Journal of Automated Reasoning
https://doi.org/10.1007/s10817-018-09507-7

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

Christoph Benzmüller[1,2] · Dana S. Scott[3]

Received: 15 October 2018 / Accepted: 3 December 2018
© Springer Nature B.V. 2018

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 for the formalisation and verification of free logic theories. Subsequently, this approach is applied to 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) issues in a prominent category theory textbook have 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 utilize interactive and automated reasoning in free logic, and to present illustrative 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

Benzmüller received funding from the German National Research Foundation DFG under Heisenberg grant *Towards Computational Metaphysics* (BE 2501/9-2) and from VolkswagenStiftung under grant *Consistent Rational Argumentation in Politics* (CRAP).

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10817-018-09507-7) contains supplementary material, which is available to authorized users.

✉ Christoph Benzmüller
c.benzmueller@gmail.com

Dana S. Scott
dana.scott@cs.cmu.edu

1 Freie Universität Berlin, Berlin, Germany

2 University of Luxembourg, Esch-sur-Alzette, Luxembourg

3 University of California, Berkeley, USA

Revised Proof

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

theorem provers based on traditional classical or intuitionistic logics provide rather inadequate support for these challenge concepts. Free logic [25,26,30,32] offers a theoretically appealing solution, but it has been considered as rather unsuited towards practical utilization.

In the first part of this article (Sects. 2, 3) we show how free logic can be elegantly "implemented" in any theorem proving system for classical higher-order logic (HOL) [8]. The proposed solution employs a semantic embedding of free logic in HOL. We present, as an example, one implementation of this idea in the proof assistant Isabelle/HOL [29]. 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 [15], so that our solution can be utilized, via Isabelle as foreground system, with a whole range of other background reasoners, such as SMT solvers and first-order and higher-order automated theorem provers.[1] 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 Sect. 4 a stepwise development of axioms 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 such as Isabelle/HOL. We also address the relation of our axiom systems to alternative proposals from the literature, including an axiom set proposed by Freyd and Scedrov in their textbook "Categories, Allegories" [23] for which we reveal a technical flaw: either all operations, e.g. morphism composition, are total or their axiom system is inconsistent. The repair for this problem is quite straightforward, however. The solution essentially corresponds to a set of axioms proposed by Scott [33] in the 1970s.

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 [16], 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 the axiom system by Freyd and Scedrov was discovered by automated theorem provers, which further emphasizes the added value of automated theorem proving in this area.

The content of article is based on the contributions reported in two previous papers [9,10].

2 Preliminaries

2.1 Free Logic

Free logic (respectively inclusive logic) [25,26,30,32] refers to a class of logic formalisms that are free of basic existence assumptions regarding the denotation of terms.[2] Remember

[1] Cf. Sect. 4.4 for further information.

[2] Calculi for free logic are presented in [30]; see also the references therein.

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

Revised Proof

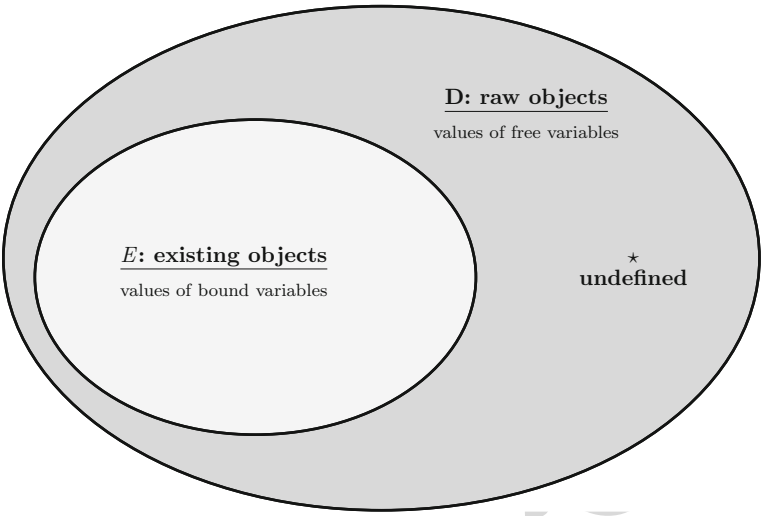

Fig. 1 Illustration of the semantical domains of free logic

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 assumptions are abolished. Terms do still denote objects in a (non-empty) domain D, but a (possibly empty) set $E \subset D$ 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 $D \backslash E$ lying outside of the scope of quantification. Moreover, a function f is *total* if and only if for all x we have $Ex \longrightarrow E(fx)$.[3] For *partial* functions f we may have some x such that Ex but not $E(fx)$. A function f is called *strict* if and only if for all $x \in D$ we have $E(fx) \longrightarrow Ex$.

The particular version of free logic as exploited in the remainder of this article was proposed by Scott [32]. A graphical illustration of this notion of free logic is presented in Fig. 1. It employs a distinguished undefined object $\star$.[4]

We next formally introduce the syntax and semantics of free logic as to be used in the remainder of this article. We refer to this logic as 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,

[3] The predication Ex represents that x is a member of E.

[4] The $\star$ symbol is not to be confused with any other symbol in Isabelle/HOL.

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

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 $\forall x_{\bullet} r$ are (compound) formulas of FFOL, and
6. given a formula r of FFOL, then $\imath x_{\bullet} r$ is a term of FFOL (definite description).

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

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. In fact, our approach below can be easily adapted for different variants of free and inclusive logic. Here we closely follow the notion of free logic as proposed by Scott [32].

Definition 2 (*Model of FFOL*) A *model (structure) for* FFOL consists of a quadruple $M = \langle D, E, I, \star \rangle$, where D is a non-empty raw domain of objects, $E \subset D$ a possible empty set of existing/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 $D \times \cdots \times D \longrightarrow D$ and n-ary predicate symbols (for $n \geq 1$) to n-ary relations $D \times \cdots \times D$. *True* or *False* denote truth and falsehood respectively. Finally, $\star \in D \backslash E$ is a designated (non-existing/undefined) object.

Definition 3 (*Evaluation function for FFOL*) The *value* $\|s\|^{M,g}$ of a term or formula $s \in$ FFOL in a model $M = \langle D, E, I, \star \rangle$ under assignment g defined in the following way:
Terms

1. $\|x\|^{M,g} = g(x)$ for variable symbols $x \in V$
2. $\|c\|^{M,g} = I(c)$, where $c \in F$ is an 0-ary function symbol
3. $\|f(t_1, \ldots, t_n)\|^{M,g} = I(f)(\|t_1\|^{M,g}, \ldots, \|t_n\|^{M,g})$, where $f \in F$ is an n-ary ($n \geq 1$) function symbol
4. $\|\imath x_{\bullet} r\|^{M,g} = d \in E$, such that $\|r\|^{M,g[d/x]} = \mathit{True}$ and $\|r\|^{M,g[d'/x]} = \mathit{False}$ 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 $\|\imath x_{\bullet} r\|^{M,g} = \star$

Formulas

5. $\|q\|^{M,g} = I(q)$, where $q \in P$ is an 0-ary predicate symbol
6. $\|t_1 = t_2\|^{M,g} = \mathit{True}$ if and only if $\|t_1\|^{M,g} = \|t_2\|^{M,g}$ (this basic notion of primitive equality on D implies that equations between "undefined" terms such as $1/0 = 1/0$ are evaluated to *True*; later, in Sect. 4, we will define and utilize further notions of equality, including *Kleene equality* and *existing equality*, which behave differently)
7. $\|p(t_1, \ldots, t_n)\|^{M,g} = \mathit{True}$ if and only if $(\|t_1\|^{M,g}, \ldots, \|t_n\|^{M,g}) \in I(p)$ for n-ary ($n \geq 1$) predicate symbols $p \in P$
8. $\|\neg r\|^{M,g} = \mathit{True}$ if and only if $\|r\|^{M,g} = \mathit{False}$
9. $\|r \rightarrow s\|^{M,g} = \mathit{True}$ if and only if $\|r\|^{M,g} = \mathit{False}$ or $\|s\|^{M,g} = \mathit{True}$
10. $\|\forall x_{\bullet} r\|^{M,g} = \mathit{True}$ if and only if for all $d \in E$ we have $\|r\|^{M,g[d/x]} = \mathit{True}$

Definition 4 (*Validity*) A formula s_o is *true* in model M under assignment g if and only if $\|s_o\|^{M,g} = True$; this is also denoted as $M, g \models^{\mathrm{FFOL}} s_o$. A formula s_o is called *valid* in M, which is denoted as $M \models^{\mathrm{FFOL}} s_o$, if and only if $M, g \models^{\mathrm{FFOL}} s_o$ for all assignments g. Finally, a formula s_o is called *valid*, which we denote by $\models^{\mathrm{FFOL}} s_o$, if and only if s_o is valid for all M.

Revised Proof

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

2.2 Classical Higher-Order Logic

Simple type theory, also referred to as classical higher-order logic (HOL) [2], is an expressive logic formalism which is based on the simply typed λ-calculus [3]. HOL has its origin in the work of Church [19].

For a detailed discussion of the syntax, semantics and automation of HOL we refer to the literature (see e.g. [2,6,8] and the references therein). Below we introduce a variant of HOL with primitive equality and definite descriptions.

Definition 5 (*Types*) The set $\boldsymbol{T}$ of simple types is freely generated from a set of basic types $\{o, i\}$ using the function type constructor $\rightarrow$. $\boldsymbol{o}$ is the type of Booleans and $\boldsymbol{i}$ is the type of individuals. We may avoid parentheses if the structure of a complex type is clear in context.

Definition 6 (*Syntax of HOL*) The terms of HOL with primitive equality and definite description are defined by the following grammar:[5]

$$s, t \ ::= \ p_\alpha \mid X_\alpha \mid (\lambda X_\alpha . s_\beta)_{\alpha\rightarrow\beta} \mid (s_{\alpha\rightarrow\beta}\, t_\alpha)_\beta \mid s_\alpha = t_\alpha \mid \neg_{o\rightarrow o}\, s_o \mid$$
$$((\vee_{o\rightarrow o\rightarrow o}\, s_o)\, t_o) \mid \forall_{(\alpha\rightarrow o)\rightarrow o}(\lambda X_\alpha . s_o) \mid \imath_{(\alpha\rightarrow o)\rightarrow\alpha}(\lambda X_\alpha . s_o)$$

where $\alpha, \beta \in \boldsymbol{T}$. 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 $\neg_{o\rightarrow o}$, $\vee_{o\rightarrow o\rightarrow o}$, $=_{\alpha\rightarrow\alpha\rightarrow o}$, $\forall_{(\alpha\rightarrow o)\rightarrow o}$ and $\imath_{(\alpha\rightarrow o)\rightarrow\alpha}$ (where $\alpha \in \boldsymbol{T}$). 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 $\forall(\lambda X_\alpha . s_o)$ 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 $\top$, $\bot$, $\wedge$, $\rightarrow$, $\equiv$ and $\exists$, can be defined in the usual way. For example, $\rightarrow$ can be defined as $\lambda X_o . \lambda Y_o . \neg X \vee Y$ and $\wedge$ as $\lambda X_o . \lambda Y_o . \neg(\neg X \vee \neg Y)$.

We assume familiarity with λ-conversion (e.g. *α-renaming* and *βη-reduction*) and *βη-normal forms* [3].

A *variable assignment* $\boldsymbol{g}$ maps variables X_α to elements in $\boldsymbol{D}_\alpha$. $\boldsymbol{g}[\boldsymbol{d}/\boldsymbol{W}]$ denotes the assignment that is identical to $\boldsymbol{g}$, except for variable $\boldsymbol{W}$, which is now mapped to $\boldsymbol{d}$.

Definition 7 (*Frame for HOL*) A *frame* $\boldsymbol{D}$ is a collection $\{\boldsymbol{D}_\alpha\}_{\alpha\in\mathrm{T}}$ of nonempty sets $\boldsymbol{D}_\alpha$, such that $\boldsymbol{D}_o = \{True, False\}$ (where, as before, *True* and *False* denote truth and falsehood). The $\boldsymbol{D}_{\alpha\rightarrow\beta}$ are collections of functions mapping $\boldsymbol{D}_\alpha$ into $\boldsymbol{D}_\beta$.

Definition 8 (*Model for HOL*) A *model (structure)* 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 p_α to appropriate elements of $\boldsymbol{D}_\alpha$, 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 $\boldsymbol{D}_{\alpha\rightarrow\alpha\rightarrow o}$ contain the respective identity relations.

Definition 9 (*Evaluation function for HOL*) The *value* $\|s_\alpha\|^{M,g}$ of a HOL term s_α 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. $\|p_\alpha\|^{M,g} = \boldsymbol{I}(p_\alpha)$

[5] It is well known that we could work with a much smaller set of logical connectives, see e.g. Sect. 1.4 of Andrews's overview article [2]. The choice here closely reflects the set of primitive connectives as chosen in higher-order automated theorem provers such as LEO-II [13], Leo-III [12], and Satallax [18].

Revised Proof

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

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

Definition 10 (*Standard and Henkin models*) A model $M = \langle D, I\rangle$ is called a *standard model* if and only if 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. For more details on Henkin semantics, its proof theory and examples of sound and complete calculi we refer to the literature (e.g. [6,7]).

Definition 11 (*Validity*) A formula s_o is *true* in model M under assignment g if and only if $\|s_o\|^{M,g} = True$; this is also denoted as $M, g \models^{\text{HOL}} s_o$. A formula s_o is called *valid* in M, which is denoted as $M \models^{\text{HOL}} s_o$, if and only if $M, g \models^{\text{HOL}} s_o$ for all assignments g. Finally, a formula s_o is called *valid*, which we denote by $\models^{\text{HOL}} s_o$, if and only if s_o is valid for all M.

3 Shallow Semantical Embedding of FFOL in HOL

We now present a shallow embedding of FFOL in HOL by identifying the language constructs of FFOL with corresponding terms of HOL. In this embedding the raw domain D of a FFOL model is identified with the domain of individuals D_i in a corresponding HOL model. The subdomain E of existing objects is characterized in the embedding by a HOL predicate E of type $i \to o$. Hence, we assume in the remainder that a respective uninterpreted constant symbol $E_{i\to o}$ is given in the signature of HOL. Moreover, we assume that an uninterpreted constant symbol $\star$ of type i is in the signature of HOL. Finally, we assume that $\|E\star_i\|^{M,g} = F$ for all M, g, i.e. that the element denoted by $\star_i$ is not an element of the domain of existing objects denoted by $E_{i\to o}$ (technically this can be achieved by postulating a respective axiom).

Definition 12 (*Embedding of FFOL in HOL*) Given a formula $s \in$ FFOL. We map s to a corresponding term $\widehat{s}$ of HOL. This mapping is defined as follows:

$\widehat{x} := X_i$ for all $x \in V$

$\widehat{f(t^1, \ldots, t^n)} := (\widehat{f}\,\widehat{t^1} \ldots \widehat{t^n})$ for all n-ary $f \in F$ $(n \geq 0)$ where $\widehat{f} = f$ is an uninterpreted constant symbol of type $\underbrace{i \to \ldots \to i}_{n\geq o} \to i$

Revised Proof

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

$\widehat{s = t}$ $:= \widehat{s} = \widehat{t}$

$\widehat{p(t^1, \ldots, t^n)}$ $:= (\widehat{p}\,\widehat{t^1} \ldots \widehat{t^n})$ for all n-ary $p \in P$ $(n \geq 0)$ where $\widehat{p} = \boldsymbol{p}$ is an uninterpreted constant symbol of type $\underbrace{\iota \to \ldots \to \iota}_{n \geq o} \to o$

$\widehat{\neg s}$ $:= \neg\widehat{s}$

$\widehat{s \to r}$ $:= \widehat{s} \to \widehat{r}$

$\widehat{\forall x.r}$ $:= \forall \boldsymbol{X_i}.\boldsymbol{EX_i} \to \widehat{r}$

$\widehat{\imath x.r}$ $:= \boldsymbol{IfThenElse}$

$(\exists \boldsymbol{X_i}.\boldsymbol{EX} \wedge \widehat{r} \wedge (\forall \boldsymbol{Y_i}.(\boldsymbol{EY} \wedge ((\lambda \boldsymbol{X_i}.\widehat{r})\boldsymbol{Y})) \to \boldsymbol{Y} = \boldsymbol{X}))$

$(\imath \boldsymbol{X_i}.\widehat{r})$

$\star$

where $\boldsymbol{IfThenElse}$ is an abbreviation for the term

$$\lambda \boldsymbol{S_o}.\lambda \boldsymbol{X_i}.\lambda \boldsymbol{Y_i}.\imath \boldsymbol{Z_i}.(\boldsymbol{S_o} \wedge \boldsymbol{Z} = \boldsymbol{X}) \vee (\neg \boldsymbol{S_o} \wedge \boldsymbol{Z} = \boldsymbol{Y})$$

The above mapping induces mappings from the sets $\widehat{V}$, $\widehat{F}$ and $\widehat{P}$ of FFOL to corresponding variables (of type i), uninterpreted function symbols and uninterpreted predicate symbols in HOL, respectively.

To prove soundness and completeness[6] for the embedding, a mapping from FFOL models into Henkin models is employed. This mapping utilizes a mapping of FFOL variable assignments g into corresponding HOL variable assignments $\boldsymbol{g}$ (remember that FFOL domains D are identified with HOL domains $\boldsymbol{D_i}$, i.e. $\widehat{D} = \boldsymbol{D_i}$): let g be a variable assignment for FFOL. Then $\widehat{g} : \widehat{V} \mapsto \widehat{D}$ for HOL is defined such that $\widehat{g}(\boldsymbol{X_i}) = \widehat{g}(\widehat{x}) = g(x)$ for all $\boldsymbol{X_i} \in \widehat{V}$. Finally, $\widehat{g}$ is extended to an assignment for arbitrary variables by choosing $\widehat{g}(\boldsymbol{Y_\alpha}) = \boldsymbol{d} \in \boldsymbol{D_\alpha}$ arbitrary whenever $\boldsymbol{Y_\alpha} \notin \widehat{V}$.

Definition 13 (*Henkin model $\widehat{M}$ for FFOL model M*) Given a FFOL model $M = \langle D, E, I, \star \rangle$. The Henkin model $\widehat{M} = \langle \{\boldsymbol{D_\alpha}\}_{\alpha \in \boldsymbol{T}}, \boldsymbol{I} \rangle$ for M is defined as follows:

- $\boldsymbol{D_i} = D$
- $\boldsymbol{D_o} = \{True, False\}$
- $\boldsymbol{D_{\alpha \to \beta}}$ are chosen as (not necessarily full) collections of functions from $\boldsymbol{D_\alpha}$ to $\boldsymbol{D_\beta}$. Remember, however, that the choice of $\boldsymbol{D_{\alpha \to \beta}}$ must always ensure that the evaluation function $\| \cdot \|^{\widehat{M}, \widehat{g}}$ below remains total, i.e. that all terms denote. In particular, it is required that $\boldsymbol{D_{i \to o}}$ contains the element $\boldsymbol{IE_{i \to o}}$ as characterized below.
- The interpretation function $\boldsymbol{I}$ of $\widehat{M}$ is chosen as follows:
- $\boldsymbol{I}\star_{\boldsymbol{i}} = \star \in \boldsymbol{D_i}$
- For all $\boldsymbol{d} \in \boldsymbol{D_i}$ we have: $(\boldsymbol{IE_{i \to o}})(\boldsymbol{d}) = \boldsymbol{T}$ if and only if $\boldsymbol{d} \in E$. Note that this implies $(\boldsymbol{IE_{i \to o}})(\star_{\boldsymbol{i}}) = \boldsymbol{F}$.
- For all $\boldsymbol{f} = \widehat{f} \in \widehat{F}$ we have: $(\boldsymbol{I}f)(\boldsymbol{d^1}, \ldots, \boldsymbol{d^n}) = (If)(\boldsymbol{d^1}, \ldots, \boldsymbol{d^n})$ for all $\boldsymbol{d^i} \in \boldsymbol{D_i}$ $(i = 1, \ldots, n$ and $n \geq 0)$.
- For all $\boldsymbol{p} = \widehat{p} \in \widehat{P}$ we have: $(\boldsymbol{I}p)(\boldsymbol{d^1}, \ldots, \boldsymbol{d^n}) = \boldsymbol{T}$ if and only if $(\boldsymbol{d^1}, \ldots, \boldsymbol{d^n}) \in (Ip)$ for all $\boldsymbol{d^i} \in \boldsymbol{D_i}$ $(i = 1, \ldots, n$ and $n \geq 0)$.
- For all other constants $\boldsymbol{c_\alpha}$, choose $\boldsymbol{Ic_\alpha} \in \boldsymbol{D_\alpha}$ arbitrary.[7]

[6] Similar soundness and completeness proofs for shallow semantical embeddings have been presented in [4] and [5].

[7] In fact, it may be safely assumed that there are no other constant symbols given in a HOL signature, except for the symbols in $\widehat{F}$ and $\widehat{P}$, the symbols $\boldsymbol{E_{i \to o}}$ and $\star_{\boldsymbol{i}}$ and the logical connectives.

Revised Proof

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

It is not hard to verify that $\widehat{M}$ is a Henkin model.[8]

Lemma 14 *Let $\widehat{M}$ be a Henkin model for FFOL model M. For all terms and formulas $s \in FFOL$ and variable assignments g we have $\|s\|^{M,g} = \|\widehat{s}\|^{\widehat{M},\widehat{g}}$.*

Proof The proof is by induction on the structure of s.

For $s = x \in V$ the claim follows from the definition of $\widehat{g}$. For $s = c \in F$, where c is 0-ary, we get the claim by the choice of $\boldsymbol{I}$ in $\widehat{M}$, and for $s = f(t^1, \ldots, t^n)$, where $f \in F$ is n-ary ($n \geq 1$), we additionally need to apply the induction hypothesis. The arguments for $s = q \in P$, where q is 0-ary, and for $s = p(t^1, \ldots, t^n)$, where $p \in P$ is n-ary ($n \geq 1$), are similar. The most complicated case is when $s = \imath x_{\bullet} r$. We here consider two cases. We either have $d \in E$ with $\|r\|^{M,g[d/x]} = \mathit{True}$ and $\|r\|^{M,g[d'/x]} = \mathit{False}$ for all $d' \neq d \in E$ (i.e. d is the unique existing object for which r holds) or there is no such $d \in E$. In the former case we have $\|\imath x_{\bullet} r\|^{M,g} = d \in E$. By the definition of the embedding $\widehat{\cdot}$, definition of $\widehat{M}$ and $\widehat{g}$, λ-conversion, induction hypothesis and a series of evaluation steps in HOL we get $\|\widehat{\imath x_{\bullet} r}\|^{\widehat{M},\widehat{g}} = \|\boldsymbol{\imath} x_{\bullet} \widehat{r}\|^{\widehat{M},\widehat{g}} = d \in E$ (for the very same d as above). In the second case we have $\|\imath x_{\bullet} r\|^{M,g} = \star$. Again we apply the definition of the embedding $\widehat{\cdot}$, the definition of $\widehat{M}$ and $\widehat{g}$, λ-conversion, induction hypothesis and a series of tedious evaluation steps in HOL to verify that $\|\widehat{\imath x_{\bullet} r}\|^{\widehat{M},\widehat{g}} = \star = \star$. The remaining cases are similar (actually simpler) and left to the reader. □

Theorem 15 *(*Soundness and completeness of the embedding*)*
For all formulas $s \in FFOL$ we have $\models^{FFOL} s$ if and only if $\models^{HOL} \widehat{s}$.

Proof (Soundness, $\leftarrow$) The proof is by contraposition. Assume $\not\models^{\text{FFOL}} s$, i.e, there is a FFOL model M and an assignment g such that $\|s\|^{M,g} = \mathit{False}$. By Lemma 14 we have $\|\widehat{s}\|^{\widehat{M},\widehat{g}} = \mathit{False}$. Hence, we get $\not\models^{\text{HOL}} \widehat{s}$.

(Completeness, $\rightarrow$) Analogous to above by contraposition and Lemma 14. □

The above results enable the employment of any theorem prover that supports HOL with definite description to reason with FFOL, including TPTP THF [34] compliant systems such as Satallax, Nitpick, LEO-II and Leo-III. Alternatively, this theory can be encoded in interactive proof assistants such as Isabelle/HOL, which is the option we have chosen here. We thereby significantly benefit from the powerful proof automation means as provided in Isabelle/HOL, in particular, from the integrated model finder Nitpick, the SMT solvers CVC4 [20] and Z3 [21], and the first-order theorem provers E [31] and Spass [17].

The different properties of FFOL could now be experimentally explored with automated reasoning tools for HOL. We have conducted such experiments in an earlier paper [9]. These experiments confirm the illustrative examples discussed in Scott's paper [32].

4 Exploring Axioms Systems for Category Theory

In an experimental theory-exploration study, utilizing the free logic reasoning framework from above, we have shown how Scott's [33] axiom system for category theory can be derived from a notion of partial monoids. These axioms systems are presented in Table 1.

[8] The fixings introduced in $\widehat{M}$ are not in conflict with any of the requirements regarding frames and interpretations. The existence of a valuation function V for an HOL interpretation crucially depends on how sparse the function spaces have been chosen in frame $\{D_\alpha\}_{\alpha \in T}$. Andrews [1] discusses criteria that are sufficient to ensure the existence of a valuation function; in $\widehat{M}$ these requirements are met.

Retracted Proof

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

The stepwise evolution has been described in detail in [10]. Below we summarize these experiments. However, first we describe some basic modeling decisions for the technical encoding in Isabelle/HOL.

The sources of our experiments as conducted here are available at http://christoph-benzmueller.de/papers/2018-JAR-sources.zip. These sources contain an embedding of full free logic in Isabelle/HOL, that is, with $\star$ and definite description (cf. Fig. 2). In [10], $\star$ and definite description were still avoided; they are in fact not really relevant for the conducted experiments.

Figure 2 displays (parts of) the embedding of FFOL in HOL, encoded in Isabelle/HOL, that we have employed in our experiments.[9] An excerpt of these experiments is shown in Fig. 3.

4.1 Modeling of Basic Concepts

Morphisms in the category are modeled as objects in D (respectively, $\boldsymbol{D_i}$). 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 Sledgehammer in Isabelle.

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

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

This definition was suggested by an exercise in the textbook by Freyd and Scedrov [23] 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.

[9] In the remainder of this article, and inline with our text so far, we present the formulas of FFOL in non-boldface font. These formulas have been encoded in Isabelle/HOL using the abbreviations as introduced in Fig. 2. In the actual source encoding, however, the usage of boldface and non-boldface is (for technical reasons) reversed.

Revised Proof

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

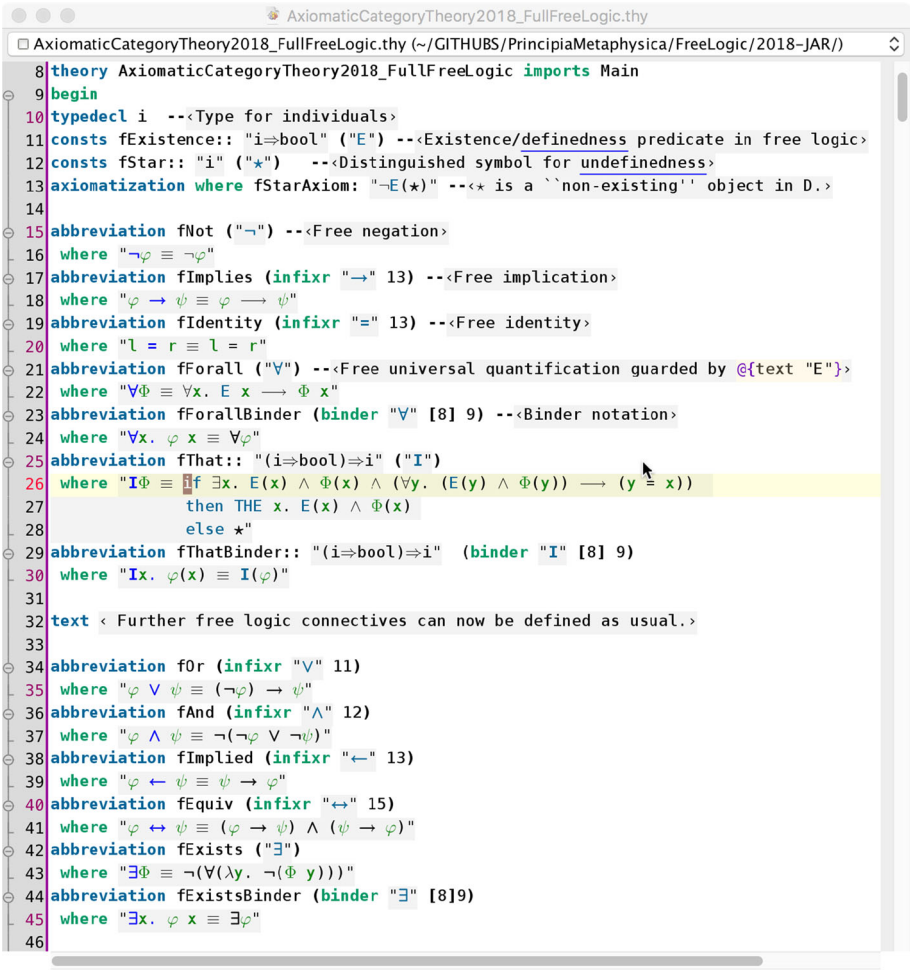

Fig. 2 Isabelle/HOL encoding of FFOL (with ⋆ and definite description)

4.2 Consistency

The model finder Nitpick confirms consistency for all of the axioms 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 axioms sets from Table 1.

Revised

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

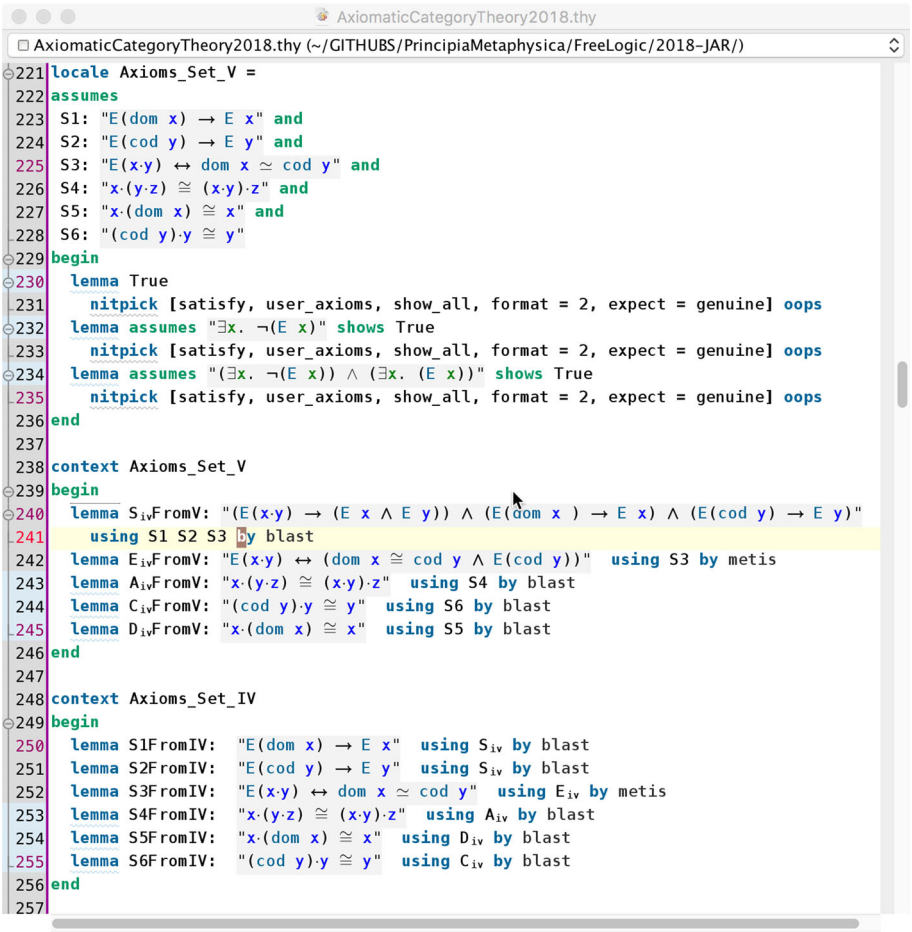

Fig. 3 Encoding of Axioms Set V in Isabelle/HOL utilizing the embedded logic FFOL; Axioms Set V is proven equivalent to Axioms Set IV

4.3 Axioms Sets I and II

Axioms Set I is our most basic set of axioms 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 generalizes 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

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

Table 1 Stepwise evolution of Scott's [33] axiom system for category theory from partial monoids

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 [33]	
$S1$	$E(dom\ x) \longrightarrow Ex$
$S2$	$E(cod\ y) \longrightarrow Ey$
$S3$	$E(x \cdot y) \longleftrightarrow dom\ x \simeq cod\ y$
$S4$	$x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
$S5$	$(cod\ y) \cdot y \cong y$
$S6$	$x \cdot (dom\ x) \cong x$

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 Axioms Sets I and II are free logic quantifiers, that is, they range over the domain E of existing objects

objects (the type $\boldsymbol{i}$ in our setting) 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.

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

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

Revised Proof

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

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

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

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.

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.

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

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

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

The proofs are found by the Sledgehammer tool and automatically reconstructed in Isabelle/HOL. Further information on these experiments are provided in Sect. 4.4 below. 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.

4.4 Remark on the Experiments

All proofs above and all proofs in the rest of this paper (unless stated otherwise) have been obtained fully automatically in very reasonable time (typically just a few seconds) with the Sledgehammer tool in Isabelle/HOL (version Isabelle2017). This tool interfaces to prominent first-order automated theorem provers such as CVC4 [20], Z3 [21], E [31] and Spass [17]. Remotely, also provers such as Vampire [24], or the higher-order provers Satallax [18] and LEO-II [13] 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, in this case C_{ii}, D_{ii}, E_{ii} and S_{ii}. 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. For further information on the use and functioning of Sledgehammer we refer to the literature [14,15].

In our experiments we have also made use of the Isabelle/HOL's SMT method, which "translates the conjecture and any user-supplied facts to the SMT solvers many-sorted first-order logic, invokes a solver, and (depending on the solver) either trusts the result or attempts

Revised Proofs

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

to reconstruct the proof in Isabelle." [15, p. 5].[10] For quite some time the use of the SMT method has been controversially discussed in the Isabelle/HOL community, and there is in fact a significant difference between using the SMT method in combination with Z3 or with CVC4, as we prefer. When setting the solver to CVC4, the contributed proofs are accepted and being trusted without replaying them in the Isabelle kernel. Proofs contributed by Z3, in contrast, are never trusted and always replayed in Isabelle's kernel. For the work presented here this community internal discussion is of minor relevance, so that we decided to continue working with CVC4 in order to keep our formalisation concise and also because CVC4 performed surprisingly well in our experiments.[11]

4.5 Axioms Sets III, IV and V

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.

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

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 V has been proposed by Scott [33] in the 1970s. This set of axioms is equivalent to the axioms set presented by Freyd and Scedrov in their textbook "Categories, Allegories" [23], when encoded in free logic, corrected/adapted and further simplified. Their axioms 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 Sect. 5 for more details.

Axioms Sets II, III, IV and V are equivalent; this has been automatically 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" [23]. In Sect. 5.1 we show that their axioms set, replicated in Table 2 as Axioms Set FS-I, becomes inconsistent in our free logic setting if we assume non-existing objects in D, 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 D. One may argue, that this is not the intention of Freyd and Scedrov. Therefore, we add a second study in Sect. 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 $(f \circ g)x \cong$

[10] Technical remark: We have selected CVC4 in our experiments as the default SMT solver, since we did run into errors when working with Z3. These errors can easily be reconstructed in the provided source files when switching back to Z3 as default.

[11] An expert reviewer of this article, to whom we are very grateful, provided alternative proofs which can be fully replayed in the kernel of Isabelle.

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

Revised Proof

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

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

$A1\ \ E(x \circ y) \longleftrightarrow (x\Box \cong \Box y)$
$A2a((\Box x)\Box) \cong \Box x$
$A2b\Box(x\Box) \cong \Box x$
$A3a(\Box x) \circ x \cong x$
$A3bx \circ (x\Box) \cong x$
$A4a\Box(x \circ y) \cong \Box(x \circ (\Box y))$
$A4b(x \circ y)\Box \cong ((x\Box) \circ y)\Box$
$A5\ \ x \circ (y \circ z) \cong (x \circ y) \circ z$

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

$A1\ \ E(x \cdot y) \longleftrightarrow dom\ x \cong cod\ y$
$A2acod\ (dom\ x) \cong dom\ x$
$A2bdom\ (cod\ y) \cong cod\ y$
$A3ax \cdot (dom\ x) \cong x$
$A3b(cod\ y) \cdot y \cong y$
$A4adom\ (x \cdot y) \cong dom\ ((dom\ x) \cdot y)$
$A4bcod\ (x \cdot y) \cong cod\ (x \cdot (cod\ y))$
$A5\ \ x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$

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

$A1'\ E(x \cdot y) \longleftrightarrow dom\ x \simeq cod\ y$
$A2acod\ (dom\ x) \cong dom\ x$
$A2bdom\ (cod\ y) \cong cod\ y$
$A3ax \cdot (dom\ x) \cong x$
$A3b(cod\ y) \cdot y \cong y$
$A4adom\ (x \cdot y) \cong dom\ ((dom\ x) \cdot y)$
$A4bcod\ (x \cdot y) \cong cod\ (x \cdot (cod\ y))$
$A5\ \ x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$

$g(fx)$ (functional composition from left to right) instead 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 FS-I corresponds to Axioms Set FS-II modulo an appropriate conversion of notation.[12]

5.1 Constricted Inconsistency in Free Logic Setting

A main difference in the system by Freyd and Scedrov to our Axioms 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 their textbook [23].[13] The

[12] 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. $dom\ x \simeq cod\ y$).

[13] 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. …"

Revised Proof

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

difference seems minor, but in our free logic setting it has the effect to cause the mentioned constricted inconsistency issue.[14]

The (constricted) inconsistency of Axioms Set FS-I, respectively Axioms 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.[15] 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 [23], 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 Sets FS-II and FS-I when assuming a non-existing object of type $\boldsymbol{i}$:

$$(\exists x_{\boldsymbol{\cdot}} \neg Ex) \longrightarrow False$$

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

Let $a \in D$ be an undefined object, that is, assume $\neg Ea$. 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 $cod\ (dom\ a) \cong dom\ a$, 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 $\neg E(a \cdot (dom\ a))$. □

As a corollary from the above constricted inconsistency result we get that all morphisms (objects in D) must be defined: $\forall x_{\boldsymbol{\cdot}} Ex$.

Obviously Axioms Sets FS-I and FS-II are also redundant, and we have previously reported on respective redundancies [9].[16] For the corrected Axioms Set VI we still get redundancies. The different options to reduce this system are reported in Table 3.

Attempts to remove axioms A1′, A3a, A3b, and A5 from Axiom Set VI failed. Nitpick shows that they are independent.

However, when assuming strictness of dom and cod , the axioms A2a, A2b, A4a and A4b are all implied. Hence, under this assumptions, the reasoning tools quickly identify (A1′ A3a

[14] This could perhaps be an oversight, or it could indicate that Freyd and Scedrov actually mean the Axioms Set discussed in Sect. 5.2 below.

[15] For this we have to inactivate the axiom that postulates that $\star$ is an undefined/non-existing object.

[16] The discussion in our releated conference paper [9] was before the discovery of the above constricted inconsistency issue, which tells us that the system (in our setting) can even be reduced to axioms A1, A2a, and A3a (when we assume undefined objects).

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

Revised Proof

Table 3 Reduced variants of Axioms Set VI

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

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

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

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

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

$S_v^1\ \ E(dom\ x) \longrightarrow Ex$
$S_v^2\ \ E(cod\ y) \longrightarrow Ey$
$A1'\ E(x \cdot y) \longleftrightarrow dom\ x \simeq cod\ y$
$A3a\ x \cdot (dom\ x) \cong x$
$A3b\ (cod\ y) \cdot y \cong y$
$A5\ \ x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$

A3b A5) as a minimal axiom set, which then exactly matches the Axioms Set V of Scott from Table 1.[17]

5.2 Missing Strictness Axioms in Alternative Setting

We study the axiom system by Freyd and Scedrov once again. However, this time we restrict the free variables in their system to range over existing objects only. In the context of algebraic theories, it could be argued that this is the preferred reading of free variables. By employing the free logic universal quantifier $\forall$, which realizes such a restriction, we thus modify Axioms Set FS-II into Axioms-Set FS-III as displayed in Table 4.

For Axioms Set FS-III the consistency checks with Nitpick succeed, even if we assume undefined objects. However, this axioms set is obviously weaker than Axioms Set V from Table 1. In fact, as has been shown by Nitpick, none of the axioms of this set are implied. The situation changes when we explicitly postulate strictness of dom, cod and $\cdot$. Doing so we obtain Axioms Set FS-IV from Table 4, which, as Nitpick confirms, is consistent even if we assume undefined objects. And the automated theorem provers via Sledgehammer confirm that Axioms Set FS-IV is equivalent to Axioms Set V, as intended. Unfortunately, however, respective strictness conditions are not mentioned in the textbook by Freyd and Scedrov.

[17] This minimal set of axioms has also been mentioned by Freyd in a note [22] and attributed to Martin Knopman. However, the proof sketch presented there seems to fail when the adapted version of A1 (with $\simeq$) is employed.

Revised Proof

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

Table 4 The axioms set of Freyd and Scedrov in our notation and with variable restriction to existing objects only

Axioms Set FS-III: Freyd and Scedrov in our notation (with issues)	
$B1$	$\forall x.\forall y.E(x \cdot y) \longleftrightarrow dom\ x \cong cod\ y$
$B2a$	$\forall x.cod\ (dom\ x) \cong dom\ x$
$B2b$	$\forall y.dom\ (cod\ y) \cong cod\ y$
$B3a$	$\forall x.x \cdot (dom\ x) \cong x$
$B3b$	$\forall y.(cod\ y) \cdot y \cong y$
$B4a$	$\forall x.\forall y.dom\ (x \cdot y) \cong dom\ ((dom\ x) \cdot y)$
$B4b$	$\forall x.\forall y.cod\ (x \cdot y) \cong cod\ (x \cdot (cod\ y))$
$B5$	$\forall x.\forall y.\forall z.x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
Axioms Set FS-IV: Freyd and Scedrov in our notation (without issues)	
$B0a$	$E(x \cdot y) \longrightarrow (Ex \wedge Ey)$
$B0b$	$E(dom\ x) \longrightarrow Ex$
$B0b$	$E(cod\ x) \longrightarrow Ex$
$B1$	$\forall x.\forall y.E(x \cdot y) \longleftrightarrow dom\ x \cong cod\ y$
$B2a$	$\forall x.cod\ (dom\ x) \cong dom\ x$
$B2b$	$\forall y.dom\ (cod\ y) \cong cod\ y$
$B3a$	$\forall x.x \cdot (dom\ x) \cong x$
$B3b$	$\forall y.(cod\ y) \cdot y \cong y$
$B4a$	$\forall x.\forall y.dom\ (x \cdot y) \cong dom\ ((dom\ x) \cdot y)$
$B4b$	$\forall x.\forall y.cod\ (x \cdot y) \cong cod\ (x \cdot (cod\ y))$
$B5$	$\forall x.\forall y.\forall z.x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$

6 Summary and Further Work

We have developed a new reasoning framework for free logic, and we have experimentally applied it for some first experiments in category theory. We have demonstrated how modern proof assistants and theorem provers for classical higher-order logic may well support the reasoning in free logic. More concretely, we have applied our new free logic reasoning framework for the systematic exploration of axiom systems for category theory. Without tools, support of such experiments would be extremely tedious and error prone. In the course of our experiments, automated theorem provers have revealed some (minor) issue in the textbook of Freyd and Scedrov [23], which we were able to correct. The correction essentially corresponds to the axiom system by Scott proposed earlier [32]. All our findings were achieved directly by or in close interaction with automated reasoning tools. Perhaps the lesson to be learned here is that, when working with partial functions, it is natural—out of caution—to assume too much, and the automated reasoning tools, as we have shown here, can help find in what ways the axioms might be reduced or simplified.

Comparisons with other theorem provers for free logic are not possible at this stage, since we are not aware of any other existing systems.

Further work includes the extension of our work towards an embedding of free higher-order logic, the continuation of our formalization studies in category theory (especially extensions of the theory involving functors) and the application of free logic to various other mathematical domains, including, for example, projective geometry. Regarding extensions towards free higher-order logic some first steps have already been taken [28,35], and a recent continuation

Revised Proof

Journal: 10817 Article No.: 9507 TYPESET DISK LE CP Disp.:2018/12/17 Pages: 20 Layout: Small

of our formalisation studies [11] now also includes an early axiom system for category theory by Saunders MacLane [27].

Moreover, as an alternative to always unfolding the mapping from FFOL to HOL, abstract level proof tactics could be provided e.g. in Isabelle/HOL to support intuitive interaction (and even automation) in FFOL on top the semantical embedding.

Acknowledgements We thank Günter Rote, Lutz Schröder and and Emil Weydert for their comments to [10], which together with [9] forms the basis for this article. We also want to express our gratitude to the reviewers of this article. Their fruitful feedback definitely helped to improve the final version.

References

1. Andrews, P.: General models and extensionality. J. Symb. Log. **37**(2), 395–397 (1972)
2. Andrews, P.: Church's type theory. In: Zalta, E. N (ed.) The Stanford Encyclopedia of Philosophy, Summer 2018 edn. Metaphysics Research Lab, Stanford University (2018). https://plato.stanford.edu/archives/sum2018/entries/type-theory-church/
3. Barendregt, H., Dekkers, W., Statman, R.: Lambda Calculus with Types. Perspectives in Logic. Cambridge University Press, Cambridge (2013)
4. Benzmüller, C.: Automating quantified conditional logics in HOL. In: Rossi, F. (ed.) Proceedings of IJCAI-23. Beijing, China (2013)
5. Benzmüller, C.: Cut-elimination for quantified conditional logic. J. Philos. Log. **46**, 333–353 (2016)
6. Benzmüller, C., Brown, C., Kohlhase, M.: Higher-order semantics and extensionality. J. Symb. Log. **69**(4), 1027–1088 (2004)
7. Benzmüller, C., Brown, C., Kohlhase, M.: Cut-simulation and impredicativity. Log. Methods Comput. Sci. **5**(1:6), 1–21 (2009)
8. Benzmüller, C., Miller, D.: Automation of higher-order logic. In: Siekmann, J., Gabbay, D., Woods, J. (eds.) Handbook of the History of Logic, Volume 9—Logic and Computation. Elsevier, Amsterdam (2014)
9. Benzmüller, C., Scott, D.: Automating free logic in Isabelle/HOL. In: Greuel, G.M., Koch, T., Paule, P., Sommese, A. (eds.) Mathematical Software—ICMS 2016, 5th International Congress, Proceedings, LNCS, vol. 9725, pp. 43–50. Springer, Berlin, Germany (2016)
10. Benzmüller, C., Scott, D.S.: Axiomatizing category theory in free logic. CoRR (2016). arXiv:1609.01493
11. Benzmüller, C., Scott, D.S.: Axiom systems for category theory in free logic. Archive of Formal Proofs (2018). https://www.isa-afp.org/entries/AxiomaticCategoryTheory.html
12. Benzmüller, C., Steen, A., Wisniewski, M.: Leo-III version 1.1 (system description). In: Eiter, T., Sands, D. (eds.) Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)—Short Papers, Kalpa Puplications. EasyChair, Maun, Botswana (2017) **(to appear)**
13. Benzmüller, C., Sultana, N., Paulson, L.C., Theiss, F.: The higher-order prover Leo-II. J. Autom. Reason. **55**(4), 389–404 (2015)
14. Blanchette, J.C.: Hammering Away—A Users Guide to Sledgehammer for Isabelle/HOL. Institut für Informatik, Technische Universität München (2018). https://isabelle.in.tum.de/doc/sledgehammer.pdf. With contributions from Lawrence C. Paulson
15. Blanchette, J.C., Böhme, S., Paulson, L.C.: Extending Sledgehammer with SMT solvers. J. Autom. Reason. **51**(1), 109–128 (2013)
16. Blanchette, J.C., Nipkow, T.: Nitpick: a counterexample generator for higher-order logic based on a relational model finder. In: Kaufmann, M., Paulson, L.C. (eds.) Interactive Theorem Proving, First International Conference, ITP 2010, Edinburgh, UK, July 11–14, 2010. Proceedings, Lecture Notes in Computer Science, vol. 6172, pp. 131–146. Springer (2010)
17. Blanchette, J.C., Popescu, A., Wand, D., Weidenbach, C.: More SPASS with Isabelle—superposition with hard sorts and configurable simplification. In: Beringer, L., Felty, A.P. (eds.) 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)
18. Brown, C.E.: Satallax: an automatic higher-order prover. In: Gramlich, B., Miller, D., Sattler, U. (eds.) 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)
19. Church, A.: A formulation of the simple theory of types. J. Symb. Log. **5**, 56–68 (1940)

Not Final Proof

Journal: **10817** Article No.: **9507** TYPESET DISK LE CP Disp.:**2018/12/17** Pages: **20** Layout: **Small**

20. Deters, M., Reynolds, A., King, T., Barrett, C.W., Tinelli, C.: A tour of CVC4: How it works, and how to use it. In: Claessen, K., Kuncak, V. (eds.) Formal Methods in Computer-Aided Design, FMCAD 2014, Lausanne, Switzerland, October 21–24, 2014, p. 7. IEEE (2014)
21. de Moura, L.M., Bjørner, N.: Z3: An efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) 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)
22. Freyd, P.: Amplifications, Diminutions, Subscorings for Categories, Allegories (2016). University of Pennsylvania. Unpublished. https://www.math.upenn.edu/~pjf/amplifications.pdf. Accessed Aug 2016
23. Freyd, P., Scedrov, A.: Categories. North Holland, Allegories (1990)
24. Kovács, L., Voronkov, A.: First-order theorem proving and vampire. In: Sharygina, N., Veith, H. (eds.) 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)
25. Lambert, K.: The definition of e(xistence)! in free logic. In: Abstracts: The International Congress for Logic, Methodology and Philosophy of Science. Stanford University Press, Stanford (1960)
26. Lambert, K.: Free Logic: Selected Essays. Cambridge University Press, Cambridge (2002)
27. MacLane, S.: Groups, categories and duality. Proc. Nat. Acad. Sci. **34**(6), 263–267 (1948)
28. Makarenko, I.: Automatisierung von freier logik in logik höherer stufe. Bachelorarbeit. Freie Universität Berlin, Institut für Informatik (2016)
29. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL: A Proof Assistant for Higher-Order Logic. No. 2283 in LNCS. Springer (2002)
30. Nolt, J.: Free logic. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy, Fall 2018 edn. Metaphysics Research Lab, Stanford University (2018). https://plato.stanford.edu/archives/fall2018/entries/logic-free/
31. Schulz, S.: System description: E 1.8. In: McMillan, K.L., Middeldorp, A., Voronkov, A. (eds.) 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). http://dx.doi.org/10.1007/978-3-642-45221-5
32. Scott, D.: Existence and description in formal logic. In: Schoenman, R. (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)
33. Scott, D.: Identity and existence in intuitionistic logic. In: Fourman, M., Mulvey, C., Scott, D. (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)
34. Sutcliffe, G., Benzmüller, C.: Automated reasoning in higher-order logic using the TPTP THF infrastructure. J. Formaliz. Reason. **3**(1), 1–27 (2010)
35. Wisniewski, M., Steen, A., Benzmüller, C.: TPTP and beyond: representation of quantified non-classical logics. In: Benzmüller, C., Otten, J. (eds.) ARQNL 2016. Automated Reasoning in Quantified Non-Classical Logics, vol. 1770, pp. 51–65. CEUR Workshop Proceedings. http://ceur-ws.org (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Revised Proof

Journal: **10817** Article No.: **9507** TYPESET DISK LE CP Disp.:**2018/12/17** Pages: **20** Layout: **Small**