Free Logic and Category Theory in Isabelle/HOL: Experiments

Christoph Benzmüller and Dana Scott

April 18, 2016

Abstract

We present an interactive and automated theorem prover for free higher-order logic. Our implementation on top of the Isabelle/HOL framework utilizes a semantic embedding of free logic in classical higher-order logic. The capabilities of our tool are demonstrated with first experiments in category theory.

1 Introduction

Although undefinedness and partiality are core concepts in many areas of mathematics, modern mathematical proof assistants and theorem proving systems — they are usually based on some classical or intuitionistic logic — offer rather unsatisfactory support for their natural treatment in practical applications. Free logic logic (resp. inclusive logic) [**?**] [5] offers a theoretically and practically interesting solution. Unfortunately, however, no implementation of a theorem proving system for free logic (or inclusive logic) is, to our best knowledge, available yet.

In this extended abstract we show how free logic can be "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 first-order and higher-order automated theorem and model finders are integrated 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 the (presumably) first interactive and automated theorem prover (and model finder) for free logic.

To demonstrate the practical relevance of our new system, we report on first experiments with our new reasoning system in category theory. In these experiments the theorem prover was able to detect a (presumably unknown) redundancy in the foundational axiom system of the category theory textbook by Freyd and Scedrov.

This paper has been written entirely within the Isabelle/HOL framework by utilizing the Isabelle BUILD tool @cite "Isabelle-build". It is thus an example of a formally verified mathematical document. The independently verifiable Isabelle source code is available at www.christoph-benzmueller.de/papers/2016-ICMS.zip. Running this code the prerequires the installation of the Isabelle system available at ???. The following command, to be executed in the downloaded and unzip-ed source directory, first verifies our text sources for formal correctness and then generates the vorliegende pdf document from them.

2 Free Logic

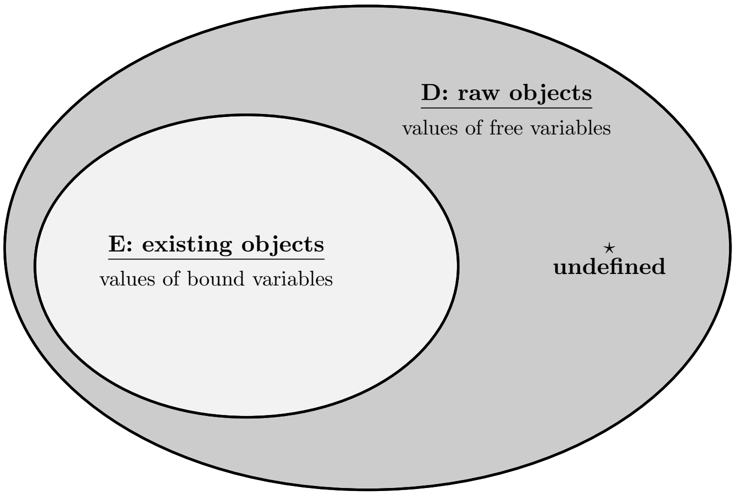
Terms in classical logic denote, without exceptions, entities in a non-empty domain of (existing) objects **D**, and it are these objects of **D** the universal and existential quantifiers do range over. Unfortunately, however, these conditions may render classical logic unsuited for handling mathematically relevant issues such as undefinedness and partiality. For example in category theory composition of maps is not always defined.

Free logic (and inclusive logic) has been proposed as an alternative to remedy these shortcomings. It distinguishes between a raw domain of possibly non-existing objects **D** and a particular subdomain **E** of **D**, containing only the "existing" entities. Free variables range over **D** and quantified variables only over **E**. Each term denotes in **D** but not necessarily in **E**. The particular notion of free logic as exploited below has been introduced by the second author his 1967 article [5]. A graphical illustration of this notion of free logic is presented in Fig. 1.

3 Free Logic in HOL

We start out with introducing a type i of individuals. The domain of objects associated with this this type will serve as the domain of raw objects **D**, cf. Fig 1. Moreover, we introduce an existence predicate **E** on type i. The idea is that **E** is characterising the subset of existing objects in D. Finally, we declare a constant symbol star $\star$. It denotes a distinguished non-existing element of **D**.

typedecl *i* — the type for indiviuals
consts *fExistence* :: *i*⇒*bool* (**E**) — Existence predicate
consts *fStar* :: *i* ($\star$) — Distinguished symbol for undefinedness



1

Figure 1: Free Logic

Next, we postulate that $\star$ denotes a "non-existing" object. Additionally we could require that the domain $\mathbf{E}$ is non-empty: $\exists\, x.\ \mathbf{E}(x)$.

axiomatization where *fStarAxiom*: $\neg\mathbf{E}(\star)$

The two primitive logical connective we introduce for free logic are negation ($\boldsymbol{\neg}$) and implication ($\boldsymbol{\rightarrow}$). They are identified with negation ($\neg$) and implication ($\rightarrow$) in the Isabelle/HOL base logic. The internal names in Isabelle/HOL of the new logical connectives are *fNot* and *fImplies* (the prefix f stands for "free", and $\boldsymbol{\neg}$ the infix operator $\boldsymbol{\rightarrow}$ are introduced as syntactical sugar.

abbreviation *fNot* :: *bool*$\Rightarrow$*bool* ($\boldsymbol{\neg}$)
where $\boldsymbol{\neg}\varphi \equiv \neg\varphi$
abbreviation *fImplies* :: *bool*$\Rightarrow$*bool*$\Rightarrow$*bool* (**infixr** $\boldsymbol{\rightarrow}$ *49*)
where $\varphi\boldsymbol{\rightarrow}\psi \equiv \varphi\longrightarrow\psi$

The main challenge is to appropriately free logic model quantification ($\boldsymbol{\forall}$) and definite description ($\mathbf{I}$). Again, we basically map these operators back to the respective logical connectives $\forall$ and *THE* of the Isabelle/HOL base logic. Different to the identical mappings for $\boldsymbol{\neg}$ and $\boldsymbol{\rightarrow}$ above, however, their mappings are relativized in the sense that the existence predicate $\mathbf{E}$ is utilized as guard in their definitions.

The definition of the free logic universal quantifier $\boldsymbol{\forall}$ thus becomes:

abbreviation *fForall* :: (*i*$\Rightarrow$*bool*)$\Rightarrow$*bool* ($\boldsymbol{\forall}$)
where $\boldsymbol{\forall}\,\Phi \equiv \forall\, x.\ \mathbf{E}(x)\longrightarrow\Phi(x)$

Apparently, this definitions restricts the set of objects $\boldsymbol{\forall}$ is ranging over to the set of existing object $\mathbf{E}$. Note that this set can be empty.

The Isabelle framework supports the introduction of syntactic sugar for binding notations. Here we make use of this option to introduce binding notation for $\forall$. With the definition below we can now use the more familiar notation $\forall x.\ \varphi(x)$ instead of writing $\forall(\lambda x.\ \varphi(x))$ resp. $\forall\varphi$.

abbreviation *fForallBinder* :: $(i \Rightarrow bool) \Rightarrow bool$ (**binder** $\forall$ [*8*] *9*)
where $\forall x.\ \varphi(x) \equiv \forall\varphi$

Definite description **I** in free logic works as follows: Given an unary set $\Phi = \{a\}$, with a being an element of **E**, **I** returns the single element a of Φ. In all other cases, that is, if Φ is not unary or a is not an element of **E**, then $\mathbf{I}\Phi$ returns the distinguished undefined object denoted by $\star$. With the help of Isabelle/HOL's definite description operator *THE*, **I** can thus be defined as follows:

abbreviation *fThat* :: $(i \Rightarrow bool) \Rightarrow i$ (**I**)
where I $\Phi \equiv$ *if* $\exists x.\ \mathbf{E}(x) \wedge \Phi(x) \wedge (\forall y.\ (\mathbf{E}(y) \wedge \Phi(y)) \longrightarrow (y = x))$
then THE $x.\ \mathbf{E}(x) \wedge \Phi(x)$
else $\star$

Analogous to above we introduce binder notation for **I**, so that we can write $\mathbf{I}x.\ \varphi(x)$ instead of $\mathbf{I}(\lambda x.\ \varphi(x))$ resp. $\mathbf{I}\varphi$.

abbreviation *fThatBinder* :: $(i \Rightarrow bool) \Rightarrow i$ (**binder I** [*8*] *9*)
where $\mathbf{I}x.\ \varphi(x) \equiv \mathbf{I}(\varphi)$

Further logical connectives of free can now be defined in the usual way (and for $\exists$ we again introduce binder notation.

abbreviation *fOr* (**infixr** $\vee$ *51*) **where** $\varphi \vee \psi \equiv (\neg\varphi) \rightarrow \psi$
abbreviation *fAnd* (**infixr** $\wedge$ *52*) **where** $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$
abbreviation *fEquiv* (**infixr** $\leftrightarrow$ *50*) **where** $\varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$
abbreviation *fEquals* (**infixr** $=$ *56*) **where** $x = y \equiv x = y$
abbreviation *fExists* ($\exists$) **where** $\exists\Phi \equiv \neg(\forall(\lambda y. \neg(\Phi\ y)))$
abbreviation *fExistsBinder* (**binder** $\exists$ [*8*]*9*) **where** $\exists x.\ \varphi(x) \equiv \exists\varphi$

4 Some Introductory Tests

We exemplarily investigate some example proof problems from [5], pp. 183-184, where a free logic with a single relation symbol **r** is discussed.

consts *f-r* :: $i \Rightarrow i \Rightarrow bool$ (**infixr r** *70*)

The implication $x\ \mathbf{r}\ x \rightarrow x\ \mathbf{r}\ x$, where x is a free variable, is valid independently whether x is defined or not. In Isabelle/HOL this confirmed by the simplification procedure simp.

lemma $x\ \mathbf{r}\ x \rightarrow x\ \mathbf{r}\ x$ **by** *simp*

However, as intended, the formula $\exists y.\ y\ \mathbf{r}\ y \rightarrow y\ \mathbf{r}\ y$ is not valid, since set of existing objects **E** could be empty. Nitpick quickly presents a respective countermodel.

lemma $\exists\, y.\ y\ \mathbf{r}\ y \rightarrow y\ \mathbf{r}\ y$ **nitpick** [*user-axioms*] **oops**

Consequently, the implication $(x\ \mathbf{r}\ x \rightarrow x\ \mathbf{r}\ x) \rightarrow (\exists\, y.\ y\ \mathbf{r}\ y \rightarrow y\ \mathbf{r}\ y)$ has a countermodel, where $\mathbf{E}$ is empty.

lemma $(x\ \mathbf{r}\ x \rightarrow x\ \mathbf{r}\ x) \rightarrow (\exists\, y.\ y\ \mathbf{r}\ y \rightarrow y\ \mathbf{r}\ y)$ **nitpick** [*user-axioms*] **oops**

If we rule out that $\mathbf{E}$ is empty, e.g. with additional condition $(\exists\, y{::}i.\ y = y)$ in the antecedent of the above formula, then we obtain a valid implication. Isabelle trivially proves this with procedure simp.

lemma $((x\ \mathbf{r}\ x \rightarrow x\ \mathbf{r}\ x) \wedge (\exists\, y{::}i.\ y = y)) \rightarrow (\exists\, y.\ y\ \mathbf{r}\ y \rightarrow y\ \mathbf{r}\ y)$ **by** *simp*

We analyse some further statements (respectively statement instances) from the exploration in [5], p. 185. We do not further commend these statements here. They confirm that our implementation of free logic obeys the intended properties.

lemma *S1*: $(\forall\, x.\ \Phi(x) \rightarrow \Psi(x)) \rightarrow ((\forall\, x.\ \Phi(x)) \rightarrow (\forall\, x.\ \Psi(x)))$ **by** *auto*
lemma *S2*: $\forall\, y.\ \exists\, x.\ x = y$ **by** *auto*
lemma *S3*: $\alpha = \alpha$ **by** *auto*
lemma *S4*: $(\Phi(\alpha) \wedge (\alpha = \beta)) \rightarrow \Phi(\beta)$ **by** *auto*
lemma *UI-1*: $((\forall\, x.\ \Phi(x)) \wedge (\exists\, x.\ x = \alpha)) \rightarrow \Phi(\alpha)$ **by** *auto*
lemma *UI-2*: $(\forall\, x.\ \Phi(x)) \rightarrow \Phi(\alpha)$ **nitpick** [*user-axioms*] **oops** — Countermodel by Nitpick
lemma *UI-cor1*: $\forall\, y.((\forall\, x.\ \Phi(x)) \rightarrow \Phi(y))$ **by** *auto*
lemma *UI-cor2*: $\forall\, y.((\forall\, x.\ \neg(x = y)) \rightarrow \neg(y = y))$ **by** *auto*
lemma *UI-cor3*: $\forall\, y.((y = y) \rightarrow (\exists\, x.\ x = y))$ **by** *auto*
lemma *UI-cor4*: $(\forall\, y.\ y = y) \rightarrow (\forall\, y.\exists\, x.\ x = y)$ **by** *simp*
lemma *Existence*: $(\exists\, x.\ x = \alpha) \longrightarrow \mathbf{E}(\alpha)$ **by** *simp*
lemma *I1*: $\forall\, y.\ ((y = (\mathbf{I}x.\ \Phi(x))) \leftrightarrow (\forall\, x.\ ((x = y) \leftrightarrow \Phi(x))))$ **by** (*smt fStarAxiom the-equality*)
abbreviation *Star* ($\bigotimes$) **where** $\bigotimes\ \equiv \mathbf{I}y.\ \neg\ (y = y)$
lemma *StarTest*: $\bigotimes\ = \star$ **by** *simp*
lemma *I2*: $\neg(\exists\, y.\ y = (\mathbf{I}x.\ \Phi(x))) \rightarrow\ (\bigotimes\ = (\mathbf{I}x.\ \Phi(x)))$ **by** (*metis* (*no-types*, *lifting*) *the-equality*)
lemma *ExtI*: $(\forall\, x.\ \Phi(x) \leftrightarrow \Psi(x)) \rightarrow ((\mathbf{I}x.\ \Phi(x)) = (\mathbf{I}x.\ \Psi(x)))$ **by** (*smt the1-equality*)
lemma *I3*: $(\bigotimes\ = \alpha \vee \bigotimes\ = \beta) \rightarrow \neg(\alpha\ \mathbf{r}\ \beta)$ **nitpick** [*user-axioms*] **oops** — Countermodel by Nitpick

5 Application in Category Theory

We exemplary employ the above implementation of free logic for an application in category theory. More precisely, by utilizing our new reasoning framework we study some properties of the category theory axiom system that has been proposed by Freyd and Scedrov, see their textbook "Categories, Allegories" [3], p. 3. As expected, the composition $x{\cdot}y$, for morphisms x and y, is introduced by Freyd and Scedrov as a partial operation; see axiom *A1* below: this composition exists if and only if the target of x coincides with the source of y. This is why free logic, as opposed to e.g. classical logic, is better suited as a starting point for the modeling of their work.[1]

In the remainder we identify the base type i of free logic with the raw type of morphisms. Moreover, we introduce constant symbols for the following operations: source of a morphism x, target of a morphism x and composition of morphisms x and y. These operations are denoted by Freyd and Scedrov as $\Box x$, $x\Box$ and $x{\cdot}y$, respectively. We adopt their notation below, even though we are not particularly fond of the use of $\Box$ in this context.

consts $\mathit{source}{::}i{\Rightarrow}i$ ($\Box$- [*108*] *109*)
  $\mathit{target}{::}i{\Rightarrow}i$ (-$\Box$ [*110*] *111*)
  $\mathit{composition}{::}i{\Rightarrow}i{\Rightarrow}i$ (**infix** $\cdot$ *110*)

Ordinary equality on morphisms is then defined as follows:

abbreviation $\mathit{OrdinaryEquality}{::}i{\Rightarrow}i{\Rightarrow}\mathit{bool}$ (**infix** $\approx$ *60*)
where $x \approx y \equiv ((\mathbf{E}\ x) \leftrightarrow (\mathbf{E}\ y)) \wedge x = y$

We are now in the position to model the axiom system of Freyd and Scedrov.

axiomatization *FreydsAxiomSystem* **where**
A1: $\mathbf{E}(x{\cdot}y) \leftrightarrow ((x\Box) \approx (\Box y))$ **and**
A2a: $((\Box x)\Box) \approx \Box x$ **and**
A2b: $\Box(x\Box) \approx \Box x$ **and**
A3a: $(\Box x){\cdot}x \approx x$ **and**
A3b: $x{\cdot}(x\Box) \approx x$ **and**
A4a: $\Box(x{\cdot}y) \approx \Box(x{\cdot}(\Box y))$ **and**
A4b: $(x{\cdot}y)\Box \approx ((x\Box){\cdot}y)\Box$ **and**
A5: $x{\cdot}(y{\cdot}z) \approx (x{\cdot}y){\cdot}z$

Experiments with our new reasoning framework for free logic and quickly confirm that axiom *A2a* is redundant. For example, as Isabelle's internal prover metis[2] confirms, *A2a* is implied by *A2b*, *A3a*, *A3b* and *A4a*.

lemma *A2aIsRedundant-1*: $(\Box x)\Box \approx \Box x$
by (*metis A2b A3a A3b A4a*)

A human readable and comprehensible reconstruction of this redundancy is presented below. This proof employs axioms *A2b*, *A3a*, *A3b*, *A4a* and *A5*, that is, this proof could be further optimized by eleminating the dependency on *A5*.

lemma *A2aIsRedundant-2*: $(\Box x)\Box \approx \Box x$
proof –
have *L1*: $\forall x.\ (\Box\Box x){\cdot}((\Box x){\cdot}x) \approx ((\Box\Box x){\cdot}(\Box x)){\cdot}x$ **using** *A5* **by** *metis*
hence *L2*: $\forall x.\ (\Box\Box x){\cdot}x \approx ((\Box\Box x){\cdot}(\Box x)){\cdot}x$ **using** *A3a* **by** *metis*

[1] The precise logic setting is unfortunately not discussed in the beginning of Freyd's and Scedrov's textbook. Appendix B, however, contains a concise formal definition of the logic setting they assume.

[2] Metis is trusted prover of the Isabelle system; it returns proofs in Isabelle's trusted kernel.

hence *L3*: $\forall x.\ (\Box\Box x)\cdot x \approx (\Box x)\cdot x$ **using** *A3a* **by** *metis*
hence *L4*: $\forall x.\ (\Box\Box x)\cdot x \approx x$ **using** *A3a* **by** *metis*
have *L5*: $\forall x.\ \Box((\Box\Box x)\cdot x) \approx \Box((\Box\Box x)\cdot(\Box x))$ **using** *A4a* **by** *auto*
hence *L6*: $\forall x\ .\Box((\Box\Box x)\cdot x) \approx \Box\Box x$ **using** *A3a* **by** *metis*
hence *L7*: $\forall x.\ \Box\Box(x\Box) \approx \Box(\Box\Box(x\Box))\cdot(x\Box)$ **by** *auto*
hence *L8*: $\forall x.\ \Box\Box(x\Box) \approx \Box(x\Box)$ **using** *L4* **by** *metis*
hence *L9*: $\forall x.\ \Box\Box(x\Box) \approx \Box x$ **using** *A2b* **by** *metis*
hence *L10*: $\forall x.\ \Box\Box x \approx \Box x$ **using** *A2b* **by** *metis*
hence *L11*: $\forall x.\ \Box\Box((\Box x)\Box) \approx \Box\Box(x\Box)$ **using** *A2b* **by** *metis*
hence *L12*: $\forall x.\ \Box\Box((\Box x)\Box) \approx \Box x$ **using** *L9* **by** *metis*
have *L13*: $\forall x.\ (\Box\Box((\Box x)\Box))\cdot((\Box x)\Box) \approx ((\Box x)\Box)$ **using** *L4* **by** *auto*
hence *L14*: $\forall x.\ (\Box x)\cdot((\Box x)\Box) \approx (\Box x)\Box$ **using** *L12* **by** *metis*
hence *L15*: $\forall x.\ (\Box x)\Box \approx (\Box x)\cdot((\Box x)\Box)$ **using** *L14* **by** *auto*
then show *?thesis* **using** *A3b* **by** *metis*
qed

Thus, axiom *A2a* can be removed from the theory. Alternatively, we could also eliminate *A2b* which is implied by *A1*, *A2a* and *A3a*:

lemma *A2bIsRedundant-2*: $\Box(x\Box) \approx \Box x$ **by** (*metis A1 A2a A3a*)

In fact, by straightforward experimentation with our provers, we can show that Freyd's and Scedroc's axiomatic theory can be reduced to just the following five axioms:

axiomatization *FreydsAxiomSystemReduced* **where**
B1: $\mathbf{E}(x\cdot y) \leftrightarrow ((x\Box) \approx (\Box y))$ **and**
B2a: $((\Box x)\Box) \approx \Box x$ **and**
B3a: $(\Box x)\cdot x \approx x$ **and**
B3b: $x\cdot(x\Box) \approx x$ **and**
B5: $x\cdot(y\cdot z) \approx (x\cdot y)\cdot z$

The dropped axioms can now be introduced as lemmas.

lemma *B2b*: $\Box(x\Box) \approx \Box x$ **by** (*metis B1 B2a B3a*)
lemma *B4a*: $\Box(x\cdot y) \approx \Box(x\cdot(\Box y))$ **by** (*metis B1 B2a B3a*)
lemma *B4b*: $(x\cdot y)\Box \approx ((x\Box)\cdot y)\Box$ **by** (*metis B1 B2a B3a*)

In the remainder of this section we present some further tests wrt Freyd's and Scedrov's theory. We leave these tests uncommented.

abbreviation *DirectedEquality* :: $i \Rightarrow i \Rightarrow bool$ (**infix** $\gtrapprox$ *60*)
where $x \gtrapprox y \equiv ((\mathbf{E}\ x) \rightarrow (\mathbf{E}\ y)) \wedge x = y$

lemma *L1-13*: $((\Box(x\cdot y)) \approx (\Box(x\cdot(\Box y)))) \leftrightarrow ((\Box(x\cdot y)) \gtrapprox \Box x)$ **by** (*metis A1 A2a A3a*)

lemma $(\exists x.\ e \approx (\Box x)) \leftrightarrow (\exists x.\ e \approx (x\Box))$ **by** (*metis A1 A2b A3b*)
lemma $(\exists x.\ e \approx (x\Box)) \leftrightarrow e \approx (\Box e)$ **by** (*metis A1 A2b A3a A3b*)
lemma $e \approx (\Box e) \leftrightarrow e \approx (e\Box)$ **by** (*metis A1 A2b A3a A3b A4a*)
lemma $e \approx (e\Box) \leftrightarrow (\forall x.\ e\cdot x \gtrapprox x)$ **by** (*metis A1 A2b A3a A3b A4a*)
lemma $(\forall x.\ e\cdot x \gtrapprox x) \leftrightarrow (\forall x.\ x\cdot e \gtrapprox x)$ **by** (*metis A1 A2b A3a A3b*)

abbreviation *IdentityMorphism* :: $i \Rightarrow bool$ (*IdM*- [*100*]*60*)
where $IdM\ x \equiv x \approx (\Box x)$

lemma $(IdM\ e \leftrightarrow (\exists\, x.\ e \approx (\Box x))) \wedge$
$(IdM\ e \leftrightarrow (\exists\, x.\ e \approx (x\Box))) \wedge$
$(IdM\ e \leftrightarrow e \approx (\Box e)) \wedge$
$(IdM\ e \leftrightarrow e \approx (e\Box)) \wedge$
$(IdM\ e \leftrightarrow (\forall\, x.\ e{\cdot}x \gtrapprox x)) \wedge$
$(IdM\ e \leftrightarrow (\forall\, x.\ x{\cdot}e \gtrapprox x))$
by (*smt A1 A2a A3a A3b*)
end

References

[1] C. Benzmüller and D. Miller. Automation of higher-order logic. In J. Siekmann, D. Gabbay, and J. Woods, editors, *Handbook of the History of Logic, Volume 9 — Logic and Computation*. Elsevier, 2014. Forthcoming; preliminary version available at http://christoph-benzmueller.de/papers/B5.pdf.

[2] J. Blanchette, S. Böhme, and L. Paulson. Extending Sledgehammer with SMT solvers. *J. of Automated Reasoning*, 51(1):109–128, 2013.

[3] P. J. Freyd and A. Scedrov. *Categories, Allegories*. North Holland, 1990.

[4] T. Nipkow, L. Paulson, and M. Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Number 2283 in LNCS. Springer, 2002.

[5] D. Scott. Existence and description in formal logic. In R. Schoenman, editor, *Bertrand Russell: Philosopher of the Century*, pages 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).