Noname manuscript No.
(will be inserted by the editor)

Automating Free Logic in HOL, with an Application in Category Theory

Christoph Benzmüller · Dana Scott

Received: date / Accepted: date

Abstract A shallow semantical embedding of free logic in classical higher-order logic is presented, which enables the off-the-shelf application of higher-order interactive and automated theorem provers (and their integrated sub-provers) for the formalisation and verification of free logic theories. Subsequently, this approach is exemplarily employed in a selected domain of mathematics: starting from a generalization of the standard axioms for a monoid we present a stepwise development of various, mutually equivalent foundational axiom systems for category theory. As a side-effect of this work some (minor) issue in a prominent category theory textbook has been revealed.

The purpose of this article is not to claim any novel results in category theory, but to demonstrate an elegant way to "implement" and utilise interactive and automated reasoning in free logic, and to present respective experiments.

Keywords Free Logic · Classical Higher-Order Logic · Category Theory · Interactive and Automated Theorem Proving

1 Introduction

Partiality and undefinedness are prominent challenges in various areas of mathematics and computer science. Unfortunately, however, modern proof assistant systems and automated theorem provers based on traditional classical or intuitionistic logics provide rather inadequate support for these challenge concepts.

Mention DFG grants ...

F. Author
first address
Tel.: +123-45-678910
Fax: +123-45-678910
E-mail: fauthor@example.com

S. Author
second address

Free logic offers a theoretically appealing solution, but is has been considered as rather unsuited towards practical utilisation.

In the first part of this article (§2 and §3) we show how free logic can be elegantly "implemented" in any theorem proving system for classical higher-order logic (HOL) [1]. The proposed solution employs a semantic embedding of free (or inclusive logic) in HOL. We present an exemplary implementation of this idea in the mathematical proof assistant Isabelle/HOL [16]. Various state-of-the-art first-order and higher-order automated theorem provers and model finders are integrated (modulo suitable logic translations) with Isabelle via the Sledgehammer tool [6], so that our solution can be utilized, via Isabelle as foreground system, with a whole range of other background reasoners. As a result we obtain an elegant and powerful implementation of an interactive and automated theorem proving (and model finding) system for free logic.

To demonstrate the practical relevance of our new system, we present, in the second part of this article, a stepwise development of axiom systems for category theory by generalizing the standard axioms for a monoid to a partial composition operation. Our purpose is not to make or claim any contribution to category theory but rather to show how formalizations involving the kind of logic required (free logic) can be implemented and validated within modern proof assistants.

A total of eight different axiom systems is studied. The systems I-VI are shown to be equivalent. The axiom system VII slightly modifies axiom system VI to obtain (modulo notational transformation) the set of axioms as proposed by Freyd and Scedrov in their textbook "Categories, Allegories" [11], published in 1990; see also Subsection **??** where we present their original system. While the axiom systems I-VI are shown to be consistent, a constricted inconsistency result is obtained for system VII (when encoded in free logic where free variables range over all objects): We can prove $\exists x.\neg(Ex) \rightarrow False$, where E is the existence predicate. Read this as: If there are undefined objects, e.g. the value of an undefined composition $x \cdot y$ then we have falsity. By contraposition, all objects (and thus all compositions) must exist. But when we assume the latter, then the axiom system VII essentially reduces categories to monoids. We note that axiom system V, which avoids this problem, corresponds to a set of axioms proposed by Scott [19] in the 1970s. The problem can also be avoided by restricting the variables in axiom system VII to range only over existing objects and by postulating strictness conditions. This gives us axiom system VIII.

Our exploration has been significantly supported by series of experiments in which automated reasoning tools have been called from within the proof assistant Isabelle/HOL via the Sledgehammer tool. Moreover, we have obtained very useful feedback at various stages from the model finder Nitpick [7] saving us from making several mistakes.

At the conceptual level this paper exemplifies a new style of explorative mathematics which rests on a significant amount of human-machine interaction with integrated interactive-automated theorem proving technology. The experiments we have conducted are such that the required reasoning is often too tedious and time-consuming for humans to be carried out repeatedly with highest level of precision. It is here where cycles of formalization and experimentation efforts in Isabelle/HOL provided significant support. Moreover, the technical inconsistency issue for axiom system VII was discovered by automated theorem provers, which further emphasises the added value of automated theorem proving in this area.

The content of article is combines, extends and clarifies the contributions reported in two previous papers [2,3].

2 Preliminaries

2.1 Free Logic

Free logic [13,18,14] refers to a class of logic formalisms that are free of basic existence assumptions regarding the denotation of terms. Remember that terms in e.g. traditional classical and intuitionistic predicate logics always denote an (existing) object in a given (non-empty) domain D, and that D is also exactly the set the quantifiers range over. In free logic these basic assumption are abolished. Terms do still denote objects in a (non-empty) domain D, but a (possibly empty) set $E \subseteq D$ is chosen to characterize the subdomain of "existing" resp. "defined" objects in D. Quantification is now restricted to set E of existing/defined objects only.

It is obvious how this can be used to model undefideness and partiality: problematic terms, e.g. division by zero or improper definite descriptions, still denote, but they refer to undefined objects, that is, objects d in $D \setminus E$ lying outside of the scope of quantification. Moreover, a function f is *total* if and only if for all x we have $Ex \longrightarrow E(fx)$ For *partial* functions f we may have some x such that Ex but not $E(fx)$. A function f is *strict* if and only if for all x we have $E(fx) \longrightarrow Ex$.

The particular notion of free logic as exploited in the remainder of this article has been proposed by Scott [18]. A graphical illustration of this notion of free logic is presented in Fig. 1. It employs a distinguished undefined object $\star$.

We now formally introduce the syntax and semantics of free logic as assumed in the remainder of this article. We refer to this logic with *FFOL*.

Definition 1 (Syntax of *FFOL*) We start with a denumerable set V of variable symbols, a denumerable set F of n-ary function symbols ($n \geq 0$), and a denumerable set P of n-ary predicate symbols ($n \geq 0$).

The *terms and formulas of FFOL* are formally defined as the smallest sets such that:

1. each variable $x \in V$ is a term of *FFOL*,
2. given any n-ary ($n \geq 0$) function symbol $f \in F$ and terms $t_1, \ldots, t_n$ of *FFOL*, then $f(t_1, \ldots, t_n)$ is a term of *FFOL*,

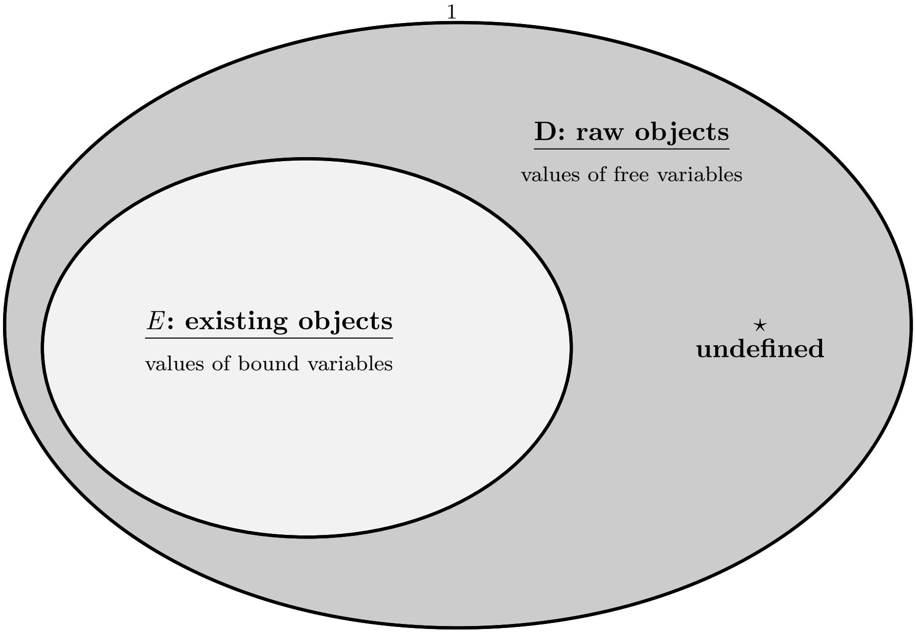

Fig. 1 Illustration of the Semantical Domains of Free Logic

3. given terms t_1 and t_2 of *FFOL*, then $t_1 = t_2$ is an (atomic) formula of *FFOL*,
4. given any n-ary ($n \geq 0$) predicate symbol $p \in P$ and terms $t_1, \ldots, t_n$ of *FFOL*, then $p(t_1, \ldots, t_n)$ is an (atomic) formula of *FFOL*,
5. given formulas r and s of *FFOL*, then $\neg r$, $r \rightarrow s$ and $\forall x.r$ are (compound) formulas of *FFOL*
6. given a formula r of *FFOL*, then $\imath x.r$ is a term of *FFOL* (definite description)

Further formulas of *FFOL*, including various defined notions of equality, can be introduced as abbreviations.

Substitution of a term r for a variable x in a term s is denoted by $[r/x]s$.

A *variable assignment* g maps variables $x \in V$ to elements in D. $g[d/x]$ denotes the assignment that is identical to g, except for variable x, which is now mapped to d.

Regarding the semantics different options have been proposed in the literature. For example, instead of a possible empty set of existing objects E, we could postulate non-emptiness of E. Here we closely follow the notion of free logic as proposed by Scott [18].

Definition 2 (Dual-domain semantics of *FFOL*) A *model structure for FFOL* consists of a triple $\langle D, E, I, \star \rangle$, where D is a non-empty raw domain of objects, $E \subseteq D$ a possible empty set of "existing" resp. "defined" objects, and I an interpretation function mapping 0-ary function symbols (constants) to defined objects $d \in E$, 0-ary predicate symbols (propositions) to *True* or *False*, n-ary function symbols (for $n \geq 1$) to n-ary functions $D \times \ldots \times D \longrightarrow D$ and n-ary predicate symbols (for $n \geq 1$) to n-ary relations $D \times \ldots \times D$. Finally, $\star \in D \setminus E$ is a designated (non-existing/undefined) object.

Given a variable assignment g, we define the interpretation function $\|\cdot\|^{I,g}$ for terms and formulas of *FFOL* as follows:

Terms
1. $\|x\|^{I,g} = g(x)$ for variable symbols $x \in V$
2. $\|c\|^{I,g} = I(c)$, where $c \in F$ is an 0-ary function symbol
3. $\|f(t_1, \ldots, t_n)\|^{I,g} = I(f)(\|t_1\|^{I,g}, \ldots, \|t_n\|^{I,g})$, where $f \in F$ is an n-ary ($n \geq 1$) function symbol
4. $\|\imath x.r\|^{I,g} = d \in E$, such that $\|r\|^{I,g[d/x]} = \mathit{True}$ and $\|r\|^{I,g[d'/x]} = \mathit{False}$ for all $d' \neq d \in E$ (i.e. d is the unique existing object for which r holds); if there is no such $d \in E$, then $\|\imath x.r\|^{I,g} = \star$

Formulas
1. $\|q\|^{I,g} = I(q)$, where $q \in P$ is an 0-ary predicate symbol
2. $\|t_1 = t_2\|^{I,g} = \mathit{True}$ iff $\|t_1\|^{I,g} = \|t_2\|^{I,g}$ (this basic notion of primitive equality on D implies that equations such as $1/0 = 1/0$ are evaluated to *True*; later, in Section **??**, we will define and utilise further notions of equality, including *Kleene equality* and *existing equality*).
3. $\|p(t_1, \ldots, t_n)\|^{I,g} = \mathit{True}$ iff $(\|t_1\|^{I,g}, \ldots, \|t_n\|^{I,g}) \in I(p)$ for n-ary ($n \geq 1$) predicate symbols $p \in P$
4. $\|\neg r\|^{I,g} = \mathit{True}$ iff $\|r\|^{I,g} = \mathit{False}$
5. $\|r \rightarrow s\|^{I,g} = \mathit{True}$ iff $\|r\|^{I,g} = \mathit{False}$ or $\|s\|^{I,g} = \mathit{True}$
6. $\|\forall x.r\|^{I,g} = \mathit{True}$ iff for all $d \in E$ we have $\|r\|^{I,g[d/x]} = \mathit{True}$

Definition 3 A formula s_o is *true* in model M under assignment g iff $\|s_o\|^{M,g} = T$; this is also denoted as $M, g \models s_o$. A formula s_o is called *valid* in M, which is denoted as $M \models s_o$, iff $M, g \models s_o$ for all assignments g. Finally, a formula s_o is called *valid*, which we denote by $\models s_o$, iff s_o is valid for all M. Moreover, we write $\Gamma \models \Delta$ (for sets of formulas Γ and Δ) iff there is a model M and an assignment g such that $M, g \models s_o$ for all $s_o \in \Gamma$ and $M, g \models t_o$ for at least one $t_o \in \Delta$.

2.2 Classical Higher-Order Logic

Simple type theory, also referred to as classical higher-order logic (HOL), is an expressive logic formalism that allows for higher-order quantification, that is quantification over arbitrary set and function variables. It is based on the simply typed λ-calculus[1] and it has its origin in the work by Church [**?**].

Definition 4 (Types) The set $\boldsymbol{T}$ of simple types freely generated from a set of basic types $\{\boldsymbol{o}, \boldsymbol{\mu}, \boldsymbol{\iota}\}$ using the function type constructor $\rightarrow$. $\boldsymbol{o}$ is the type of Booleans, $\boldsymbol{\mu}$ is the type of individuals, and type $\boldsymbol{\iota}$ is employed as the type of

[1] For a detailed discussion of typed λ-calculi, we refer to the literature [**?**].

possible worlds below. We may avoid parentheses if the structure of a complex type is clear in context.

Definition 5 (Syntax of HOL) The language of higher-order logic HOL is defined by the following grammar:[2]

$$\begin{aligned} s,t \quad ::= \quad & p_\alpha \mid X_\alpha \mid (\lambda X_\alpha . s_\beta)_{\alpha\to\beta} \mid (s_{\alpha\to\beta}\, t_\alpha)_\beta \mid \neg_{o\to o}\, s_o \mid \\ & ((\vee_{o\to o\to o}\, s_o)\, t_o) \mid \forall_{(\alpha\to o)\to o}(\lambda X_\alpha . s_o) \mid \imath(\lambda X_\alpha . s_o) \end{aligned}$$

where $\alpha, \beta \in T$. p_α denotes typed constants and X_α typed variables (distinct from p_α). Complex typed terms are constructed via abstraction and application. The type of each term is given as a subscript. Terms s_o of type o are called formulas. The logical connectives of choice are $\neg_{o\to o}$, $\vee_{o\to o\to o}$, $\forall_{(\alpha\to o)\to o}$ (for $\alpha \in T$) and $\imath_{(\alpha\to o)\to\alpha}$ (for $\alpha \in T$). Type subscripts may be dropped if irrelevant or obvious. Similarly, parentheses may be avoided. Binder notation $\forall X_\alpha . s_o$ and $\imath X_\alpha s_o$ is used as shorthand for $\forall_{(\alpha\to o)\to o}(\lambda X_\alpha . s_o)$ and $\imath(\lambda X_\alpha . s_o)$, and infix notation $s \vee t$ is employed instead of $((\vee s)\, t)$. From the above connectives, other logical connectives, such as $\top$, $\bot$, $\wedge$, $\longrightarrow$, $\equiv$ and $\exists$, can be defined in the usual way.

Substitution of a term A_α for a variable X_α in a term B_β is denoted by $[A/X]B$. Since we consider α-conversion implicitly, we assume the bound variables of B avoid variable capture.

Two common relations on terms are given by β-reduction and η-reduction. A β-redex has the form $(\lambda X . s)t$ and β-reduces to $[t/X]s$. An η-redex has the form $(\lambda X . sX)$ where variable X is not free in s; it η-reduces to s. We write $s =_\beta t$ to mean s can be converted to t by a series of β-reductions and expansions. Similarly, $s =_{\beta\eta} t$ means s can be converted to t using both β and η. For each $s_\alpha \in$ HOL there is a unique *β-normal form* and a unique *$\beta\eta$-normal form.*

A *variable assignment* g maps variables X_α to elements in D_α. $g[d/W]$ denotes the assignment that is identical to g, except for variable W, which is now mapped to d.

Definition 6 A *frame* D is a collection $\{D_\alpha\}_{\alpha\in\mathrm{T}}$ of nonempty sets D_α, such that $D_o = \{T, F\}$ (for truth and falsehood). The $D_{\alpha\to\beta}$ are collections of functions mapping D_α into D_β.

Definition 7 A *model* for HOL is a tuple $M = \langle D, I\rangle$, where D is a frame, and I is a family of typed interpretation functions mapping constant symbols p_α to appropriate elements of D_α, called the *denotation of* p_α (the logical connectives $\neg$, $\vee$, and $\forall$ are always given the standard denotations, see below). Moreover, we assume that the domains $D_{\alpha\to\alpha\to o}$ contain the respective identity relations.

Definition 8 The *value* $\|s_\alpha\|^{M,g}$ of a HOL term s_α on a model $M = \langle D, I\rangle$ under assignment g is an element $d \in D_\alpha$ defined in the following way:

[2] Ednote: Maybe better to define this analagous to *FFOL*.

1. $\|p_\alpha\|^{M,g} = I(p_\alpha)$ and $\|X_\alpha\|^{M,g} = g(X_\alpha)$
2. $\|(s_{\alpha\to\beta}\, t_\alpha)_\beta\|^{M,g} = \|s_{\alpha\to\beta}\|^{M,g}(\|t_\alpha\|^{M,g})$
3. $\|(\lambda X_\alpha \text{.} s_\beta)_{\alpha\to\beta}\|^{M,g} =$ the function f from D_α to D_β such that $f(d) = \|s_\beta\|^{M,g[d/X_\alpha]}$ for all $d \in D_\alpha$
4. $\|(\neg_{o\to o}\, s_o)_o\|^{M,g} = T$ iff $\|s_o\|^{M,g} = F$
5. $\|((\vee_{o\to o\to o}\, s_o)\, t_o)_o\|^{M,g} = T$ iff $\|s_o\|^{M,g} = T$ or $\|t_o\|^{M,g} = T$
6. $\|(\forall_{(\alpha\to o)\to o}(\lambda X_\alpha \text{.} s_o))_o\|^{M,g} = T$ iff for all $d \in D_\alpha$ we have $\|s_o\|^{M,g[d/X_\alpha]} = T$
7. $\|(\iota_{(\alpha\to o)\to\alpha}(\lambda X_\alpha \text{.} s_o))_o\|^{M,g} = d$ if there exists a unique $d \in D_\alpha$ such that $\|s_o\|^{M,g[d/X_\alpha]} = T$, otherwise $\|(\iota_{(\alpha\to o)\to\alpha}(\lambda X_\alpha \text{.} s_o))_o\|^{M,g} = e$ for an arbitrary element $e \in D_\alpha$

Definition 9 A model $M = \langle D, I\rangle$ is called a *standard model* iff for all $\alpha, \beta \in T$ we have $D_{\alpha\to\beta} = \{f \mid f : D_\alpha \longrightarrow D_\beta\}$. In a *Henkin model* function spaces are not necessarily full. Instead it is only required that $D_{\alpha\to\beta} \subseteq \{f \mid f : D_\alpha \longrightarrow D_\beta\}$ (for all $\alpha, \beta \in T$) and that the valuation function $\| \cdot \|^{M,g}$ from above is total (i.e., every term denotes). Any standard model is obviously also a Henkin model. We consider Henkin models in the remainder.

Definition 10 A formula s_o is *true* in model M under assignment g iff $\|s_o\|^{M,g} = T$; this is also denoted as $M, g \models^{\mathrm{HOL}} s_o$. A formula s_o is called *valid* in M, which is denoted as $M \models^{\mathrm{HOL}} s_o$, iff $M, g \models^{\mathrm{HOL}} s_o$ for all assignments g. Finally, a formula s_o is called *valid*, which we denote by $\models^{\mathrm{HOL}} s_o$, iff s_o is valid for all M. Moreover, we write $\Gamma \models^{\mathrm{HOL}} \Delta$ (for sets of formulas Γ and Δ) iff there is a model M and an assignment g such that $M, g \models^{\mathrm{HOL}} s_o$ for all $s_o \in \Gamma$ and $M, g \models^{\mathrm{HOL}} t_o$ for at least one $t_o \in \Delta$.

3 Shallow Semantical Embedding of *FFOL* in HOL

The existence predicate E is realized in the embedding as a HOL predicate E of type $i \to o$; we assume that a respective constant symbol is given in the signature of HOL. The raw domain D is simply associated with the set of all objects of type i, respectively with the universal predicate $\lambda X_i.T$. Moreover, let constant symbol $\star$ of type i be in the signature of HOL. We assume that $\|E \star\|^{M,g} = F$ for all M, g i.e. that the element denoted by $\star$ is not an element of the domain of existing objects denoted by E.

Definition 11 (Embedding of *FFOL* in HOL) Given a formula $s \in$ *FFOL*. We map each language construct s in the signature of *FFOL* to a corresponding term $\hat{s}$ of HOL. This mapping is defined as follows:

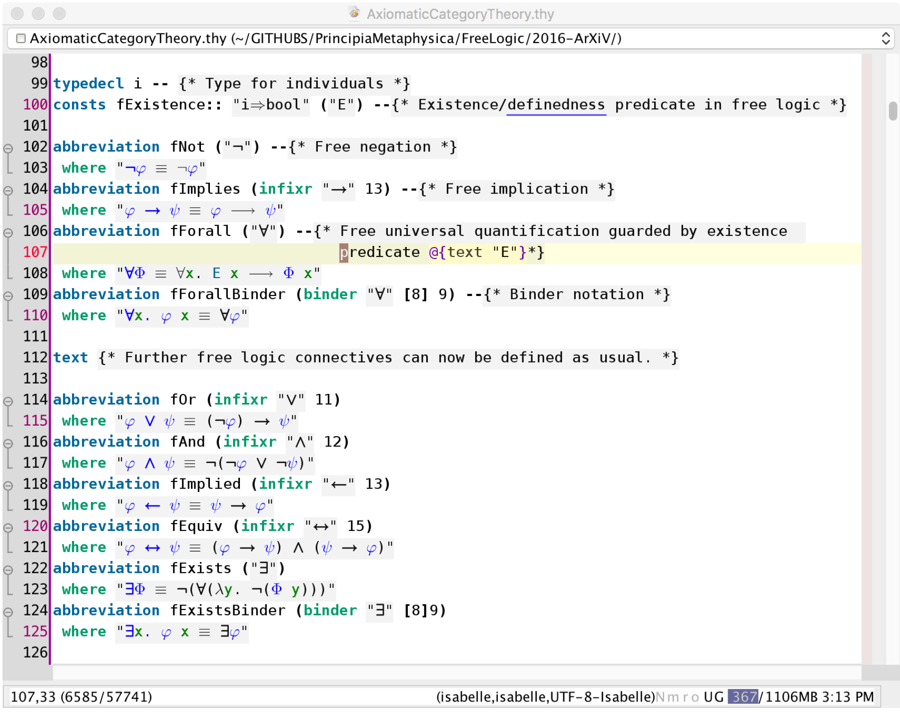

Fig. 2 Isabelle/HOL formalisation of *FFOL* in HOL

$$
\begin{array}{lcll}
\hat{x} & := & \boldsymbol{X_i} & \text{for all } x \in V \\
f(t^1, \hat{\ldots}, t^n) & := & (\boldsymbol{f}\hat{t^1} \ldots \hat{t^n}) & \text{for all } n\text{-ary } f \in F \\
 & & & \boldsymbol{f} \text{ has type } \underbrace{\boldsymbol{i} \to \ldots \to \boldsymbol{i}}_{n \geq o} \to \boldsymbol{i} \\
 & & & \\
s \mathrel{\hat{=}} t & := & \hat{s} = \hat{t} & \\
p(t^1, \hat{\ldots}, t^n) & := & (\boldsymbol{p}\hat{t^1} \ldots \hat{t^n}) & \text{for all } n\text{-ary } p \in P \\
 & & & \boldsymbol{p} \text{ has type } \underbrace{\boldsymbol{i} \to \ldots \to \boldsymbol{i}}_{n \geq o} \to \boldsymbol{o} \\
 & & & \\
\hat{\neg}\varphi & := & \neg\hat{\varphi} & \\
\varphi \mathrel{\hat{\to}} \psi & := & \hat{\varphi} \to \hat{\psi} & \\
\hat{\forall} x.s & := & \forall \boldsymbol{X_i}.\boldsymbol{E}\boldsymbol{X_i} \to \hat{s} & \\
\imath\hat{x}.s & := & \textbf{IfThenElse} & \\
 & & \quad (\exists \boldsymbol{x}.\boldsymbol{E}\boldsymbol{x} \wedge \hat{s} \wedge (\forall \boldsymbol{y}.(\boldsymbol{E}\boldsymbol{y} \wedge ((\boldsymbol{\lambda X_i}.\hat{s})\boldsymbol{y})) \to \boldsymbol{y} = \boldsymbol{x})) & \\
 & & \quad (\imath \boldsymbol{X_i}.\hat{s}) & \\
 & & \quad \star &
\end{array}
$$

where **IfThenElse** is an abbreviation for the term $\boldsymbol{\lambda S_o. \lambda X_i. \lambda Y_i. \imath Z_i.(S_o} \wedge \boldsymbol{Z} = \boldsymbol{X}) \vee (\neg \boldsymbol{S_o} \wedge \boldsymbol{Z} = \boldsymbol{Y})$.

Further connectives can be introduced as usual: $r \vee s := \neg r \to s$, $r \wedge s := \neg(\neg r \vee \neg s)$,

Todo ... the embedding in Isabelle/HOL is depicted in figure 2

4 Exploring Axioms Systems for Category Theory

In an exemplary theory exploration study, we have shown how Scott's [19] axiom system for category theory can be derived from a notion of partial monoids. These axioms systems are presented in Table 1.

The stepwise evolution has been described in [4]. Below we summarise these experiments. However, first we describe some basic modeling decisions for the technical encoding in Isabelle/HOL. The Isabelle/HOL sources of our experiments are available at `todo-sources`. Note that in these sources we did add neither the definite description operator nor the designated undefined object $\star$, since both were not required in our experiments.

4.1 Modeling of basic concepts

Morphisms in the category are modeled as objects of type i. We introduce three partial functions, *dom* (domain), *cod* (codomain), and $\cdot$ (morphism composition). Partiality of composition is handled exactly as expected: we generally may have non-existing compositions $x \cdot y$ (i.e. $\neg(E(x \cdot y))$) for some existing morphisms x and y (i.e. Ex and Ey).

For composition $\cdot$ we assume set-theoretical composition here (i.e., functional composition from right to left). This means that

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

and that

$$(x \cdot y)a \cong x(ya) \quad \text{when} \quad dom\ x \simeq cod\ y$$

The equality symbol $\cong$ denotes Kleene equality and it is defined as follows (where $=$ is identity on all objects, existing or non-existing, of type i):

$$x \cong y := (Ex \vee Ey) \longrightarrow x = y$$

Existing identity $\simeq$ is defined as:

$$x \simeq y := Ex \wedge Ey \wedge x = y$$

$\cong$ is an equivalence relation. $\simeq$, in contrast, is only symmetric and transitive, and lacks reflexivity. These observations are quickly confirmed by Isabelle.

Next, we define the identity morphism predicate I as follows:
abbreviation I where

$$Ii := (\forall x. E(i \cdot x) \longrightarrow i \cdot x \cong x) \wedge (\forall x. E(x \cdot i) \longrightarrow x \cdot i \cong x)$$

This definition was suggested by an exercise in [11] on p. 4. In earlier experiments we used a longer definition which can be proved equivalent on the basis of the other axioms. For monoids, where composition is total, Ii means i is a two-sided identity and such are unique. For categories the property is much weaker.

Axioms Set I

S_i	$E(x \cdot y) \longrightarrow (Ex \wedge Ey)$
E_i	$E(x \cdot y) \longleftarrow (Ex \wedge Ey \wedge (\exists z. z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y))$
A_i	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_i	$\forall y. \exists i. Ii \wedge i \cdot y \cong y$
D_i	$\forall x. \exists j. Ij \wedge x \cdot j \cong x$

Axioms Set II

S_{ii}	$E(x \cdot y) \longrightarrow (Ex \wedge Ey) \wedge (E(dom\ x) \longrightarrow Ex) \wedge (E(cod\ y) \longrightarrow Ey)$
E_{ii}	$E(x \cdot y) \longleftarrow (Ex \wedge Ey \wedge (\exists z. z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y))$
A_{ii}	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_{ii}	$Ey \longrightarrow (I(cod\ y) \wedge (cod\ y) \cdot y \cong y)$
D_{ii}	$Ex \longrightarrow (I(dom\ x) \wedge x \cdot (dom\ x) \cong x)$

Axioms Set III

S_{iii}	$E(x \cdot y) \longrightarrow (Ex \wedge Ey) \wedge (E(dom\ x) \longrightarrow Ex) \wedge (E(cod\ y) \longrightarrow Ey)$
E_{iii}	$E(x \cdot y) \longleftarrow (dom\ x \cong cod\ y \wedge E(cod\ y)))$
A_{iii}	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_{iii}	$Ey \longrightarrow (I(cod\ y) \wedge (cod\ y) \cdot y \cong y)$
D_{iii}	$Ex \longrightarrow (I(dom\ x) \wedge x \cdot (dom\ x) \cong x)$

Axioms Set IV

S_{iv}	$E(x \cdot y) \longrightarrow (Ex \wedge Ey) \wedge (E(dom\ x) \longrightarrow Ex) \wedge (E(cod\ y) \longrightarrow Ey)$
E_{iv}	$E(x \cdot y) \longleftrightarrow (dom\ x \cong cod\ y \wedge E(cod\ y)))$
A_{iv}	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
C_{iv}	$(cod\ y) \cdot y \cong y$
D_{iv}	$x \cdot (dom\ x) \cong x$

Axioms Set V (Scott 79, [19])

$S1$	$E(dom\ x) \longrightarrow Ex$
$S2$	$E(cod\ y) \longrightarrow Ey$
$S3$	$E(x \cdot y) \longleftrightarrow dom\ x \simeq cod\ y$
$S4$	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
$S5$	$(cod\ y) \cdot y \cong y$
$S6$	$x \cdot (dom\ x) \cong x$

Table 1 Stepwise evolution of Scott's [19] axiom system for category theory from partial monoids. The axiom names are motivated as follows: S stands for strictness, E for existence, A for associativity, C for codomain, D for Domain. The free variables x, y, z range over the raw domain D. The quantifiers in Axiom Sets I and II are free logic quantifiers, that is, they range over the domain E of existing objects.

4.2 Consistency

The model finder Nitpick confirms consistency for all of the axiom sets from Table 1. For example, when asked to consider at least one defined and one undefined object, then Nitpick generates for all cases the following model (called M_1 in the remainder): $D = \{i_i, i_2\}$ and $E = \{i_1\}$; $i_1 \cdot i_1$ is i_1, and i_2 in all other cases; *cod* and *dom* are identity on D. Without constraining the request, Nitpick generates an even simpler model (called M_0 in the remainder): $D = \{i_i\}$ and $E = \emptyset$; $i_1 \cdot i_1$ is i_1; *cod* and *dom* are identity on D. It is trivial to check that these models indeed confirm the consistency of all axiom sets from Table 1.

4.3 Axioms Set I

Axioms Set I is our most basic axiom set for category theory generalizing the axioms for a monoid to a partial composition operation. Remember that 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.

Axioms Set I generalises the notion of a monoid by introducing a partial, strict binary composition operation $\cdot$. The existence of left and right identity elements is addressed in the last two axioms. The notions of *dom* (Domain) and *cod* (Codomain) abstract from their common meaning in the context of sets. In category theory we work with just a single type of objects (the type i of morphisms) and therefore identity morphisms are employed to suitably characterize their meanings.

We can prove that the i in axiom C_i and the j in axiom D_i are unique. The proofs and the dependencies can be found automatically by Sledgehammer.

$$\forall y. \exists i. Ii \wedge i \cdot y \cong y \wedge (\forall j. d(Ij \wedge j \cdot y \cong y) \longrightarrow i \cong j) \quad \text{(by } A_i, C_i, S_i)$$

$$\forall x. \exists j. Ij \wedge x \cdot j \cong x \wedge (\forall i. (Ii \wedge x \cdot i \cong x) \longrightarrow j \cong i) \quad \text{(by } A_i, D_i, S_i)$$

However, the i and j need not be equal. Using existential variables C and D, this can be encoded in our formalization as follows:

$$\exists C, D. (\forall y. I(Cy) \wedge (Cy) \cdot y \cong y) \wedge (\forall x. I(Dx) \wedge x \cdot (Dx) \cong x) \wedge D \neq C$$

The model finder Nitpick confirms that this formula is satisfiable: e.g. choose domain $D = \{i_1, i_2\}$ and $E = \{i_2\}$; $i_2 \cdot i_2$ returns i_2, and i_1 in all other cases; variable D is identity on domain D, but C maps both i_1 and i_2 to i_2.

4.4 Axioms Set II

Axioms Set II is developed from Axioms Set I by Skolemization of the existentially quantified variables i and j in axioms C_i and D_i. We can argue semantically that every model of Axioms Set I has such functions. Hence, we get a conservative extension of Axioms Set I. This could be done for any theory with an $\forall x.\exists i.$-axiom. The strictness axiom S is extended, so that strictness is now also postulated for the new Skolem functions *dom* and *cod* . Note that the values of Skolem functions outside E can just be given by the identity function.

The left-to-right direction of existence axiom E_{ii} is implied.

$$E(x \cdot y) \longrightarrow (Ex \wedge Ey \wedge (\exists z. z \cdot z \cong z \wedge x \cdot z \cong x \wedge z \cdot y \cong y)) \quad (\text{by } A_{ii}, C_{ii}, S_{ii})$$

Axioms C_{ii} and D_{ii}, together with S_{ii}, show that *dom* and *cod* are total functions, as intended:

$$Ex \longrightarrow E(dom\ x) \quad (\text{by } D_{ii}, S_{ii})$$

$$Ex \longrightarrow E(cod\ x) \quad (\text{by } C_{ii}, S_{ii})$$

The proofs are found by Sledgehammer and verified in Isabelle/HOL. Using Sledgehammer we have also shown that Axioms Set II implies Axioms Set I. Vice versa, Axioms Set I also implies Axioms Set II. This can easily be shown by semantical means on the meta-level.

4.5 Remark on the Experiments

All proofs above and all proofs in the rest of this paper (unless stated otherwise) have been obtained fully automatically with the Sledgehammer tool in Isabelle/HOL. This tool interfaces to prominent first-order automated theorem provers such as CVC4 [10], Z3 [15], E [17] and Spass [8]. Remotely, also provers such as Vampire [12], or the higher-order provers Satallax [9] and LEO-II [5] can be reached. For example, to prove axiom E_{iii} from Axioms Set II, we have called Sledgehammer on all axioms of Axioms Set II. The provers then, via Sledgehammer, suggested to call trusted/verified tools in Isabelle/HOL with the exactly required dependencies they detected. With the provided dependency information the trusted tools in Isabelle/HOL were then able to reconstruct the external proofs on their own. This way we obtain a verification of our claims in Isabelle/HOL, in which all the proofs have nevertheless been contributed by automated theorem provers.

4.6 Axioms Set III

In Axioms Set III the existence axiom E_{ii} from Axioms Set II is simplified by taking advantage of the two new Skolem functions *dom* and *cod* .

The left-to-right direction of existence axiom E_{iii} is implied.

$$E(x \cdot y) \longrightarrow (dom\ x \cong cod\ y \wedge E(cod\ y)) \quad (\text{by } A_{iii}, C_{iii}, D_{iii}, S_{iii})$$

Axioms Set II and Axioms Set III are equivalent; this has been confirmed by the automated theorem provers and verified in Isabelle/HOL.

4.7 Axioms Set IV

Axioms Set IV simplifies the axioms C_{iii} and D_{iii} . However, as it turned out, these simplifications also require the existence axiom E_{iii} to be strengthened into an equivalence.

Axioms Set III and Axioms Set IV are equivalent; this has been confirmed by the automated theorem provers and verified in Isabelle/HOL.

4.8 Axioms Set V

Axioms Set V has been proposed by Scott [19] in the 1970s. This set of axioms is equivalent to the axiom set presented by Freyd and Scedrov in their textbook “Categories, Allegories” [11], when encoded in free logic, corrected/adapted and further simplified. Their axiom set is technically flawed when encoded in our given context. This issue has been detected by automated theorem provers with the same technical infrastructure as employed so far. See Section5 for more details.

Axioms Set IV and Axioms Set V are equivalent; again, this has been confirmed by the automated theorem provers and verified in Isabelle/HOL.

5 Assessment of the Axiom System by Freyd and Scedrov

In this section we study the axioms set of Freyd and Scedrov from their textbook “Categories, Allegories” [11]. In Subsection 5.1 we show, that their axioms set, replicated in Table 2 as Axioms Set VI, becomes inconsistent in our free logic setting if we assume non-existing objects of type i, respectively, if we assume that the operations are non-total.

Note, however, that the free variables in this first study range over the existing and non-existing objects in domain D. One may argue, that this is not the intention of Freyd and Scedrov. Therefore, we add a second study in Subsection 5.2, in which we restrict the variables to range only over existing objects in E. However, also in this case the axiom system of Freyd and Scedrov remains unsatisfactory. Now it turns out incomplete, since strictness conditions/axioms are required which are not mentioned in the textbook.

Freyd and Scedrov employ a different notation for *dom* x and *cod* x. They denote these operations by $\Box x$ and $x\Box$. Moreover, they employ diagrammatic composition $(f \circ g)x \cong g(fx)$ (functional composition from left to right) instead

Axioms Set FS-I: Freyd and Scedrov in original notation (with issues)

$A1$	$E(x \circ y) \longleftarrow (x\Box \cong \Box y)$
$A2a$	$((\Box x)\Box) \cong \Box x$
$A2b$	$\Box(x\Box) \cong \Box x$
$A3a$	$(\Box x) \circ x \cong x$
$A3b$	$x \circ (x\Box) \cong x$
$A4a$	$\Box(x \circ y) \cong \Box(x \circ (\Box y))$
$A4b$	$(x \circ y)\Box \cong ((x\Box) \circ y)\Box$
$A5$	$x \circ (y \circ z) \cong (x \circ y) \circ z$

Axioms Set FS-II: Freyd and Scedrov in our notation (with issues)

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

Axioms Set VI: Freyd and Scedrov in our notation and corrected

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

Table 2 The axioms set of Freyd and Scedrov in their and our notation, together with a proposed correction.

of the set-theoretic definition $(f \cdot g)x \cong f(gx)$ (functional composition from right to left) used so far. We leave it to the reader to verify that their Axioms

Set VI corresponds to Axioms Set VII modulo an appropriate conversion of notation.[3]

5.1 Constricted Inconsistency in Free Logic Setting

A main difference in the system by Freyd and Scedrov to our Axiom Set V from Table 1 concerns axiom $S3$ respectively $A1$. Namely, instead of the non-reflexive existing identity $\simeq$, they use Kleene equality $\cong$, cf. definition 1.11 on page 3 of [11].[4] The difference seems minor, but in our free logic setting it has the effect to cause the mentioned constricted inconsistency issue. This could perhaps be an oversight, or it could indicate that Freyd and Scedrov actually mean the Axioms Set VIII below (where the variables in the axioms range over defined objects only). However, in Axioms Set VIII we had to (re-)introduce explicit strictness conditions to ensure equivalence to the Axiom Set V by Scott.

The (constricted) inconsistency of Axioms Set FS-I, respectively Axiom Set FS-II, from Table 2 has been detected first by the model finder Nitpick. When we asked Nitpick to generate a model with at least one non-existing object, it claimed that there is no such model. However, a model can still be constructed if we do not make any assumptions about non-existing objects. In fact, the model presented by Nitpick for this case consists of a single, existing morphism.

However, one can see directly that Axiom $A1$ is problematic as written: If x and y are undefined, then (presumably) $dom\ x$ and $cod\ y$ are undefined as well, and by the definition of Kleene equality, $dom\ x \cong cod\ y$. $A1$ stipulates that $x \cdot y$ should be defined in this case, which appears unintended.

As we will demonstrate now, the consequences of this version of the axiom are even stronger. It implies that *all* objects are defined, that is, composition (as well as dom and cod) become total operations. The theory described by these axioms "collapses" to the theory of monoids: If all objects are defined, then one can conclude from $A1$ that $dom\ x \cong dom\ y$ (resp. $dom\ x \cong cod\ y$) and $cod\ x \cong cod\ y$), and according to 1.14 of [11], the category reduces to a monoid provided that it is not empty.

In fact, the automated theorem provers, via Sledgehammer, quickly prove falsity from Axioms Set FS-II (or FS-I) when assuming a non-existing object of type i:

$$(\exists x.\neg Ex) \longrightarrow False$$

The provers identify the axioms $A1$, $A2a$ and $A3a$ to cause the problem under this assumption. A human-intuitive proof argument is as follows:

[3] A recipe for this translation is as follows: (i) replace all $x \circ y$ by $y \cdot x$, (ii) rename the variables to get them again in alphabetical order, (iii) replace $\varphi\Box$ by $cod\ \varphi$ and $\Box\varphi$ by $dom \varphi$, and finally (iv) replace $cod\ y \cong dom\ x$ (resp. $cod\ y \simeq dom\ x$) by $dom\ x \cong cod\ y$ (resp. $dom\ x \simeq cod\ y$).

[4] Def. 1.11 in Freyd Scedrov: "The ordinary equality sign = [i.e., our $\cong$] will be used in the symmetric sense, to wit: if either side is defined then so is the other and they are equal. . . ."

Let $a \in D$ be an undefined object, that is, assume Ea. By instantiating axiom $A3a$ with a we have $a \cdot (dom\ a) \cong a$. From this and definition of $\cong$ we know that $a \cdot (dom\ a)$ is not defined. This is easy to see, since if $a \cdot (dom\ a)$ were defined, we also had that a is defined, which is not the case by assumption. Hence, $\neg E(a \cdot (dom\ a)$. Next, we instantiate $A1$ with a and $dom\ a$ to obtain $E(a \cdot (dom\ a)) \longleftrightarrow dom\ a \cong cod\ (dom\ a)$. Moreover, by instantiating $A2a$ with a we obtain $cod\ (dom\ a) \cong doma$, which we use (modulo symmetry and transitivity of $\cong$) to rewrite the former result into $E(a \cdot (dom\ a)) \longleftrightarrow dom\ a \cong dom\ a$. By reflexivity of $\cong$ we thus get $E(a \cdot (dom\ a))$, i.e. that $a \cdot (dom\ a)$ is defined, which contradicts $\neg E(a \cdot (dom\ a)$. $\Box$

As a corollary from the above constricted inconsistency result we get that all must be defined: $\forall x.Ex$

5.2 Missing Strictness Axioms in Alternative Setting

6 Conclusion

. . . todo

Acknowledgements DFG projects . . . todo

References

1. Benzmüller, C., Miller, D.: Automation of higher-order logic. In: J. Siekmann, D. Gabbay, J. Woods (eds.) Handbook of the History of Logic, Volume 9 — Logic and Computation. Elsevier (2014)
2. Benzmüller, C., Scott, D.: Automating free logic in Isabelle/HOL. In: G.M. Greuel, T. Koch, P. Paule, A. Sommese (eds.) Mathematical Software – ICMS 2016, 5th International Congress, Proceedings, *LNCS*, vol. 9725, pp. 43–50. Springer, Berlin, Germany (2016). DOI 10.1007/978-3-319-42432-3_6. URL `http://christoph-benzmueller.de/papers/C57.pdf`
3. Benzmüller, C., Scott, D.S.: Axiomatizing category theory in free logic. CoRR **abs/1609.01493** (2016). URL `http://arxiv.org/abs/1609.01493`
4. Benzmüller, C., Scott, D.S.: Axiomatizing category theory in free logic. arXiv, `http://arxiv.org/abs/1609.01493` (2016). URL `http://arxiv.org/abs/1609.01493`
5. Benzmüller, C., Sultana, N., Paulson, L.C., Theiss, F.: The higher-order prover Leo-II. J. of Automated Reasoning **55**(4), 389–404 (2015). DOI 10.1007/s10817-015-9348-y. URL `http://dx.doi.org/10.1007/s10817-015-9348-y`
6. Blanchette, J.C., Böhme, S., Paulson, L.C.: Extending Sledgehammer with SMT solvers. Journal of Automated Reasoning **51**(1), 109–128 (2013). DOI 10.1007/s10817-013-9278-5. URL `http://dx.doi.org/10.1007/s10817-013-9278-5`
7. Blanchette, J.C., Nipkow, T.: Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In: ITP 2010, no. 6172 in LNCS, pp. 131–146. Springer (2010)
8. Blanchette, J.C., Popescu, A., Wand, D., Weidenbach, C.: More SPASS with Isabelle - Superposition with Hard Sorts and Configurable Simplification. In: Interactive Theorem Proving - Third International Conference, ITP 2012, Princeton, NJ, USA, August 13-15, 2012. Proceedings, *Lecture Notes in Computer Science*, vol. 7406, pp. 345–360. Springer (2012). DOI 10.1007/978-3-642-32347-8. URL `http://dx.doi.org/10.1007/978-3-642-32347-8`

Freyd and Scedrov in our notation (corrected and reduced I)

$B1'$	$E(x \cdot y) \longleftrightarrow \mathit{dom}\ x \simeq \mathit{cod}\ y$
$B3a$	$x \cdot (\mathit{dom}\ x) \cong x$
$B3b$	$(\mathit{cod}\ y) \cdot y \cong y$
$B4a$	$\mathit{dom}\ (x \cdot y) \cong \mathit{dom}\ ((\mathit{dom}\ x) \cdot y)$
$B4b$	$\mathit{cod}\ (x \cdot y) \cong \mathit{cod}\ (x \cdot (\mathit{cod}\ y))$
$B5$	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$

Freyd and Scedrov in our notation (corrected and reduced II)

$B1'$	$E(x \cdot y) \longleftrightarrow \mathit{dom}\ x \simeq \mathit{cod}\ y$
$B2a$	$\mathit{cod}\ (\mathit{dom}\ x) \cong \mathit{dom}\ x$
$B2b$	$\mathit{dom}\ (\mathit{cod}\ y) \cong \mathit{cody}$
$B3a$	$x \cdot (\mathit{dom}\ x) \cong x$
$B3b$	$(\mathit{cod}\ y) \cdot y \cong y$
$B5$	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$

Freyd and Scedrov in our notation (corrected and reduced III)

S_v^1	$E(\mathit{dom}\ x) \longrightarrow Ex$
S_v^2	$E(\mathit{cod}\ y) \longrightarrow Ey$
$B1'$	$E(x \cdot y) \longleftrightarrow \mathit{dom}\ x \simeq \mathit{cod}\ y$
$B3a$	$x \cdot (\mathit{dom}\ x) \cong x$
$B3b$	$(\mathit{cod}\ y) \cdot y \cong y$
$B5$	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$

Table 3 Bla

9. Brown, C.E.: Satallax: An automatic higher-order prover. In: Automated Reasoning - 6th International Joint Conference, IJCAR 2012, Manchester, UK, June 26-29, 2012. Proceedings, *Lecture Notes in Computer Science*, vol. 7364, pp. 111–117. Springer (2012). DOI 10.1007/978-3-642-31365-3. URL `http://dx.doi.org/10.1007/978-3-642-31365-3`
10. Deters, M., Reynolds, A., King, T., Barrett, C.W., Tinelli, C.: A tour of CVC4: how it works, and how to use it. In: Formal Methods in Computer-Aided Design, FMCAD 2014, Lausanne, Switzerland, October 21-24, 2014, p. 7. IEEE (2014). URL `http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6975680`
11. Freyd, P., Scedrov, A.: Categories, Allegories. North Holland (1990)
12. Kovács, L., Voronkov, A.: First-Order Theorem Proving and Vampire. In: Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings, *Lecture Notes in Computer Science*, vol. 8044, pp. 1–35. Springer (2013). DOI 10.1007/978-3-642-39799-8. URL `http://dx.doi.org/10.1007/978-3-642-39799-8`

13. Lambert, K.: The definition of e(xistence)! in free logic. In: Abstracts: The International Congress for Logic, Methodology and Philosophy of Science. Stanford: Stanford University Press (1960)
14. Lambert, K.: Free Logic: Selected Essays. Cambridge: Cambridge University Press (2002)
15. de Moura, L.M., Bjørner, N.: Z3: An Efficient SMT Solver. In: Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings, *Lecture Notes in Computer Science*, vol. 4963, pp. 337–340. Springer (2008). DOI 10.1007/978-3-540-78800-3. URL `http://dx.doi.org/10.1007/978-3-540-78800-3`
16. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL: A Proof Assistant for Higher-Order Logic. No. 2283 in LNCS. Springer (2002)
17. Schulz, S.: System description: E 1.8. In: Logic for Programming, Artificial Intelligence, and Reasoning - 19th International Conference, LPAR-19, Stellenbosch, South Africa, December 14-19, 2013. Proceedings, *Lecture Notes in Computer Science*, vol. 8312, pp. 735–743. Springer (2013). DOI 10.1007/978-3-642-45221-5. URL `http://dx.doi.org/10.1007/978-3-642-45221-5`
18. Scott, D.: Existence and description in formal logic. In: R. Schoenman (ed.) Bertrand Russell: Philosopher of the Century, pp. 181–200. George Allen & Unwin, London (1967). (Reprinted with additions in: Philosophical Application of Free Logic, edited by K. Lambert. Oxford Universitry Press, 1991, pp. 28 - 48)
19. Scott, D.: Identity and existence in intuitionistic logic. In: M. Fourman, C. Mulvey, D. Scott (eds.) Applications of Sheaves: Proceedings of the Research Symposium on Applications of Sheaf Theory to Logic, Algebra, and Analysis, Durham, July 9–21, 1977, *Lecture Notes in Mathematics*, vol. 752, pp. 660–696. Springer Berlin Heidelberg (1979)