Free Logic and Category Theory in Isabelle/HOL: Experiments

Christoph Benzmüller and Dana Scott

March 20, 2016

Abstract

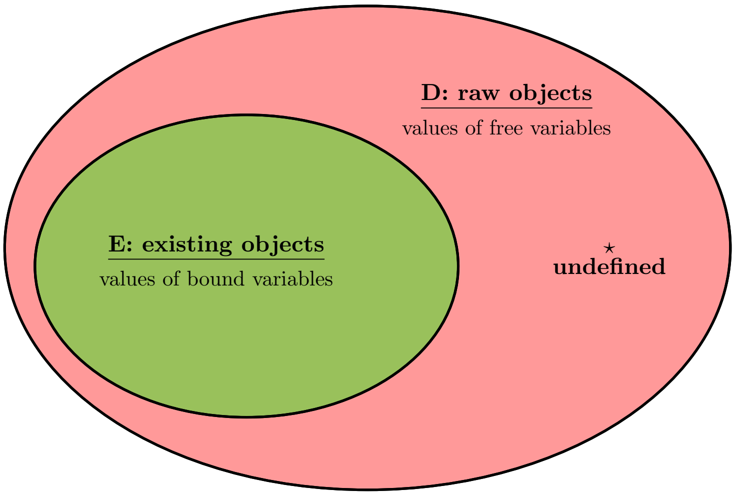
We present a semantic embedding of free logic (and inclusive logic) in classical higher-order logic (HOL). This embedding enables state-of-art theorem provers and model finders for HOL, such as the first author's Leo provers, the proof assistant Isabelle/HOL and the model finder Nitpick, to reason within and about free logic in practical applications.

To illustrate the approach we report on first experiments in which we have analysed axioms systems in category theory. In our experiments theorem provers were able to detect a (presumably unknown) redundancy in the foundational axiom system of the category theory textbook by Freyd and Scedrov.

1 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 [1]. This notion is graphically illustrated in Figure 1.



1

Figure 1: Free Logic

2 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. 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. It denotes a distinguished non-existing element of D.

typedecl *i* — the type for indiviuals
consts *fExistence* :: *i*⇒*bool* (*E*- [*8*] *60*)
consts *fStar* :: *i* (⋆)

axiomatization where *fStarDoesNotExist*: $\neg E(\star)$
axiomatization where *fNonEmptinessOfE*: $\exists\, x.\ E(x)$

Negation and implication in free logic are mapped to negation in HOl.

abbreviation *fNot* :: *bool*⇒*bool* (¬- [*58*] *59*)
where $\neg\varphi \equiv \neg\varphi$
abbreviation *fImplies* :: *bool*⇒*bool*⇒*bool* (**infixr** → *49*)
where $\varphi{\to}\psi \equiv \varphi{\longrightarrow}\psi$

Our embedding of *Free Logic* in HOL exploits and adapts the idea of relativized quantifiers

Universal quantification in free logic is restricted to the domain of existing objects

abbreviation *fForall* :: (*i*⇒*bool*)⇒*bool* (∀)
where $\forall\,\Phi \equiv \forall\, x.\ E(x){\longrightarrow}\Phi(x)$
abbreviation *fForallBinder* :: (*i*⇒*bool*)⇒*bool* (**binder** ∀ [*8*] *9*)
where $\forall\, x.\ \varphi(x) \equiv \forall\,\varphi$

abbreviation *fThat* :: $(i{\Rightarrow}bool){\Rightarrow}i$ (**I**)
where **I** $\Phi \equiv$ *if* $\exists\, x.\ E(x) \wedge \Phi(x) \wedge (\forall\, y.\ (E(y) \wedge \Phi(y)) \longrightarrow (y = x))$
then *THE* $x.\ E(x) \wedge \Phi(x)$
else $\star$
abbreviation *fThatBinder* :: $(i{\Rightarrow}bool){\Rightarrow}i$ (**binder I** [*8*] *9*)
where $\mathbf{I}x.\ \varphi(x) \equiv \mathbf{I}(\varphi)$

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

3 Some Introductory Tests

— See Scott, Existence and Description in Formal Logic, 1967, pages 183-184

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

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

lemma $\exists\, y.\ y\ \mathbf{r}\ y \rightarrow y\ \mathbf{r}\ y$ **nitpick oops**

lemma $(x\ \mathbf{r}\ x \rightarrow x\ \mathbf{r}\ x) \rightarrow (\exists\, y.\ y\ \mathbf{r}\ y \rightarrow y\ \mathbf{r}\ y)$ **nitpick oops**

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*

— See Scott 1967, page 185
lemma *S1-inst* : $(\forall\, x.\ \Phi(x) \rightarrow \Psi(x)) \rightarrow ((\forall\, x.\ \Phi(x)) \rightarrow (\forall\, x.\ \Psi(x)))$ **by** *auto*
lemma *S2* : $\forall\, y.\ \exists\, x.\ x = y$ **by** *auto*
lemma *S3* : $\alpha = \alpha$ **by** *auto*
lemma *S4-inst* : $(\Phi(\alpha) \wedge (\alpha = \beta)) \rightarrow \Phi(\beta)$ **by** *auto*
lemma *UI-inst* : $((\forall\, x.\ \Phi(x)) \wedge (\exists\, x.\ x = \alpha)) \rightarrow \Phi(\alpha)$ **by** *auto*
lemma *UI-test* : $(\forall\, x.\ \Phi(x)) \rightarrow \Phi(\alpha)$ **nitpick** [*user-axioms*] **oops** — Countermodel
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 $(\exists\, x.\ x = \alpha) \longrightarrow E(\alpha)$ **by** *simp*

lemma *I1-inst* : $\forall\, y.\ ((y = (\mathbf{I}x.\ \Phi(x))) \leftrightarrow (\forall\, x.\ ((x = y) \leftrightarrow \Phi(x))))$ **by** (*smt fStarDoesNotExist the-equality*)

abbreviation *star* ($\bigotimes$) **where** $\bigotimes \equiv \mathbf{I}y.\ \neg\ (y = y)$

lemma *test* : $\bigotimes\ = \star$ **by** *simp*

lemma *I2-inst* : $\neg(\exists\, y.\ y = (\mathbf{I}x.\ \Phi(x))) \rightarrow (\bigotimes\ = (\mathbf{I}x.\ \Phi(x)))$ **by** (*metis* (*no-types*, *lifting*) *the-equality*)

lemma *Ext-inst* : $(\forall\, x.\ \Phi(x) \leftrightarrow \Psi(x)) \rightarrow ((\mathbf{I}x.\ \Phi(x)) = (\mathbf{I}x.\ \Psi(x)))$ **by** (*smt the1-equality*)

lemma *I3* : $(\bigotimes\ = \alpha \vee \bigotimes\ = \beta) \rightarrow \neg(\alpha\ \mathbf{r}\ \beta)$ **nitpick** [*user-axioms*] **oops**

lemma *Russel-inst* :
$((\bigotimes\ = \alpha \vee \bigotimes\ = \beta) \rightarrow \neg(\alpha\ \mathbf{r}\ \beta))$
$\rightarrow$
$((\alpha\ \mathbf{r}\ (\mathbf{I}x.\ \Phi(x))) \leftrightarrow (\exists\, y.((\forall\, x.\ ((x = y) \leftrightarrow \Phi(x))) \wedge (\alpha\ \mathbf{r}\ y))))$
nitpick [*user-axioms*] **oops**

lemma $\neg(\exists\, x.\ (x = (\mathbf{I}y.\ \neg\ (y = y))))$ **using** *fStarDoesNotExist* **by** *auto*
lemma $(\exists\, x.\ x = a) \rightarrow E(a)$ **by** *simp*

consts *ca::i cb::i*
axiomatization where *ax1*: $\mathcal{A}(ca) \wedge \mathcal{A}(cb) \wedge \neg\ (ca = cb) \wedge \neg\ (ca = \bigotimes) \wedge \neg\ (cb = \bigotimes)$
lemma *test2*: $\bigotimes\ = (\mathbf{I}\ (\lambda x.\ x\ = ca \vee x = cb))$ **by** (*metis ax1*)

end
theory *Freyd* **imports** *FreeFOL*

begin
type-synonym $e = i$ — raw type of morphisms

abbreviation *OrdinaryEquality* :: $e \Rightarrow e \Rightarrow bool$ (**infix**$\approx$ *60*)
where $x \approx y \equiv ((E\ x) \leftrightarrow (E\ y)) \wedge x = y$

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

axiomatization *FreydsAxioms* **where**
A1: $(E\ x{\cdot}y) \leftrightarrow ((x\Box) \approx (\Box y))$ **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: $x{\cdot}(y{\cdot}z) \approx (x{\cdot}y){\cdot}z$

lemma *A2a*: $(\Box x)\Box \approx \Box x$
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*
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

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

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* :: *e*$\Rightarrow$*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] 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).