`.

Narrowing

(**wip**) given a problem where every symbol is computable (defined as a function rather than uninterpreted symbol + axioms), use symbolic evaluation + refinement of variables of inductive type to obtain counter-example values.

Some References

Category Theory and Free Logic in HOL (jww Dana Scott)

- Axiom Systems for Category Theory in Free Logic, AFP, 2018
- Some Reflections on a Computer-aided Theory Exploration Study in Category Theory (Extended Abstract), AITP 2018
- Axiomatizing Category Theory in Free Logic, CoRR, abs/1609.01493, 2016 (submitted)
- Automating Free Logic in Isabelle/HOL, ICMS, 2016

Sledgehammer (SMT cooperation)

- Extending Sledgehammer with SMT Solvers (**Jasmin C. Blanchette, Sascha Böhme, Lawrence C. Paulson**), JAR 2013
- Proving Theorems of Higher-Order Logic with SMT Solvers (**Sascha Böhme**), PhD thesis, TUM, 2012

Nunchaku (SMT cooperation)

- Model Finding for Recursive Functions in SMT(**Andrew Reynolds, Jasmin C. Blanchette, Simon Cruanes, Cesare Tinelli**), IJCAR, 2016

Nitpick (cooperation with FO relational model finder Kodkod)

- Relational Analysis of (Co)inductive Predicates, (Co)algebraic Datatypes, and (Co)recursive Functions (**Jasmin C. Blanchette**), Software Quality Journal, 2013

Conclusion

Interesting and useful exploration study in Category Theory

- Domain expert (Dana) — tool expert (myself) — proof assistant (Isabelle)

Conclusion

Interesting and useful exploration study in Category Theory

- Domain expert (Dana) — ~~tool expert (myself)~~ — proof assistant (Isabelle)

Conclusion

Interesting and useful exploration study in Category Theory

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?

Conclusion

Interesting and useful exploration study in Category Theory

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?

First implementation and automation of Free Logic

Conclusion

Interesting and useful exploration study in Category Theory

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?

First implementation and automation of Free Logic

HOL utilised as (quite) Universal Metalogic (via SSE approach):

- **Lean and elegant** approach to integrate and combine heterogeneous logics
- **Reuse** of SMT solvers and ATPs to achieve a high degree of **automation**
- **Uniform proofs** (modulo the embeddings)
- Supports **Intuitive user interaction** at abstract level, but further improvements required
- Approach very well suited for (interdisciplinary) **teaching** of logics

Conclusion

Interesting and useful exploration study in Category Theory

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?

First implementation and automation of Free Logic

HOL utilised as (quite) Universal Metalogic (via SSE approach):

- **Lean and elegant** approach to integrate and combine heterogeneous logics
- **Reuse** of SMT solvers and ATPs to achieve a high degree of **automation**
- **Uniform proofs** (modulo the embeddings)
- Supports **Intuitive user interaction** at abstract level, but further improvements required
- Approach very well suited for (interdisciplinary) **teaching** of logics

Lots of further work

- Philosophy, Maths, CS, AI, NLP, ...
- Rational Argumentation
- **Legal- and Ethical-Reasoning in Intelligent Machines**

```
lemma InconsistencyInteractive: assumes NEx: "∃x. ¬(E x)" shows False
proof -
 (* Let "a" be an undefined object. *)
 obtain a where 1: "¬(E a)" using assms by auto
 (* We instantiate axiom "A3a" with "a". *)
 have 2: "(□a)·a ≅ a" using A3a by blast
 (* By unfolding the definition of "≅" we get from 1 that "(□a)·a" is not defined. This is
    easy to see, since if "(□a)·a" were defined, we also had that "a" is defined, which is
    not the case by assumption. *)
 have 3: "¬(E((□a)·a))" using 1 2 by metis
 (* We instantiate axiom "A1" with "□a" and "a". *)
 have 4: "E((□a)·a) ↔ (□a)□ ≅ □a" using A1 by blast
 (* We instantiate axiom "A2a" with "a". *)
 have 5: "(□a)□ ≅ □a" using A2a by blast
 (* From 4 and 5 we obtain "(E((□a)·a))" by propositional logic. *)
 have 6: "E((□a)·a)" using 4 5 by blast
 (* We have "¬(E((□a)·a))" and "E((□a)·a)", hence Falsity. *)
 then show ?thesis using 6 3 by blast
qed
```

```
lemma InconsistencyInteractive: assumes NEx: "∃x. ¬(E x)" shows False
proof -
 (* Let "a" be an undefined object. *)
 obtain a where 1: "¬(E a)" using assms by auto
 (* We instantiate axiom "A3a" with "a". *)
 have 2: "(□a)·a ≅ a" using A3a by blast
 (* By unfolding the definition of "≅" we get from 1 that "(□a)·a" is not defined. This is
    easy to see, since if "(□a)·a" were defined, we also had that "a" is defined, which is
    not the case by assumption. *)
 have 3: "¬(E((□a)·a))" using 1 2 by metis
 (* We instantiate axiom "A1" with "□a" and "a". *)
 have 4: "E((□a)·a) ↔ (□a)□ ≅ □a" using A1 b
 (* We instantiate axiom "A2a" with "a". *)
 have 5: "(□a)□ ≅ □a" using A2a by blast
 (* From 4 and 5 we obtain "(E((□a)·a))" by pr
 have 6: "E((□a)·a)" using 4 5 by blast
 (* We have "¬(E((□a)·a))" and "E((□a)·a)", he
 then show ?thesis using 6 3 by blast
qed
```

```
assumes
  A1: "E(x·y) ↔ (x□ ≅ □y)" and
 A2a: "((□x)□) ≅ □x" and
 A2b: "□(x□) ≅ □x" and
 A3a: "(□x)·x ≅ x" and
 A3b: "x·(x□) ≅ x" and
 A4a: "□(x·y) ≅ □(x·(□y))" and
 A4b: "(x·y)□ ≅ ((x□)·y)□" and
  A5: "x·(y·z) ≅ (x·y)·z"
begin
```

```
lemma InconsistencyInteractiveVII:
  assumes NEx: "∃x. ¬(E x)" shows False
proof -
 (* Let "a" be an undefined object. *)
 obtain a where 1: "¬(E a)" using NEx by auto
 (* We instantiate axiom "A3a" with "a". *)
 have 2: "a·(dom a) ≅ a" using A3a by blast
 (* By unfolding the definition of "≅" we get from 1 that "a·(dom a)" is
    not defined. This is easy to see, since if "a·(dom a)" were defined, we also
    had that "a" is defined, which is not the case by assumption. *)
 have 3: "¬(E(a·(dom a)))" using 1 2 by metis
 (* We instantiate axiom "A1" with "a" and "dom a". *)
 have 4: "E(a·(dom a)) ↔ dom a ≅ cod(dom a)" using A1 by blast
 (* We instantiate axiom "A2a" with "a". *)
 have 5: "cod(dom a) ≅ dom a" using A2a by blast
 (* We use 5 (and symmetry and transitivity of "≅") to rewrite the
    right-hand of the equivalence 4 into "dom a ≅ dom a". *)
 have 6: "E(a·(dom a)) ↔ dom a ≅ dom a" using 4 5 by auto
 (* By reflexivity of "≅" we get that "a·(dom a)" must be defined. *)
 have 7: "E(a·(dom a))" using 6 by blast
 (* We have shown in 7 that "a·(dom a)" is defined, and in 3 that it is undefined.
    Contradiction. *)
 then show ?thesis using 7 3 by blast
qed
```

```
lemma InconsistencyInteractiveVII:
  assumes NEx: "∃x. ¬(E x)" shows False
proof -
 (* Let "a" be an undefined object. *)
 obtain a where 1: "¬(E a)" using NEx by auto
 (* We instantiate axiom "A3a" with "a". *)
 have 2: "a·(dom a) ≅ a" using A3a by blast
 (* By unfolding the definition of "≅" we get from 1 that "a·(dom a)" is
    not defined. This is easy to see, since if "a·(dom a)" were defined, we also
    had that "a" is defined, which is not the case by assumption. *)
 have 3: "¬(E(a·(dom a)))" using 1 2 by metis
 (* We instantiate axiom "A1" with "a" and "dom a". *)
 have 4: "E(a·(dom a)) ↔ dom a ≅ cod(dom a)" using A1 by blast
 (* We instantiate axiom "A2a" with "a". *)
 have 5: "cod(dom a) ≅ dom a" using A2a b
 (* We use 5 (and symmetry and transitivi
    right-hand of the equivalence 4 into
 have 6: "E(a·(dom a)) ↔ dom a ≅ dom a" u
 (* By reflexivity of "≅" we get that "a·
 have 7: "E(a·(dom a))" using 6 by blast
 (* We have shown in 7 that "a·(dom a)" is
    Contradiction. *)
 then show ?thesis using 7 3 by blast
qed
```

```
assumes
  A1: "E(x·y) ↔ dom x ≅ cod y" and
 A2a: "cod(dom x) ≅ dom x " and
 A2b: "dom(cod y) ≅ cod y" and
 A3a: "x·(dom x) ≅ x" and
 A3b: "(cod y)·y ≅ y" and
 A4a: "dom(x·y) ≅ dom((dom x)·y)" and
 A4b: "cod(x·y) ≅ cod(x·(cod y))" and
  A5: "x·(y·z) ≅ (x·y)·z"
begin
```

Evaluation (Sledgehammer in Standard Setting in Isabelle 2017)

	CVC4	Z3	E	Spass	Vampire	LEO-II
IDPredicates	-	-	✓	✓	✓	✓
			AxiomsSet1			
E_i Impl	-	-	-	-	-	-
E_i Impl-local	✓	✓	✓	-	✓	-
UC_i	-	-	-	-	-	-
UC_i-local	✓	e	-	-	✓	-
UD_i	-	-	-	-	-	-
UD_i-local	✓	e	-	-	✓	-
			AxiomsSet2			
E_{ii} Impl	-	-	-	-	-	-
E_{ii} Impl-local	✓	-	✓	✓	✓	-
domTotal	-	u	u	-	u	-
domTotal-local	✓	✓	✓	✓	✓	-
codTotal	-	-	u	-	u	-
codTotal-local	✓	✓	✓	✓	✓	-
			AxiomsSet2 entails AxiomsSet1			
S_i	✓	✓	✓	✓	✓	✓
S_i-local	✓	✓	✓	✓	✓	-
E_i	-	-	-	-	-	-
E_i-local	✓	✓	✓	✓	✓	-
A_i	✓	✓	✓	✓	✓	✓
A_i-local	✓	✓	✓	✓	✓	-
C_i	-	-	-	-	-	-
C_i-local	✓	✓	✓	✓	✓	-
D_i	-	-	-	-	-	-
D_i-local	✓	✓	✓	✓	✓	-
	-	-	-	-	-	-
	-	-	-	-	-	-

Evaluation (Sledgehammer in Standard Setting in Isabelle 2017)

	CVC4	Z3	E	Spass	Vampire	LEO-II
IDPredicates	--	--	✓	✓	✓	✓
			AxiomsSet1			
E_iImpl	-✓	-✓	-✓	--	-✓	--
UC_i	-✓	-e	--	--	-✓	--
UD_i	-✓	-e	--	--	-✓	--
			AxiomsSet2			
E_{ii}Impl	-✓	--	-✓	-✓	-✓	--
domTotal	-✓	u✓	u✓	-✓	u✓	--
codTotal	-✓	-✓	u✓	-✓	u✓	--
			AxiomsSet2 entails AxiomsSet1			
S_i	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
E_i	-✓	-✓	-✓	-✓	-✓	--
A_i	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
C_i	-✓	-✓	-✓	-✓	-✓	--
D_i	-✓	-✓	-✓	-✓	-✓	--
			AxiomsSet3			
E_{iii}Impl	-✓	-✓	-✓	-✓	-✓	--
			AxiomsSet3 entails AxiomsSet2			
S_{ii}	✓✓	✓✓	✓✓	✓✓	✓✓	--
E_{ii}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A_{ii}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
C_{ii}	-✓	-✓	-✓	-✓	-✓	--
D_{ii}	-✓	-✓	-✓	-✓	-✓	--
			AxiomsSet2 entails AxiomsSet3			
S_{iii}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
E_{iii}	-✓	-✓	-✓	-✓	-✓	--
A_{iii}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
C_{iii}	-✓	-✓	-✓	-✓	-✓	--
D_{iii}	-✓	-✓	-✓	-✓	-✓	--

	CVC4	Z3	E	Spass	Vampire	LEO-II
			AxiomsSet4 entails AxiomsSet3			
S_{iii}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
E_{iii}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A_{iii}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
C_{iii}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
D_{iii}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
			AxiomsSet3 entails AxiomsSet4			
S_{iv}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
E_{iv}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A_{iv}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
C_{iv}	-✓	-✓	-✓	-✓	-✓	--
D_{iv}	-✓	-✓	-✓	-✓	-✓	--
			AxiomsSet5 entails AxiomsSet4			
S_{iv}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
E_{iv}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A_{iv}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
C_{iv}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
D_{iv}	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
			AxiomsSet4 entails AxiomsSet5			
S1	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
S2	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
S3	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
S4	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
S5	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
S6	✓✓	✓✓	✓✓	✓✓	✓✓	✓-

Evaluation (Sledgehammer in Standard Setting in Isabelle 2017)

	CVC4	Z3	E	Spass	Vampire	LEO-II
			AxiomsSet6 entails AxiomsSet5			
S1	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
S2	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
S3	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
S4	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
S5	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
S6	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A4aRedundant	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A4bRedundant	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A2aRedundant	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A2bRedundant	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
			AxiomsSet5 entails AxiomsSet6			
A1	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A2a	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A2b	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A3a	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A3b	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A4a	✓✓	✓✓	✓✓	✓✓	✓✓	--
A4b	✓✓	✓✓	✓✓	✓✓	✓✓	--
A5	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
			AxiomsSet7			
IncInt1	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓
IncInt2	✓✓	✓✓	✓✓	✓✓	✓✓	--
IncInt3	✓✓	✓-	✓✓	✓✓	✓✓	--
IncInt4	✓✓	✓✓	✓✓	✓✓	✓✓	--
IncInt5	✓✓	✓✓	✓✓	✓✓	✓✓	--
IncInt6	✓✓	✓✓	✓✓	✓✓	✓✓	--
IncInt7	✓✓	✓✓	✓✓	✓✓	✓✓	--
IncInt8	✓✓	✓-	✓✓	✓✓	✓✓	--

	CVC4	Z3	E	Spass	Vampire	LEO-II
			AxiomsSet7orig			
IncInt1	✓✓	✓✓	✓✓	✓✓	✓✓	--
IncInt2	✓✓	✓✓	✓✓	✓✓	✓✓	--
IncInt3	✓-	✓-	✓✓	✓-	✓✓	--
IncInt4	✓✓	✓✓	✓✓	✓✓	✓✓	--
IncInt5	✓✓	✓✓	✓✓	✓✓	✓✓	--
IncInt6	✓✓	✓✓	✓✓	✓✓	✓✓	--
IncInt7	✓-	✓-	✓✓	✓-	✓✓	--
			AxiomsSet8strict entails AxiomsSet5			
S1	✓✓	✓✓	✓✓	✓✓	✓✓	--
S2	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
S3	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
S4	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
S5	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
S6	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
			AxiomsSet5 entails AxiomsSet6			
A1	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A2a	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A2b	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A3a	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A3b	✓✓	✓✓	✓✓	✓✓	✓✓	✓-
A4a	✓✓	✓✓	✓✓	✓✓	✓✓	--
A4b	✓✓	✓✓	✓✓	✓✓	✓✓	--
A5	✓✓	✓✓	✓✓	✓✓	✓✓	✓-

Some Reflections

- Domain expert (Dana) — tool expert (myself) — proof assistant (Isabelle)

Some Reflections

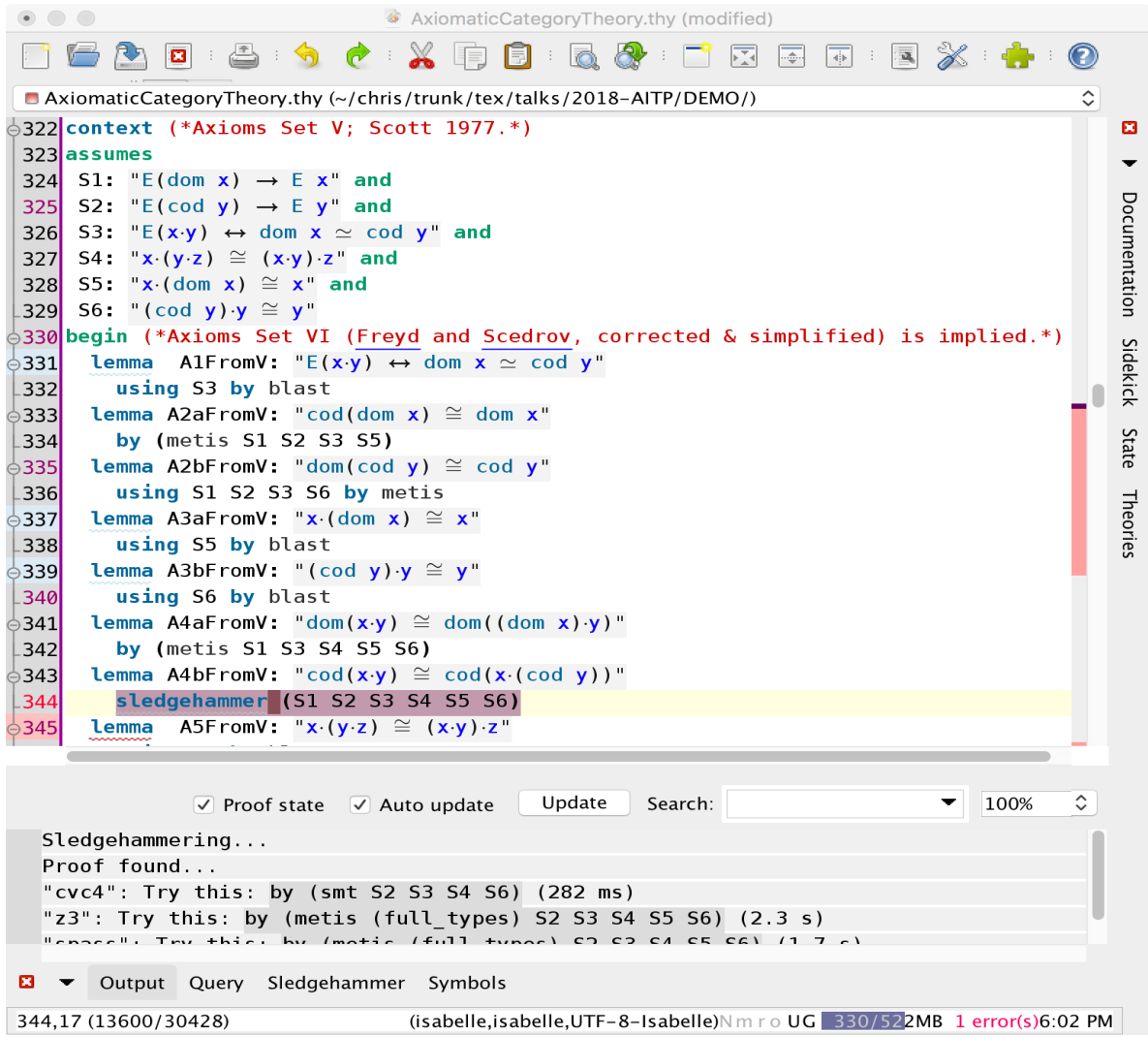
- Domain expert (Dana) — ~~tool expert (myself)~~ — proof assistant (Isabelle)

Some Reflections

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?

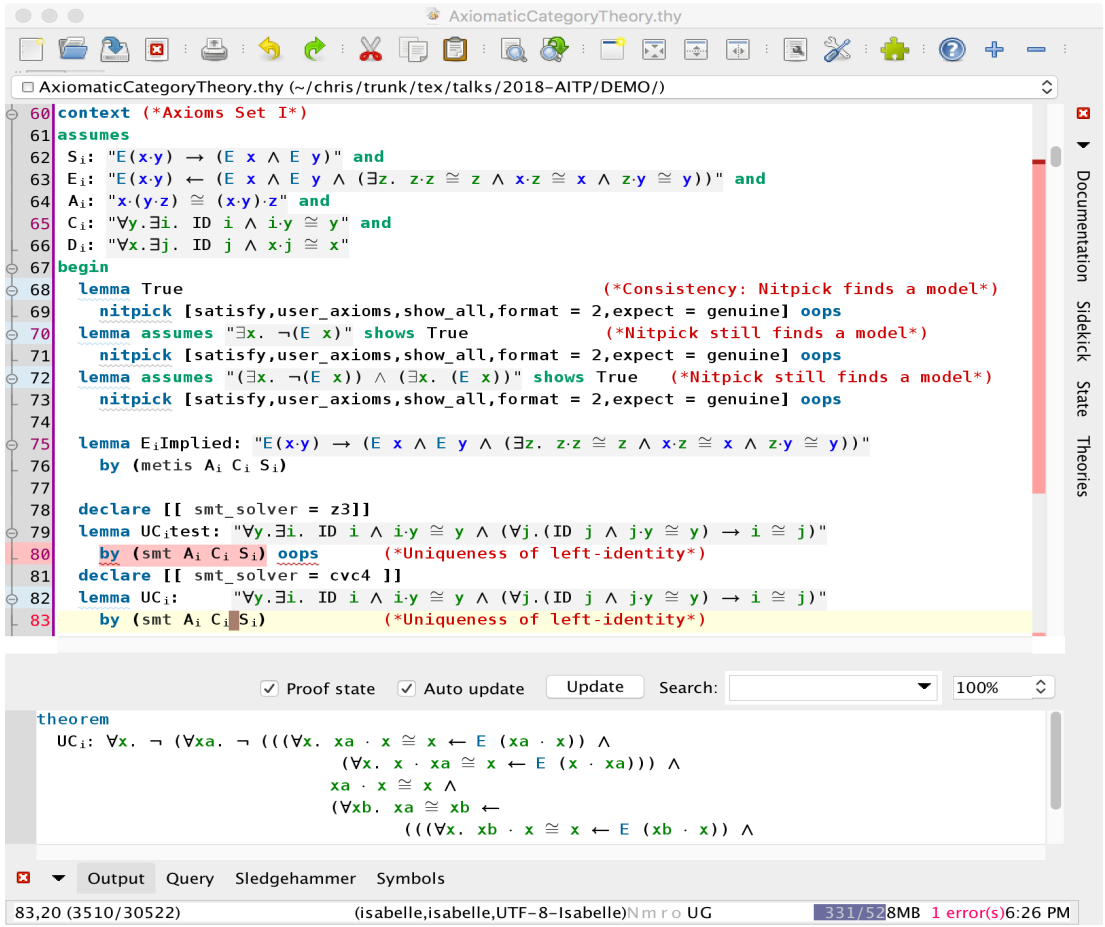
Some Reflections

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?
- Automation granularity much better than expected



Some Reflections

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?
- Automation granularity much better than expected
- Only initially ATPs found proofs which Isabelle could not verify
  - intermediate lemmata
  - switched from Z3 to CVC4
  - etc.



Some Reflections

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?
- Automation granularity much better than expected
- Only initially ATPs found proofs which Isabelle could not verify
- Due to use of “smt”-tactic our document was initially rejected by AFP

Some Reflections

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?
- Automation granularity much better than expected
- Only initially ATPs found proofs which Isabelle could not verify
- Due to use of “smt”-tactic our document was initially rejected by AFP
- Removing certain axioms from proof attempts often useful (associativity)

Some Reflections

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?
- Automation granularity much better than expected
- Only initially ATPs found proofs which Isabelle could not verify
- Due to use of “smt”-tactic our document was initially rejected by AFP
- Removing certain axioms from proof attempts often useful (associativity)
- Issues in Sledgehammer
  - Z3 may give false feedback: “The generated problem is unprovable”
  - Z3 ran into errors: “A prover error occurred ... (line 82 of General/basics.ML)”
  - SPASS ran into errors: “An internal error occurred”

Some Reflections

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?
- Automation granularity much better than expected
- Only initially ATPs found proofs which Isabelle could not verify
- Due to use of “smt”-tactic our document was initially rejected by AFP
- Removing certain axioms from proof attempts often useful (associativity)
- Issues in Sledgehammer
  - Z3 may give false feedback: “The generated problem is unprovable”
  - Z3 ran into errors: “A prover error occurred ... (line 82 of General/basics.ML)”
  - SPASS ran into errors: “An internal error occurred”
- CVC4 seems to perform best in this application domain

Some Reflections

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?
- Automation granularity much better than expected
- Only initially ATPs found proofs which Isabelle could not verify
- Due to use of “smt”-tactic our document was initially rejected by AFP
- Removing certain axioms from proof attempts often useful (associativity)
- Issues in Sledgehammer
  - Z3 may give false feedback: “The generated problem is unprovable”
  - Z3 ran into errors: “A prover error occurred ... (line 82 of General/basics.ML)”
  - SPASS ran into errors: “An internal error occurred”
- CVC4 seems to perform best in this application domain
- Overall: strengths of ATPs surprisingly complementary; they all contributed

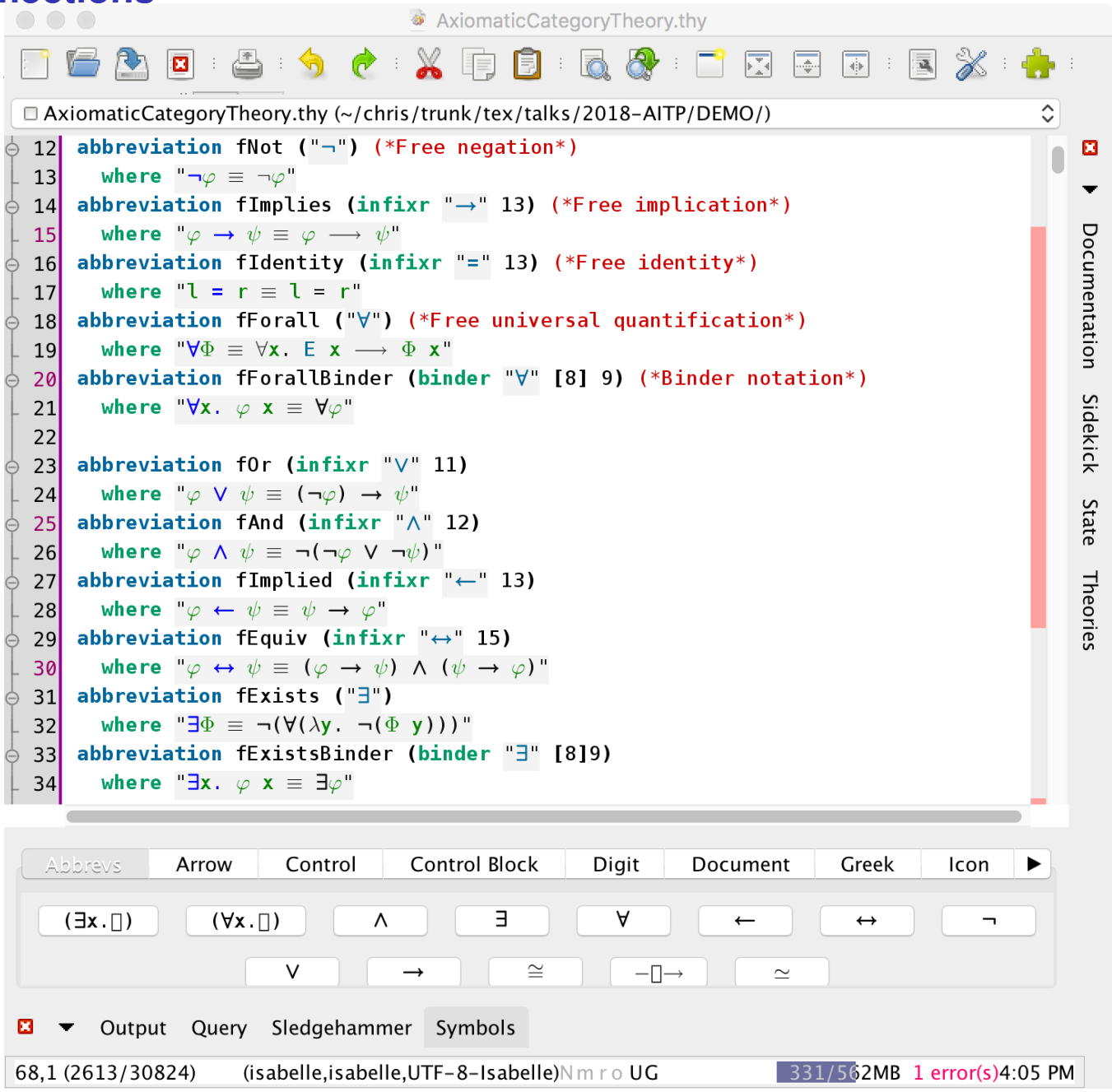
Some Reflections

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?
- Automation granularity much better than expected
- Only initially ATPs found proofs which Isabelle could not verify
- Due to use of “smt”-tactic our document was initially rejected by AFP
- Removing certain axioms from proof attempts often useful (associativity)
- Issues in Sledgehammer
  - Z3 may give false feedback: “The generated problem is unprovable”
  - Z3 ran into errors: “A prover error occurred ... (line 82 of General/basics.ML)”
  - SPASS ran into errors: “An internal error occurred”
- CVC4 seems to perform best in this application domain
- Overall: strengths of ATPs surprisingly complementary; they all contributed
- Most valuable tool: Nitpick (but results should be better presented)

Some Reflections

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?
- Automation granularity much better than expected
- Only initially ATPs found proofs which Isabelle could not verify
- Due to use of “smt”-tactic our document was initially rejected by AFP
- Removing certain axioms from proof attempts often useful (associativity)
- Issues in Sledgehammer
  - Z3 may give false feedback: “The generated problem is unprovable”
  - Z3 ran into errors: “A prover error occurred ... (line 82 of General/basics.ML)”
  - SPASS ran into errors: “An internal error occurred”
- CVC4 seems to perform best in this application domain
- Overall: strengths of ATPs surprisingly complementary; they all contributed
- Most valuable tool: Nitpick (but results should be better presented)
- Very useful: flexible support in GUI of Isabelle

Some Reflections



Isabelle) ?

AFP

ciativity)

asics.ML)”

ontributed

d)

Some Reflections

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?
- Automation granularity much better than expected
- Only initially ATPs found proofs which Isabelle could not verify
- Due to use of “smt”-tactic our document was initially rejected by AFP
- Removing certain axioms from proof attempts often useful (associativity)
- Issues in Sledgehammer
  - Z3 may give false feedback: “The generated problem is unprovable”
  - Z3 ran into errors: “A prover error occurred ... (line 82 of General/basics.ML)”
  - SPASS ran into errors: “An internal error occurred”
- CVC4 seems to perform best in this application domain
- Overall: strengths of ATPs surprisingly complementary; they all contributed
- Most valuable tool: Nitpick (but results should be better presented)
- Very useful: flexible support in GUI of Isabelle
- Very useful: Production of latex documents out of Isabelle

Some Reflections

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?
- Automation granularity much better than expected
- Only initially ATPs found proofs which Isabelle could not verify
- Due to use of “smt”-tactic our document was initially rejected by AFP
- Removing certain axioms from proof attempts often useful (associativity)
- Issues in Sledgehammer
  - Z3 may give false feedback: “The generated problem is unprovable”
  - Z3 ran into errors: “A prover error occurred ... (line 82 of General/basics.ML)”
  - SPASS ran into errors: “An internal error occurred”
- CVC4 seems to perform best in this application domain
- Overall: strengths of ATPs surprisingly complementary; they all contributed
- Most valuable tool: Nitpick (but results should be better presented)
- Very useful: flexible support in GUI of Isabelle
- Very useful: Production of latex documents out of Isabelle
- Further remark: No definitional hierarchy used in our experiments

Some Reflections

- ~~Domain expert (Dana)~~ — ~~tool expert (myself)~~ — proof assistant (Isabelle) ?
- Automation granularity much better than expected
- Only initially ATPs found proofs which Isabelle could not verify
- Due to use of “smt”-tactic our document was initially rejected by AFP
- Removing certain axioms from proof attempts often useful (associativity)
- Issues in Sledgehammer
  - Z3 may give false feedback: “The generated problem is unprovable”
  - Z3 ran into errors: “A prover error occurred ... (line 82 of General/basics.ML)”
  - SPASS ran into errors: “An internal error occurred”
- CVC4 seems to perform best in this application domain
- Overall: strengths of ATPs surprisingly complementary; they all contributed
- Most valuable tool: Nitpick (but results should be better presented)
- Very useful: flexible support in GUI of Isabelle
- Very useful: Production of latex documents out of Isabelle
- Further remark: No definitional hierarchy used in our experiments
- Proof assistant (in combination with ATPs and Nitpick) strongly fostered the intuitive exploration of the domain instead of behindering it

Some Remarks

Universal Logical Reasoning Approach: Selected Highlights

- Ontological Argument for the Existence of God
  - Different Variants of Extensional and Intensional Higher-Order Modal Logics
- Principia Logica-Metaphysica of Ed Zalta
  - Hyperintensional Higher-Order Modal Logic (based on Relational Type-Theory)
- Principle of Generic Consistency by Alan Gewirth
  - Combination of Higher-Order Modal Logic with a Modern Dyadic Deontic Logic
- Bostrom's Simulation Argument
- Boolos' Textbook on Provability Logic
- ...

First approach to cope with such a wide range of topics!

ATP Leo-III meanwhile accepts numerous ($\geq$ 120) Higher-Order Modal Logics and different Higher-Order Deontic Logics as native input!

Some Remarks

Universal Logical Reasoning Approach: Selected Highlights

- Ontological Argument for the Existence of God
  - Different Variants of Extensional and Intensional Higher-Order Modal Logics
- Principia Logica-Metaphysica of Ed Zalta
  - Hyperintensional Higher-Order Modal Logic (based on Relational Type-Theory)
- Principle of Generic Consistency by Alan Gewirth
  - Combination of Higher-Order Modal Logic with a Modern Dyadic Deontic Logic
- Bostrom's Simulation Argument
- Boolos' Textbook on Provability Logic
- ...

First approach to cope with such a wide range of topics!

ATP Leo-III meanwhile accepts numerous ($\geq$ 120) Higher-Order Modal Logics and different Higher-Order Deontic Logics as native input!

Some Remarks

Universal Logical Reasoning Approach: Selected Highlights

- Ontological Argument for the Existence of God
  - Different Variants of Extensional and Intensional Higher-Order Modal Logics
- Principia Logica-Metaphysica of Ed Zalta
  - Hyperintensional Higher-Order Modal Logic (based on Relational Type-Theory)
- Principle of Generic Consistency by Alan Gewirth
  - Combination of Higher-Order Modal Logic with a Modern Dyadic Deontic Logic
- Bostrom's Simulation Argument
- Boolos' Textbook on Provability Logic
- ...

First approach to cope with such a wide range of topics!

ATP Leo-III meanwhile accepts numerous ($\geq$ 120) Higher-Order Modal Logics and different Higher-Order Deontic Logics as native input!