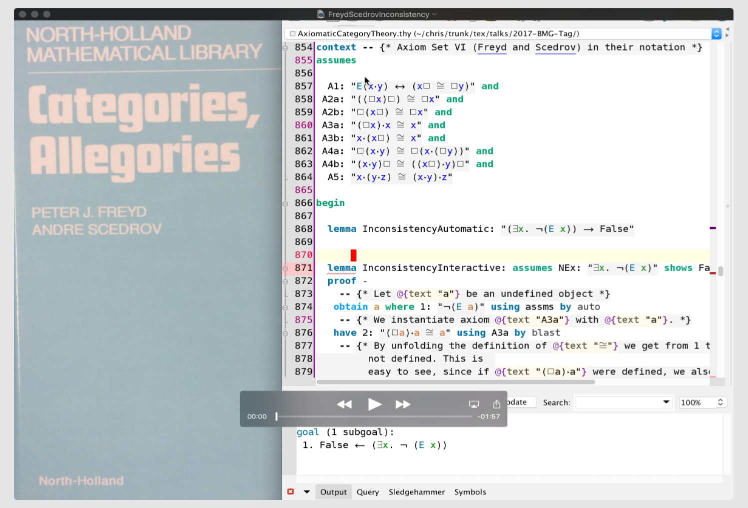
Some Reflections on a Computer-aided Theory Exploration Study in Category Theory

Christoph Benzmüller and Dana Scott



AITP 2018

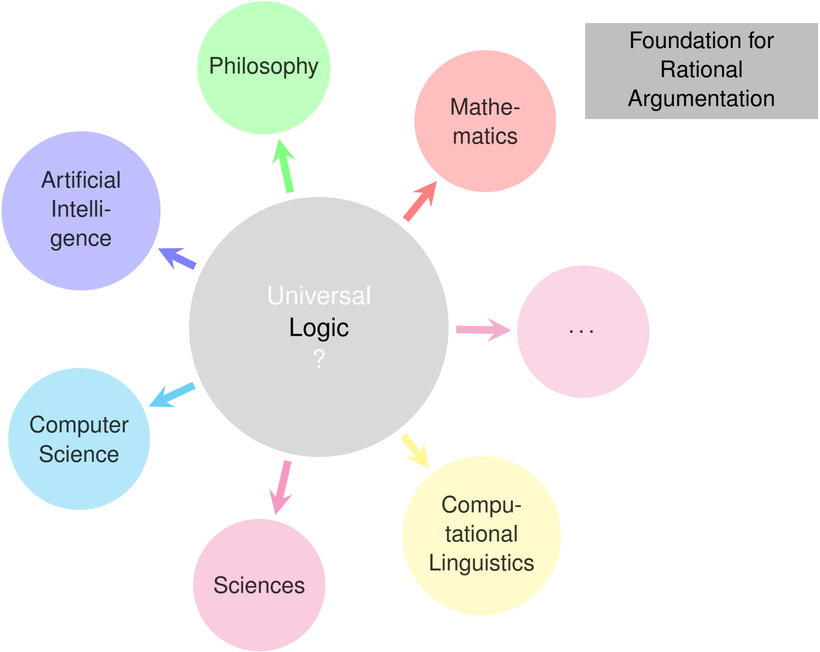
Presentation Outline

"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."

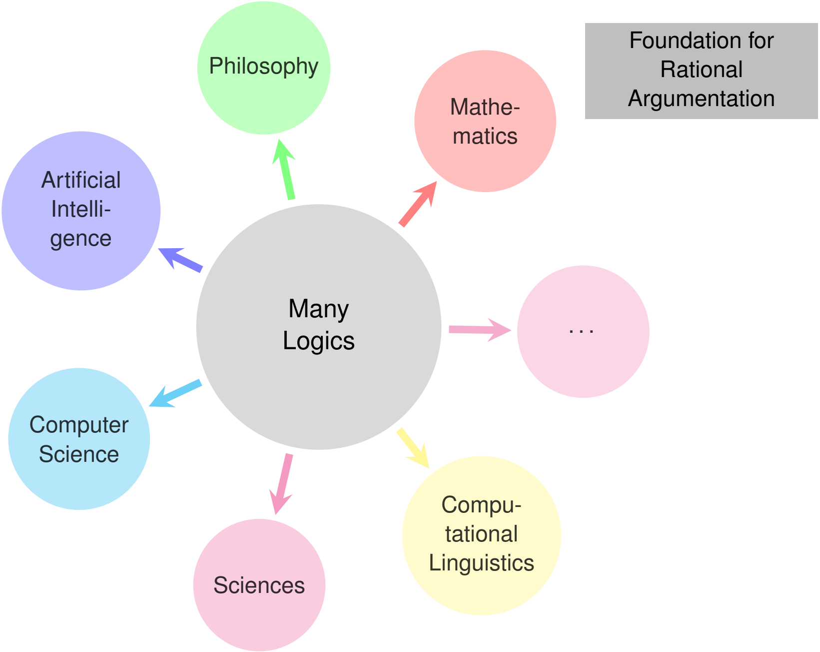
(Leibniz, 1677)

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

Part A
Universal Reasoning in Meta-logic HOL
(utilising Shallow Semantical Embeddings):

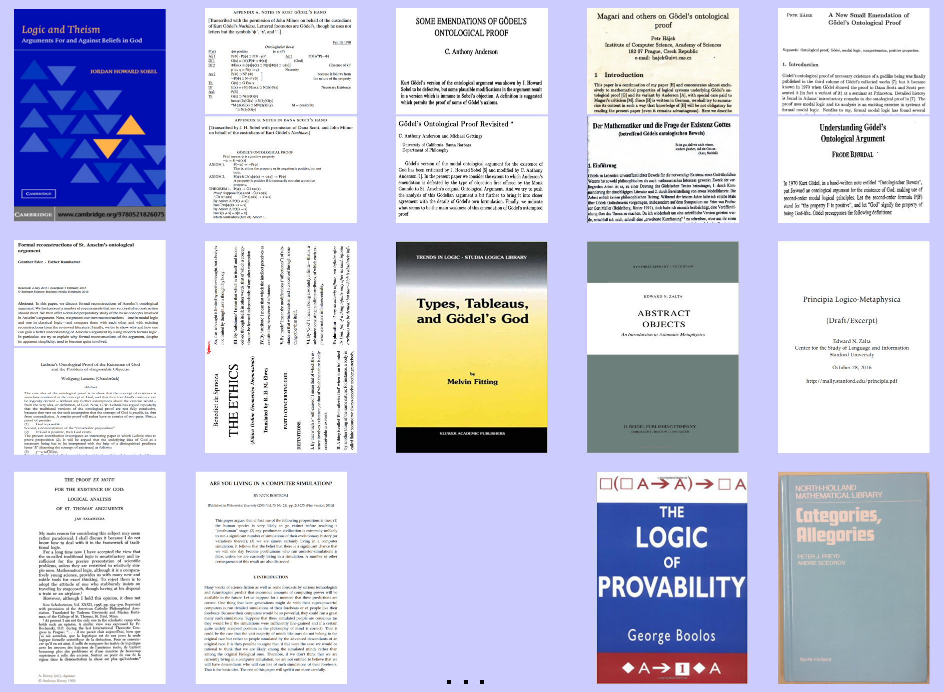


C. Benzmüller & D. Scott, 2018

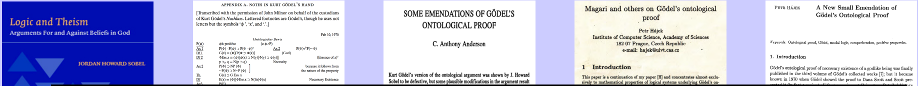


C. Benzmüller & D. Scott, 2018

Logic Zoo



C. Benzmüller & D. Scott, 2018

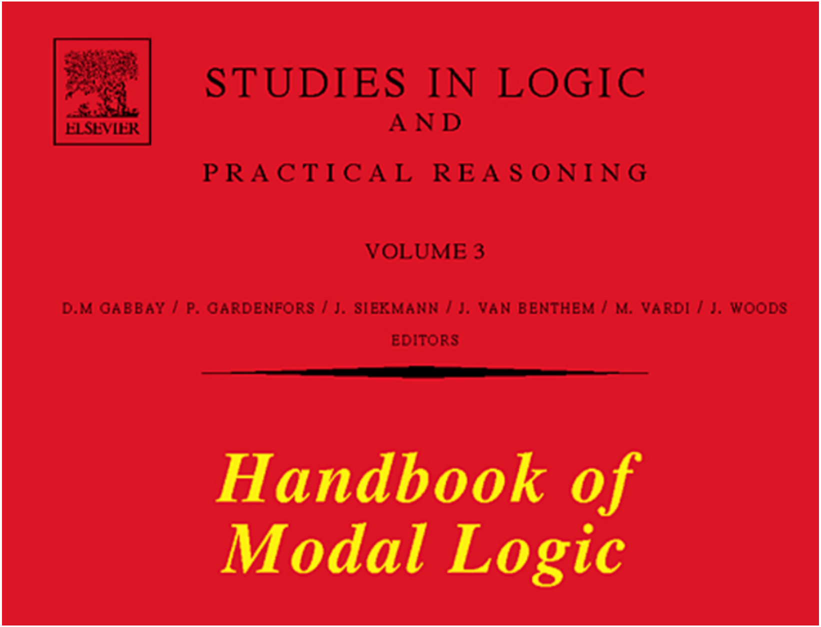


Jww colleagues: formalisation of scientific articles and textbooks

- ... 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:

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

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):

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

Example: Modal Logic Textbook

A *model* (or *Kripke model*) $\mathfrak{M}$ for the basic modal language (over some fixed signature) is a triple $\mathfrak{M} = (W, \{R^m\}_{m \in \mathrm{MOD}}, V)$. 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 V… position symbol p in PROP a subset $V(p)$ … p is true. The first 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 $\mathfrak{M} = (W, \{R^m\}_{m \in \mathrm{MOD}}, V)$. 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):

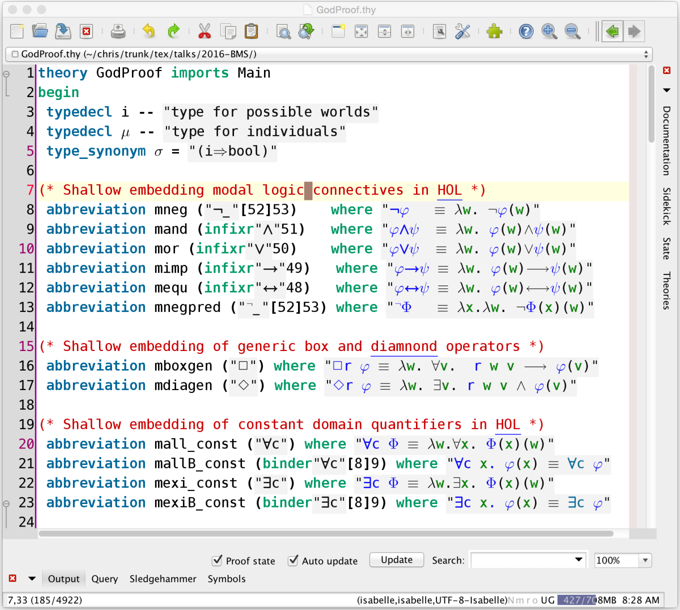
Semantics

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

Example: Modal Logic Textbook

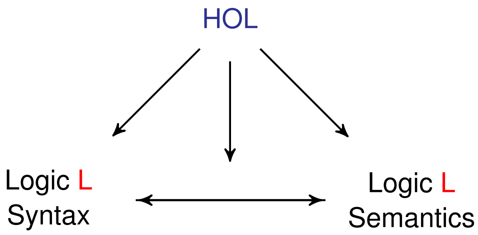
$$
\begin{array}{rcl}
\mathfrak{M}, w \models p & \text{iff} & w \in V(p), \\
\mathfrak{M}, w \models \top & & \text{always}, \\
\mathfrak{M}, w \models \bot & & \text{never}, \\
\mathfrak{M}, w \models \neg\varphi & \text{iff} & \text{not } \mathfrak{M}, w \models \varphi \text{ (notation: } \mathfrak{M},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

Universal Logic Reasoning in Isabelle/HOL



```
theory GodProof imports Main
begin
 typedecl i -- "type for possible worlds"
 typedecl μ -- "type for individuals"
 type_synonym σ = "(i⇒bool)"

(* Shallow embedding modal logic connectives in HOL *)
 abbreviation mneg ("¬_"[52]53)    where "¬φ   ≡ λw. ¬φ(w)"
 abbreviation mand (infixr"∧"51)   where "φ∧ψ  ≡ λw. φ(w)∧ψ(w)"
 abbreviation mor (infixr"∨"50)    where "φ∨ψ  ≡ λw. φ(w)∨ψ(w)"
 abbreviation mimp (infixr"→"49)   where "φ→ψ  ≡ λw. φ(w)⟶ψ(w)"
 abbreviation mequ (infixr"↔"48)   where "φ↔ψ  ≡ λw. φ(w)⟷ψ(w)"
 abbreviation mnegpred ("¬_"[52]53) where "¬Φ  ≡ λx.λw. ¬Φ(x)(w)"

(* Shallow embedding of generic box and diamnond operators *)
 abbreviation mboxgen ("□") where "□r φ ≡ λw. ∀v.  r w v ⟶ φ(v)"
 abbreviation mdiagen ("◇") where "◇r φ ≡ λw. ∃v. r w v ∧ φ(v)"

(* Shallow embedding of constant domain quantifiers in HOL *)
 abbreviation mall_const ("∀c") where "∀c Φ ≡ λw.∀x. Φ(x)(w)"
 abbreviation mallB_const (binder"∀c"[8]9) where "∀c x. φ(x) ≡ ∀c φ"
 abbreviation mexi_const ("∃c") where "∃c Φ ≡ λw.∃x. Φ(x)(w)"
 abbreviation mexiB_const (binder"∃c"[8]9) where "∃c x. φ(x) ≡ ∃c φ"
```

Universal Logic Reasoning in HOL



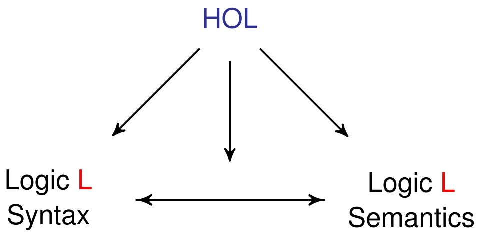
Examples for L we have already studied:
Intuitionistic Logics, Modal Logics, Description Logics, Conditional Logics, Access Control Logics, Hybrid Logics, Multivalued Logics, Paraconsistent Logics, **Hyper-intensional Higher-Order Modal Logic**, **Free Logic**, **Dyadic Deontic Logic**, **Input/Output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



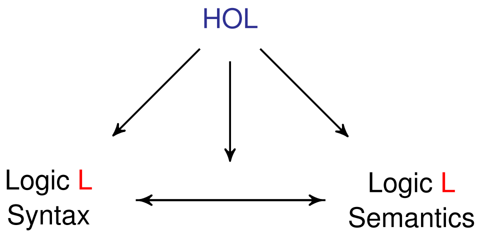
Examples for L we have already studied:
Intuitionistic Logics, Modal Logics, Description Logics, Conditional Logics, Access Control Logics, Hybrid Logics, Multivalued Logics, Paraconsistent Logics, **Hyper-intensional Higher-Order Modal Logic**, **Free Logic**, **Dyadic Deontic Logic**, **Input/Output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

Part B:
Free Logic in HOL

[Free Logic in Isabelle/HOL, ICMS, 2016]
[Axiomatizing Category Theory in Free Logic, arXiv:1609.01493, 2016]

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

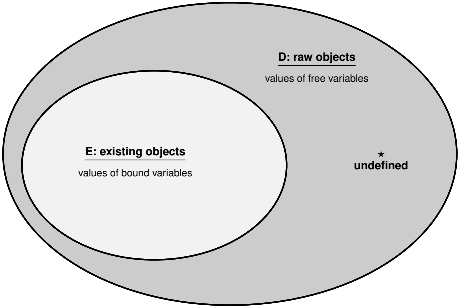
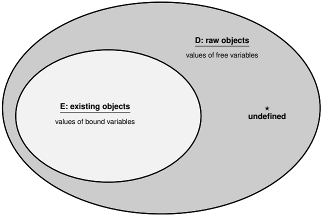
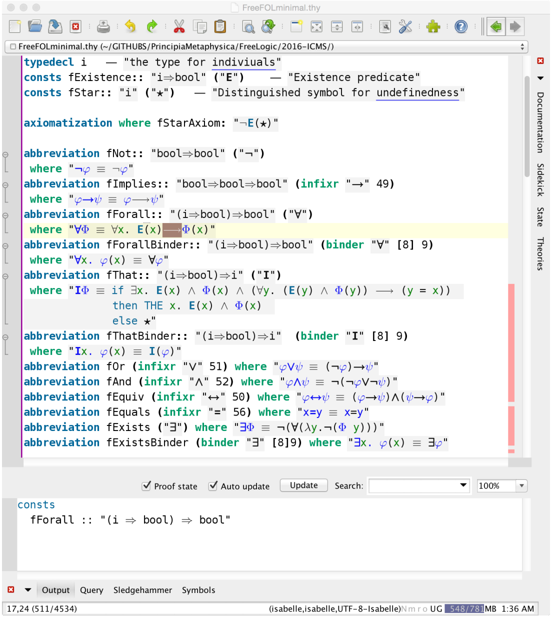


Figure: Illustration of the semantical domains of free logic

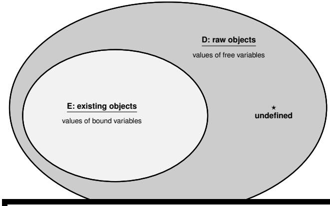
Free Logic in HOL





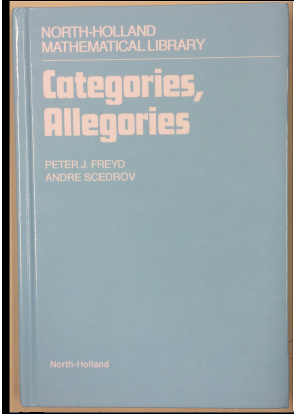
Free Logic in HOL

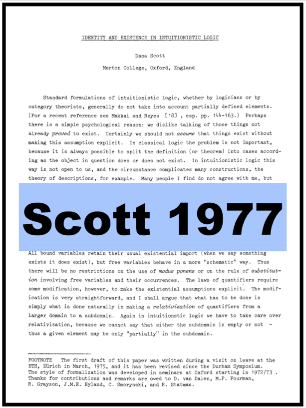
```
abbreviation fForall ("∀") (*Free universal quantification*)
  where "∀Φ ≡ ∀x. E x ⟶ Φ x"
abbreviation fForallBinder (binder "∀" [8] 9) (*Binder notation*)
  where "∀x. φ x ≡ ∀φ"
```




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

```
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)"
```





IDENTITY AND EXISTENCE 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 [18] , esp. pp. 144-163.) 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

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 intuitionstic logic we have to take care over relativization, 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, 1975, and it has been revised since the Durham Symposium. The style of formalization was developed in seminars at Oxford starting in 1972/73 . Thanks for contributions and remarks are owed to D. van Dalen, M.P. Fourman, R. Grayson, J.M.E. Hyland, C. Smorynski, and R. Statman.

Part C:
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

Axioms Set I

Generalized Monoids

Axioms Set II

Dana Scott

Exemplary Case Study: Exploration of Axioms Sets for Category Theory

Axioms Set I

Generalized Monoids

Axioms Set II

Axioms Set III

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

Dana Scott

Exemplary Case Study: Exploration of Axioms Sets for Category Theory

Axioms Set I

Generalized Monoids

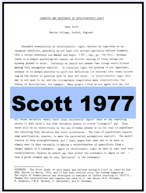
Axioms Set II

Axioms Set III

Axioms Set IV

Axioms Set V

Dana Scott's Axioms from 1977


IDENTITY AND EXISTENCE 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 [18], esp. pp. 144-163.) 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 relativization, 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, 1975, and it has been revised since the Durham Symposium. The style of formalisation was developed in seminars at Oxford starting in 1972/73. Thanks for contributions and remarks are owed to D. van Dalen, M.P. Fourman, R. Grayson, J.M.E. Hyland, C. Smorynski, and R. Statman.

Dana Scott

Exemplary Case Study: Exploration of Axioms Sets for Category Theory

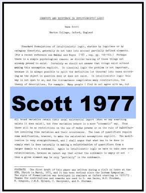
Axioms Set I

Generalized Monoids

Axioms Set II

Axioms Set III

Axioms Set IV



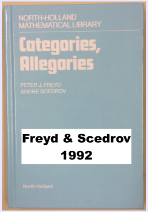
Axioms Set V

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

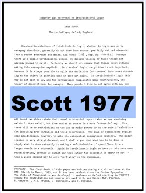
Axioms Set I

Generalized Monoids

Axioms Set II

Axioms Set III

Axioms Set IV



IDENTITY AND EXISTENCE 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 [18], esp. pp. 144-163.) 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

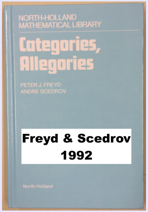
All bound variables retain their usual existential import (when we say something exists it does exist), but free variables behave in a more "schematic" way. Thus there will be no restrictions on the use of modus ponens or on the rule of substitution involving free variables and their occurrences. The 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 relativization, 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, 1975, and it has been revised since the Durham Symposium. The style of formalization was developed in seminars at Oxford starting in 1972/73. Thanks for contributions and remarks are owed to D. van Dalen, M.P. Fourman, R. Grayson, J.M.E. Hyland, C. Smorynski, and R. Statman.

Axioms Set V

Dana Scott's Axioms from 1977

—?—



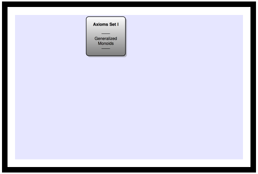
Axioms Set VI

Freyd & Scedrov's Axioms from 1992

Dana Scott

all equivalent?

Preliminaries

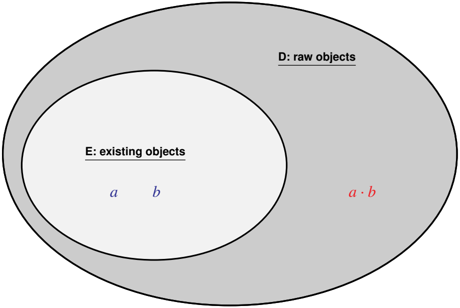


Morphisms: objects of type of i (raw domain D)

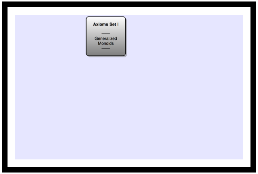
Partial functions:

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

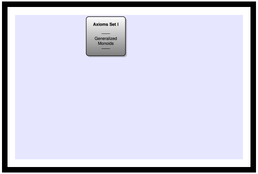


Morphisms: objects of type of i (raw domain D)

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:

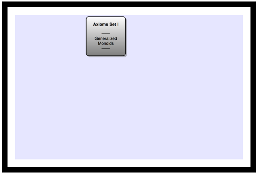
domain	dom	of type $i \rightarrow i$
codomain	cod	of type $i \rightarrow i$
composition	$\cdot$	of type $i \rightarrow i \rightarrow i$ (resp. $i \times i \rightarrow i$)

$\cong$ 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**.

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
- $1/0 \not\simeq 1/0 \quad 1/0 \not\simeq 2/0 \quad \dots$
- $Ix.pkoFrance(x) \not\simeq Ix.pkoFrance(x)$
  $Ix.pkoFrance(x) \not\simeq Ix.pkoPoland(x)$

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

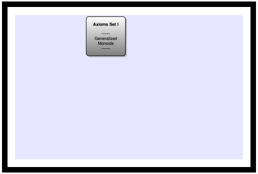
(where $=$ is identity on all objects of type i, existing or non-existing)

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

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

$\simeq$ is symmetric and transitive, but lacks reflexivity: **Sledgehammer**, **Nitpick**.

From Monoids to Categories



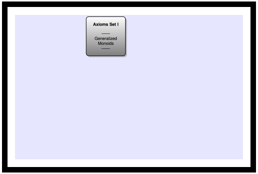
Monoid

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, .

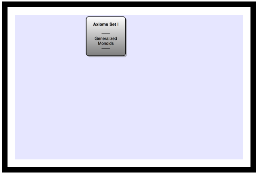
Categories: Axioms Set I

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:

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

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	$\forall x. \exists j. ID(j) \wedge x \cdot j \cong x$

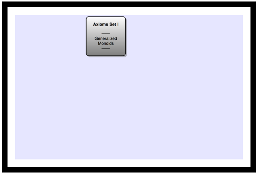
where I is an identity morphism predicate:

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

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	$\forall x. \exists j. ID(j) \wedge x \cdot j \cong x$

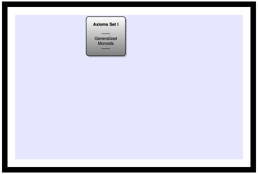
where I is an identity morphism predicate:

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

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:

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

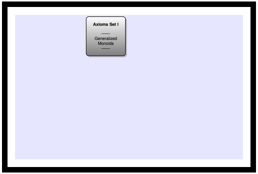
Experiments with Isabelle/HOL

- 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:

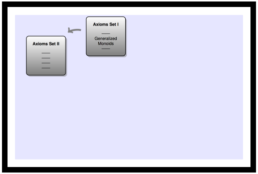
dom(i1)=i1, dom(i2)=i2, dom(i3)=i3
cod(i1)=i1, cod(i2)=i2, cod(i3)=i3
i1.i1=i1, i1.i2=i3, i1.i3=i3
i2.i1=i3, i2.i2=i2, i2.i3=i3
i3.i1=i3, i3.i2=i3, i3.i3=i3
E(i1),E(i2), ~E(i3)

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.

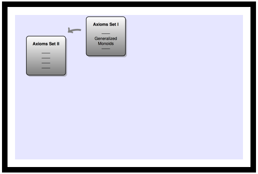
Categories: Axioms Set II

S_{ii}	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey) \wedge (E(dom\ x) \rightarrow Ex) \wedge (E(cod\ y) \rightarrow Ey)$
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	$Ex \rightarrow (ID(dom\ x) \wedge x \cdot (dom\ x) \cong x)$

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	$\forall x. \exists j. ID(j) \wedge x \cdot j \cong x$

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

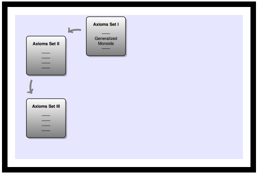
S_{ii}	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey) \wedge (E(dom\ x) \rightarrow Ex) \wedge (E(cod\ y) \rightarrow Ey)$
E_{ii}	Existence	$E(x \cdot y) \leftarrow (Ex \wedge Ey \wedge (\exists z. z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y))$
A_{ii}	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
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**.
- Axiom Set II implies Axioms Set I: easily proved by **Sledgehammer**.
- Axiom 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.



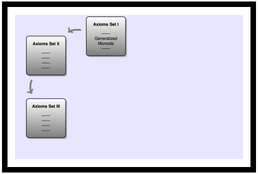
Categories: Axioms Set III

S_{iii}	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey) \wedge (E(dom\ x) \rightarrow Ex) \wedge (E(cod\ y) \rightarrow Ey)$
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	$Ex \rightarrow (ID(dom\ x) \wedge x \cdot (dom\ x) \cong x)$

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	$Ex \rightarrow (ID(dom\ x) \wedge x \cdot (dom\ x) \cong x)$

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)

AxiomaticCategoryTheorySimplifiedAxiomSetIE1.thy

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

```
153 context (* Axiom Set III *)
154 assumes
155  Sᵢᵢᵢ: "(E(x·y) ⟶ (E x ∧ E y)) ∧ (E(dom x ) ⟶ E x) ∧ (E(cod y) ⟶ E y)"  and
156  Eᵢᵢᵢ: "E(x·y) ⟵ (dom x ≅ cod y ∧ E(cod y))" and
157  Aᵢᵢᵢ: "x·(y·z) ≅ (x·y)·z" and
158  Cᵢᵢᵢ: "E y ⟶ (ID(cod y) ∧ (cod y)·y ≅ y)" and
159  Dᵢᵢᵢ: "E x ⟶ (ID(dom x) ∧ x·(dom x) ≅ x)"
160 begin
161  (* lemma EᵢᵢFromIII: "E(x·y) ⟵ (E x ∧ E y ∧ (∃z. z·z ≅ z ∧ x·z ≅ x ∧ z·y ≅ y))" *)
162     lemma EᵢᵢFromIII: "E(x·y) ⟵ (E x ∧ E y)" nitpick [show_all,format=2] (*Countermodel*) (
163 end
```

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

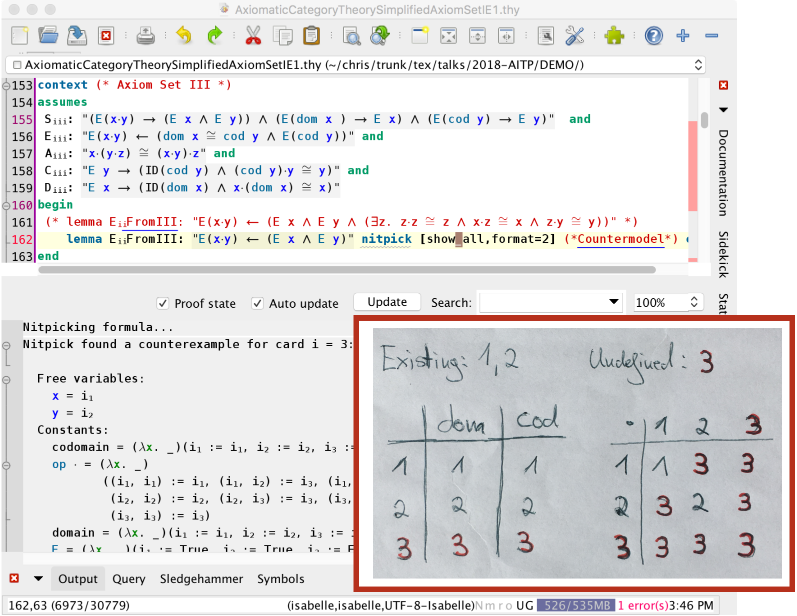
```
Nitpicking formula...
Nitpick found a counterexample for card i = 3:

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

Output · Query · Sledgehammer · Symbols

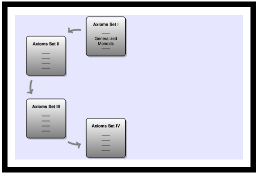
Documentation · Sidekick · State · Theories

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

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



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

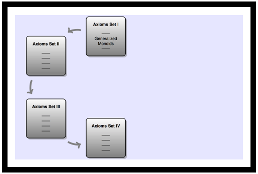
S_{iv}	Strictness	$E(x \cdot y) \rightarrow (Ex \wedge Ey) \wedge (E(dom\ x) \rightarrow Ex) \wedge (E(cod\ y) \rightarrow Ey)$
E_{iv}	Existence	$E(x \cdot y) \leftrightarrow (dom\ x \cong cod\ y \wedge E(cod\ y))$
A_{iv}	Associativity	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_{iv}	Codomain	$(cod\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



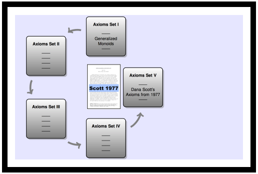
Categories: Axioms Set IV

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.

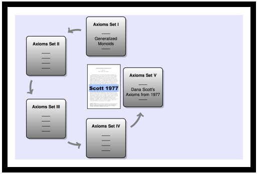
Categories: Axioms Set V (Scott, 1977)

$S1$	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.

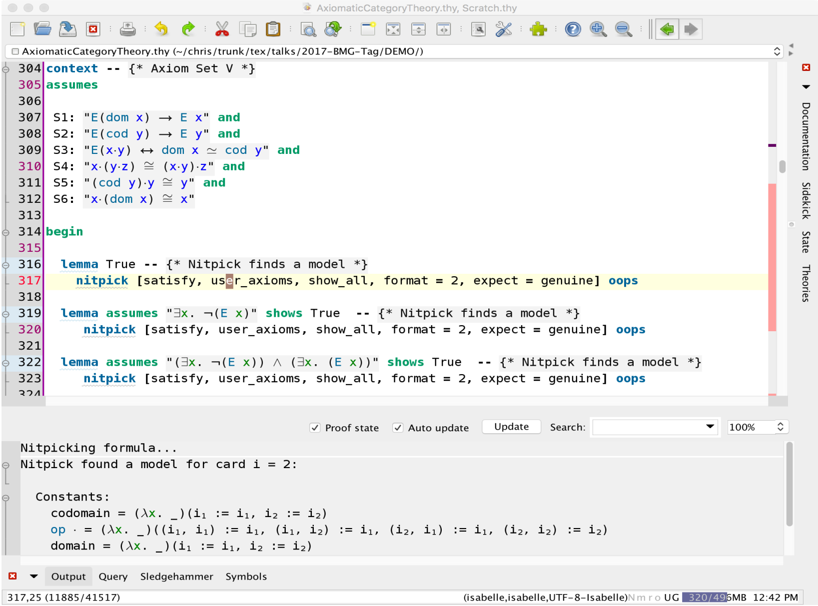
Categories: Axioms Set V (Scott, 1977)

$S1$	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	$x \cdot (dom\ x) \cong x$

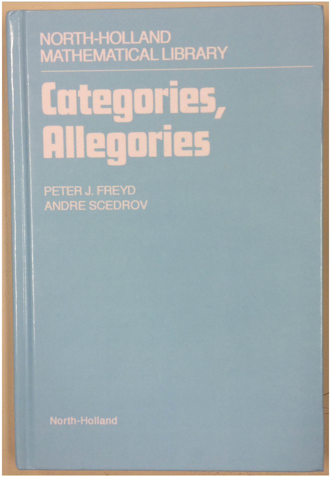
Experiments with Isabelle/HOL

- 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/)
context -- {* Axiom Set V *}
assumes
S1: "E(dom x) ⟶ E x" and
S2: "E(cod y) ⟶ E y" and
S3: "E(x·y) ⟷ dom x ≃ cod y" and
S4: "x·(y·z) ≅ (x·y)·z" and
S5: "(cod y)·y ≅ y" and
S6: "x·(dom x) ≅ x"
begin
lemma True -- {* Nitpick finds a model *}
nitpick [satisfy, user_axioms, show_all, format = 2, expect = genuine] oops
lemma assumes "∃x. ¬(E x)" shows True -- {* Nitpick finds a model *}
nitpick [satisfy, user_axioms, show_all, format = 2, expect = genuine] oops
lemma assumes "(∃x. ¬(E x)) ∧ (∃x. (E x))" shows True -- {* Nitpick finds a model *}
nitpick [satisfy, user_axioms, show_all, format = 2, expect = genuine] oops
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
Documentation
Sidekick
State
Theories
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:

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

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

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

The axioms:

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

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

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

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

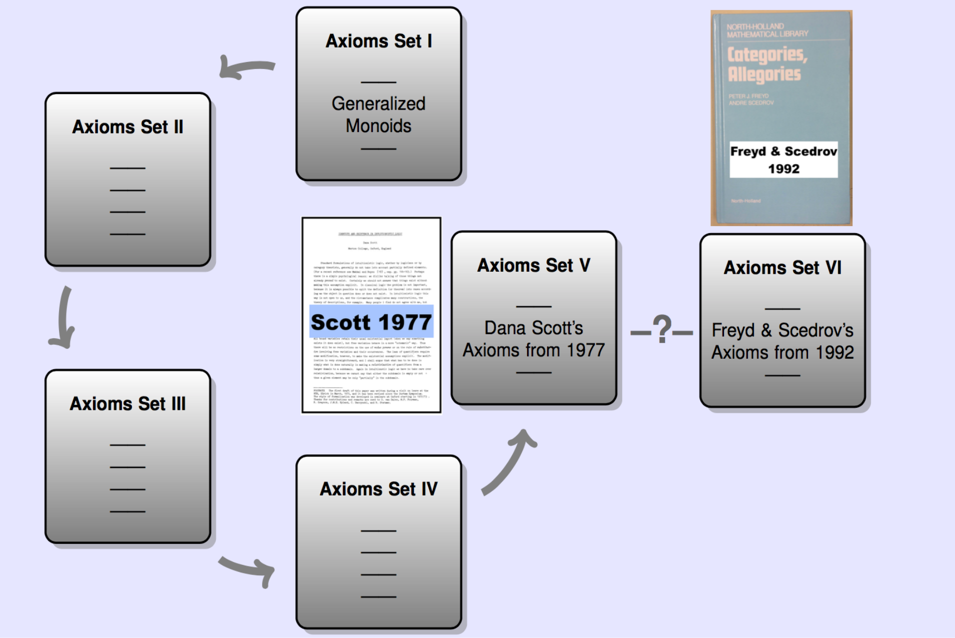
A5 $x(yz) = (xy)z$.

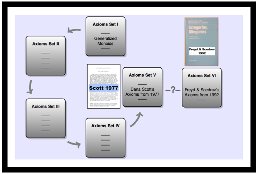
1.11. The ordinary equality sign = will be used only in the symmetric sense, to wit: if either side is defined then so is the other and they are equal. A theory, such as this, built on an ordered list of partial operations, the domain of definition of each given by equations in the previous, and with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

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

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

Cats & Alligators



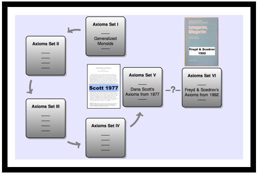


Categories: Original axiom set by Freyd and Scedrov (modulo notation)

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: $A1, A2a, A3a$)



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

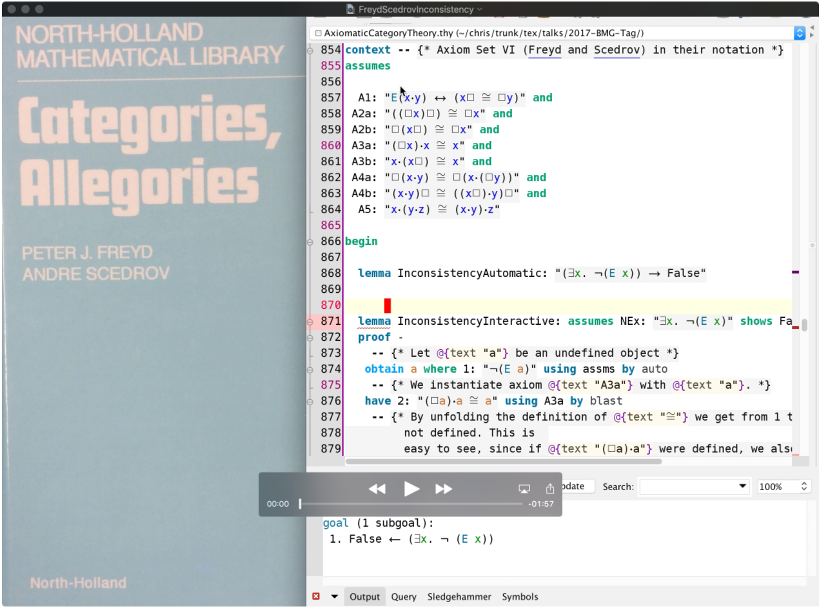
A1	$E(x \cdot y) \leftrightarrow dom\ x \ \cong cod\ y$
A2a	$cod(dom\ x) \cong dom\ x$
A2b	$dom(cod\ y) \cong cod\ y$
A3a	$x \cdot (dom\ x) \cong x$
A3b	$(cod\ y) \cdot y \cong y$
A4a	$dom(x \cdot y) \cong dom((dom\ x) \cdot y)$
A4b	$cod(x \cdot y) \cong cod(x \cdot (cod\ y))$
A5	$x \cdot (y \cdot z)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

Experiments with Isabelle/HOL

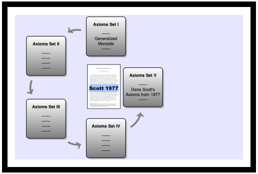
- Consistency? — Nitpick finds a model.
- Consistency when assuming $\exists x. \neg Ex$ — Nitpick does not find a model.
- lemma $(\exists x. \neg Ex) \rightarrow False$: **Sledgehammer**. (Problematic axioms: $A1, A2a, A3a$)

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


FreydScedrovInconsistency
NORTH-HOLLAND MATHEMATICAL LIBRARY
Categories, Allegories
PETER J. FREYD
ANDRE SCEDROV
North-Holland
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



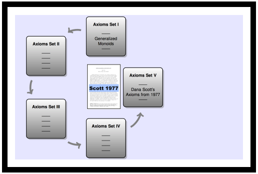
Categories: Axioms Set VI (Freyd and Scedrov, when corrected)

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.



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	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$

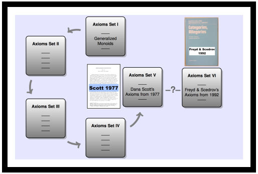
Experiments with Isabelle/HOL

- 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:



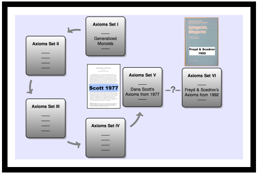
Categories: "Algebraic reading" of axiom set by Freyd and Scedrov.

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).

Part D: Some Reflections

Some Reflections

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

Some Reflections

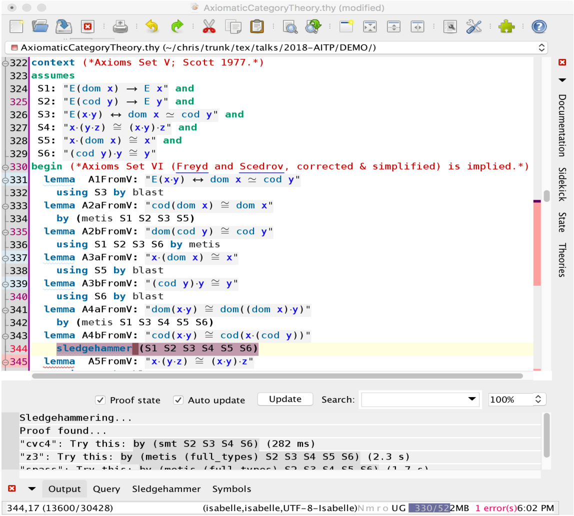
- Domain expert (Dana) — ~~tool expert (myself)~~ — proof assistant (Isabelle)

Some Reflections

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

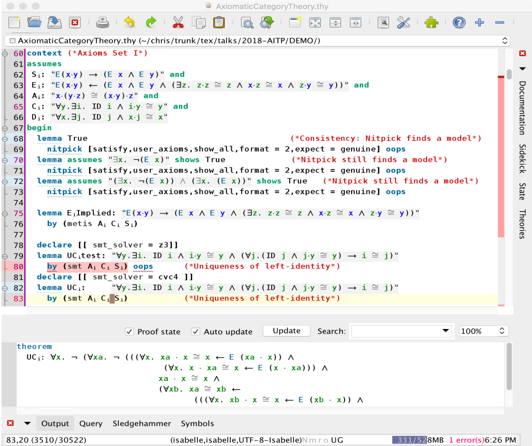
Some Reflections

- ~~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 is not (yet) in 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 is not (yet) in 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 is not (yet) in 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 is not (yet) in 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 is not (yet) in 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

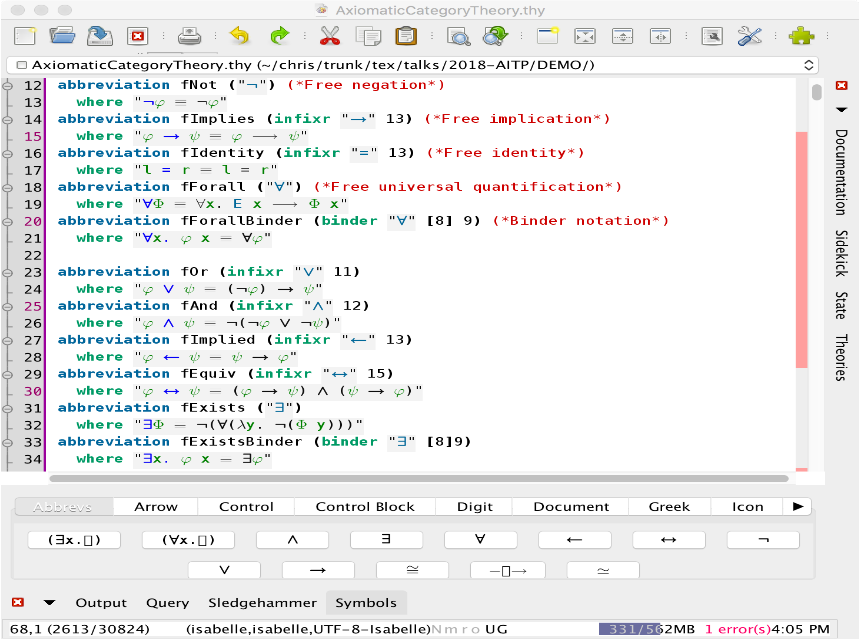
Some Reflections

- ~~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 is not (yet) in 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 is not (yet) in 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



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 is not (yet) in 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 is not (yet) in 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 is not (yet) in 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

Conclusion

Interesting and useful exploration study in Category Theory

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 existing ITP/ATPs, high degree of **automation**
- **Uniform proofs** (modulo the embeddings)
- **Intuitive user interaction** at abstract level
- 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
```