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

Exploring Axiom Systems for Category Theory

Utilizing a Novel Approach to Automate Free Logic in HOL

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 claim any novel results in category theory, but to demonstrate an elegant way to "implement" and utilise 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.

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 [4]. 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 [2], 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 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" [3], 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 @cite "Scott79" 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 @cite "Isabelle" via the Sledgehammer tool @cite "Sledgehammer". Moreover, we have obtained very useful feedback at various stages from the model finder Nitpick @cite "Nitpick" 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 Preliminaries

2.1 Free Logic

Free logic [**?**,**?**] 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 $\mathbf{D}$, and that $\mathbf{D}$ 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 $\mathbf{D}$, but a (possibly empty) set $\mathbf{E} \subseteq \mathbf{D}$ is chosen to characterize the subdomain of "existing" resp. "defined" objects in $\mathbf{D}$. Quantification is now restricted to set $\mathbf{E}$ 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 $\mathbf{D} \setminus \mathbf{E}$ lying outside of the scope of quantification.

The particular notion of free logic as exploited in the remainder of this article has been introduced by Scott [**?**]. 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*,
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*,

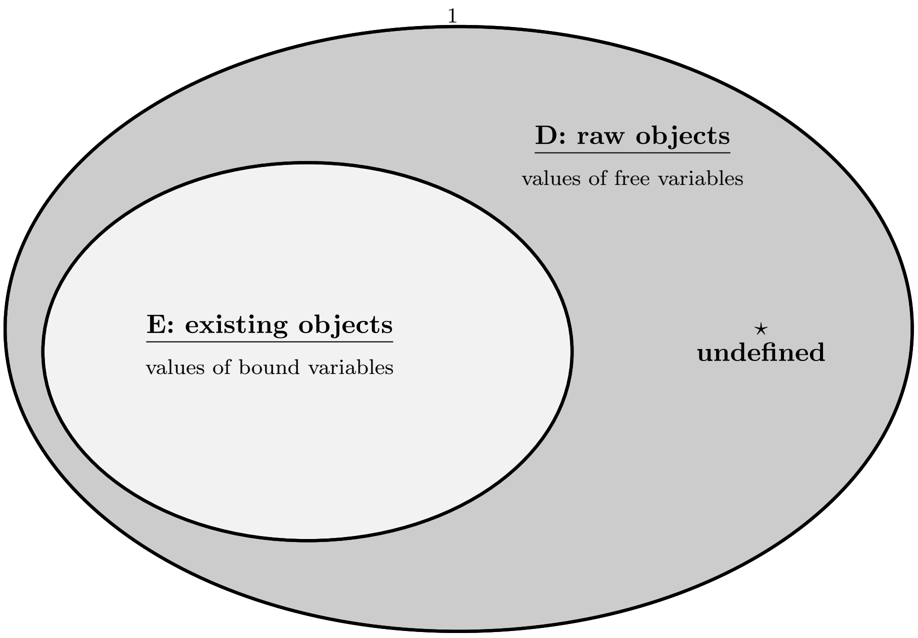


Fig. 1 Illustration of the Semantical Domains of Free Logic

5. given formulas r and s of *FFOL*, then $\neg r$, $r \to s$ 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* can be defined as usual.

Regarding the semantics different options may be considered. For example, instead of a possible empty set of existing objects E, we could postulate non-emptiness of E. We e.g. also have a choice when interpreting equations $t = t$ for undefined terms t; such equations could be mapped to *True* or *False*. Different combinations of such choice points have been studied in the literature. Here we closely follow the logic of Scott [**?**].

Definition 2 (Dual-domain (positive) 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 $g : V \longrightarrow D$, 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 $x \in F$
2. $\|c\|^{I,g} = I(c)$, where $c \in F$ is an 0-ary function symbol
3. $\|f(t_1,\dots,t_n)\|^{I,g} = I(f)(\|t_1\|^{I,g},\dots,\|t_n\|^{I,g})$, 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. $\|p\|^{I,g} = I(p)$, where $p \in F$ 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 implies that equations such as $1/0 = 1/0$ are evaluated to *True*; however, we have alternative options do define this case: e.g. we could additionally require that $\|t_1\|^{I,g} \in E$ and $\|t_2\|^{I,g} \in E$, in which case $1/0 = 1/0$ would be *False*).
3. $\|p(t_1,\dots,t_n)\|^{I,g} = \mathit{True}$ iff $(\|t_1\|^{I,g},\dots,\|t_n\|^{I,g}) \in I(p)$ 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 $\|r\|^{I,g[d/x]} = \mathit{True}$ (here $g[d/x]$ denotes the variable assignment that is identical to g, except for variable x, which is now mapped to object d)

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 $\mathit{\Gamma} \models \mathit{\Delta}$ (for sets of formulas $\mathit{\Gamma}$ and $\mathit{\Delta}$) iff there is a model M and an assignment g such that $M, g \models s_o$ for all $s_o \in \mathit{\Gamma}$ and $M, g \models t_o$ for at least one $t_o \in \mathit{\Delta}$.

2.2 Classical higher-order logic

HOL is easily obtained from HOML by removing the modal operator $\Box$ from the grammar, and by dropping the set of possible worlds W and the accessibility relation R from the definition of a model. Nevertheless, we explicitly state the most relevant definitions for the particular notion of HOL as employed in this paper. One reason is that we do want to carefully distinguish the HOL and HOML languages in the remainder (we use boldface fonts for HOL and standard fonts for HOML). There is also a subtle, but harmless, difference in the HOL language as employed here in comparison to the standard presentation: here three base types are employed, whereas usually only two base types are considered. The third base type plays a crucial role in our embedding of HOML in HOL.

Definition 4 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 possible worlds below. As before we may avoid parentheses.

Definition 5 The grammar for higher-order logic HOL is:

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

where $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \boldsymbol{T}$. The text from Def. **??** analogously applies, except that we do not consider the modal connectives $\Box$ and $\Diamond$.

The definitions for substitution (Def. **??**), β- and η-reduction (Def. **??**), frame (Def. **??**), and assignment (Def. **??**) remain unchanged.

Definition 6 A *model* for HOL is a tuple $\boldsymbol{M} = \langle \boldsymbol{D}, \boldsymbol{I} \rangle$, where $\boldsymbol{D}$ is a frame, and $\boldsymbol{I}$ is a family of typed interpretation functions mapping constant symbols $\boldsymbol{p_\alpha}$ to appropriate elements of $\boldsymbol{D_\alpha}$, called the *denotation of* $\boldsymbol{p_\alpha}$ (the logical connectives $\neg$, $\vee$, and $\forall$ are always given the standard denotations, see below). Moreover, we assume that the domains $\boldsymbol{D_{\alpha\to\alpha\to o}}$ contain the respective identity relations.

Definition 7 The *value* $\|\boldsymbol{s_\alpha}\|^{\boldsymbol{M,g}}$ of a HOL term $\boldsymbol{s_\alpha}$ on a model $\boldsymbol{M} = \langle \boldsymbol{D}, \boldsymbol{I} \rangle$ under assignment $\boldsymbol{g}$ is an element $\boldsymbol{d} \in \boldsymbol{D_\alpha}$ defined in the following way:

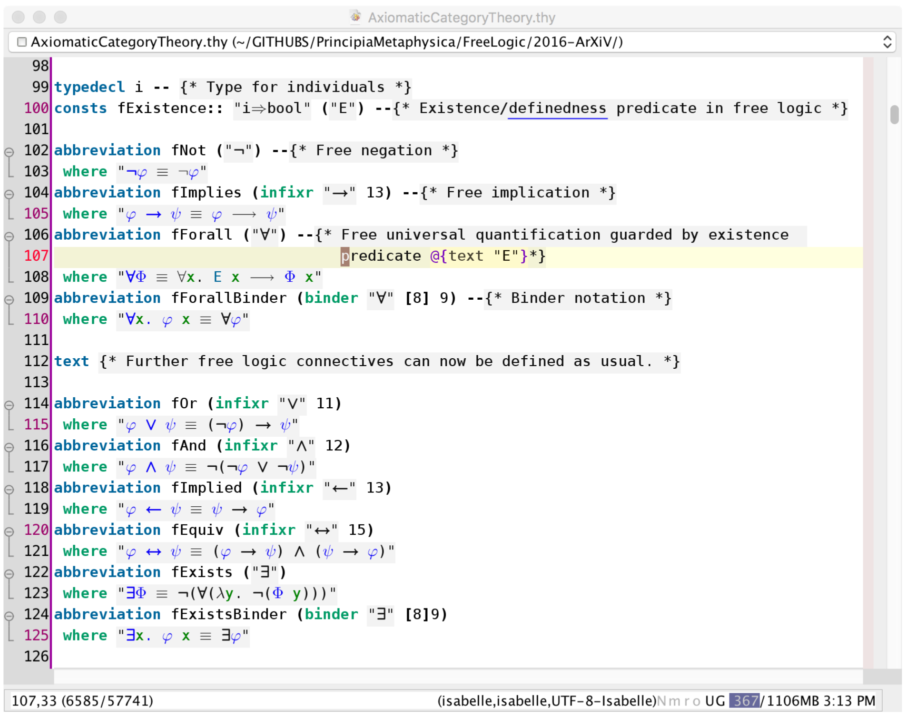
1. $\|\boldsymbol{p_\alpha}\|^{\boldsymbol{M,g}} = \boldsymbol{I(p_\alpha)}$ and $\|\boldsymbol{X_\alpha}\|^{\boldsymbol{M,g}} = \boldsymbol{g(X_\alpha)}$
2. $\|(\boldsymbol{s_{\alpha\to\beta}\, t_\alpha})_{\boldsymbol{\beta}}\|^{\boldsymbol{M,g}} = \|\boldsymbol{s_{\alpha\to\beta}}\|^{\boldsymbol{M,g}}(\|\boldsymbol{t_\alpha}\|^{\boldsymbol{M,g}})$
3. $\|(\boldsymbol{\lambda X_\alpha . s_\beta})_{\boldsymbol{\alpha\to\beta}}\|^{\boldsymbol{M,g}}$ = the function $\boldsymbol{f}$ from $\boldsymbol{D_\alpha}$ to $\boldsymbol{D_\beta}$ such that $\boldsymbol{f(d)} = \|\boldsymbol{s_\beta}\|^{\boldsymbol{M,g[d/X_\alpha]}}$ for all $\boldsymbol{d} \in \boldsymbol{D_\alpha}$
4. $\|(\neg_{\boldsymbol{o\to o}}\, \boldsymbol{s_o})_{\boldsymbol{o}}\|^{\boldsymbol{M,g}} = \boldsymbol{T}$ iff $\|\boldsymbol{s_o}\|^{\boldsymbol{M,g}} = \boldsymbol{F}$
5. $\|((\vee_{\boldsymbol{o\to o\to o}}\, \boldsymbol{s_o})\, \boldsymbol{t_o})_{\boldsymbol{o}}\|^{\boldsymbol{M,g}} = \boldsymbol{T}$ iff $\|\boldsymbol{s_o}\|^{\boldsymbol{M,g}} = \boldsymbol{T}$ or $\|\boldsymbol{t_o}\|^{\boldsymbol{M,g}} = \boldsymbol{T}$
6. $\|(\forall_{\boldsymbol{(\alpha\to o)\to o}}(\boldsymbol{\lambda X_\alpha . s_o}))_{\boldsymbol{o}}\|^{\boldsymbol{M,g}} = \boldsymbol{T}$ iff for all $\boldsymbol{d} \in \boldsymbol{D_\alpha}$ we have $\|\boldsymbol{s_o}\|^{\boldsymbol{M,g[d/X_\alpha]}} = \boldsymbol{T}$

Definition 8 A model $\boldsymbol{M} = \langle \boldsymbol{D}, \boldsymbol{I} \rangle$ 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 9 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 $t_o \in \Delta$.

3 Sound and Complete Embedding in HOL

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



AxiomaticCategoryTheory.thy

AxiomaticCategoryTheory.thy (~/GITHUBS/PrincipiaMetaphysica/FreeLogic/2016-ArXiV/)

```

typedecl i -- {* Type for individuals *}
consts fExistence:: "i⇒bool" ("E") --{* Existence/definedness predicate in free logic *}

abbreviation fNot ("¬") --{* Free negation *}
 where "¬φ ≡ ¬φ"
abbreviation fImplies (infixr "→" 13) --{* Free implication *}
 where "φ → ψ ≡ φ ⟶ ψ"
abbreviation fForall ("∀") --{* Free universal quantification guarded by existence
                                 predicate @{text "E"}*}
 where "∀Φ ≡ ∀x. E x ⟶ Φ x"
abbreviation fForallBinder (binder "∀" [8] 9) --{* Binder notation *}
 where "∀x. φ x ≡ ∀φ"

text {* Further free logic connectives can now be defined as usual. *}

abbreviation fOr (infixr "∨" 11)
 where "φ ∨ ψ ≡ (¬φ) → ψ"
abbreviation fAnd (infixr "∧" 12)
 where "φ ∧ ψ ≡ ¬(¬φ ∨ ¬ψ)"
abbreviation fImplied (infixr "←" 13)
 where "φ ← ψ ≡ ψ → φ"
abbreviation fEquiv (infixr "↔" 15)
 where "φ ↔ ψ ≡ (φ → ψ) ∧ (ψ → φ)"
abbreviation fExists ("∃")
 where "∃Φ ≡ ¬(∀(λy. ¬(Φ y)))"
abbreviation fExistsBinder (binder "∃" [8]9)
 where "∃x. φ x ≡ ∃φ"

```

107,33 (6585/57741) (isabelle,isabelle,UTF-8-Isabelle)Nmro UG 367/1106MB 3:13 PM

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

Table 1 Please write your table caption here

first	second	third
number	number	number
number	number	number

4 Section title

Text with citations [**?**] and [**?**].

4.1 Subsection title

as required. Don't forget to give each section and subsection a unique label (see Sect. 4).

Paragraph headings Use paragraph headings as needed.

$$a^2 + b^2 = c^2 \tag{1}$$

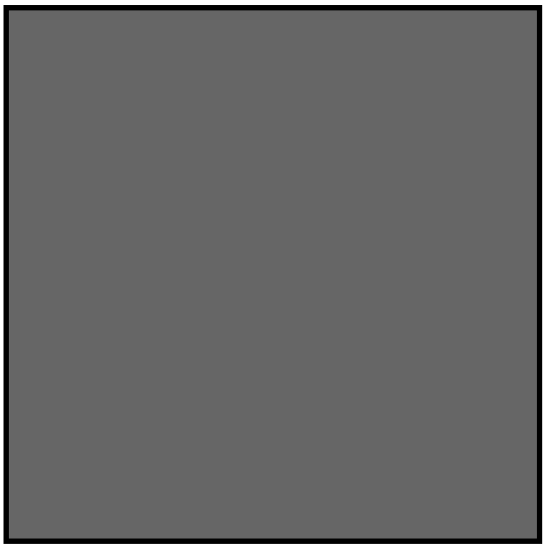


Fig. 3 Please write your figure caption here

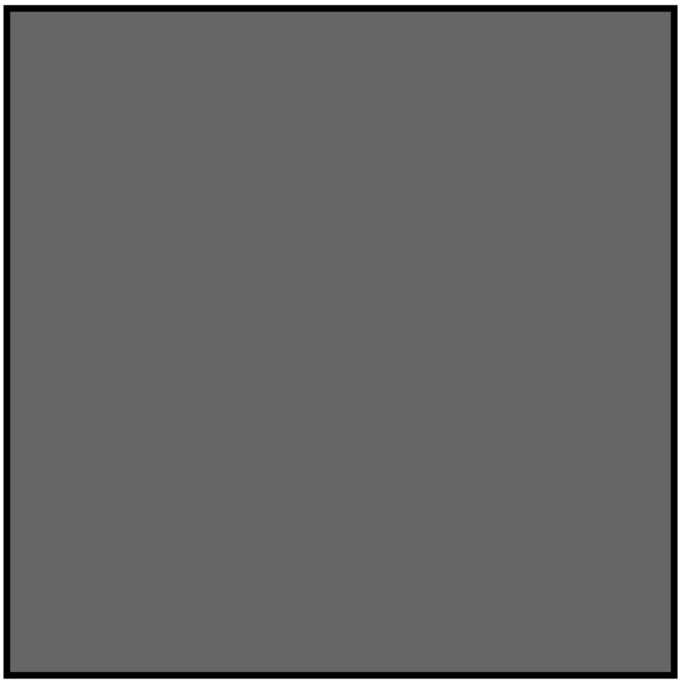


Fig. 4 Please write your figure caption here

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. 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
3. Freyd, P., Scedrov, A.: Categories, Allegories. North Holland (1990)
4. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL: A Proof Assistant for Higher-Order Logic. No. 2283 in LNCS. Springer (2002)