Axiomatizing Category Theory in Free Logic

Christoph Benzmüller¹ and Dana S. Scott²

¹ University of Luxemburg, Luxemburg & Freie Universität Berlin, Germany ² Visiting Scholar at University of California, Berkeley, USA

April 27, 2017

Abstract

Starting from a generalization of the standard axioms for a monoid we present a step-wise development of various, mutually equivalent foundational axiom systems for category theory. Our axiom sets have been formalized in the Isabelle/HOL interactive proof assistant, and this formalization utilizes a semantically correct embedding of free logic in classical higher-order logic. The modeling and formal analysis of our axiom sets has been significantly supported by series of experiments with automated reasoning tools integrated with 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 for which we reveal a technical issue (when encoded in free logic): either all operations, e.g. morphism composition, are total or their axiom system is inconsistent. The repair for this problem is quite straightforward, however.

1 Introduction

We present a stepwise development of axiom systems for category theory by generalizing the standard axioms for a monoid to a partial composition operation. Our purpose is not to make or claim any contribution to category theory but rather to show how formalizations involving the kind of logic required (free logic) can be validated within modern proof assistants.

A total of eight different axiom systems is studied. The systems I-VI are shown to be equivalent. The axiom system VII slightly modifies axiom system VI to obtain (modulo notational transformation) the set of axioms as proposed by Freyd and Scedrov in their textbook "Categories, Allegories" [9], published in 1990; see also Subsection 9.2 where we present their original system. While the axiom systems I-VI are shown to be consistent, a constricted inconsistency result is obtained for system VII (when encoded in free logic where free variables range over all objects): We can prove $(\exists x. \neg(E x)) \rightarrow False$, where E is the existence predicate. Read this as: If there are undefined objects, e.g. the value of an undefined composition $x \cdot y$, then we have falsity. By contraposition, all objects (and thus all compositions) must exist. But when we assume the latter, then the axiom system VII essentially reduces categories to monoids. We note that axiom system V, which avoids this problem, corresponds to a set of axioms proposed by Scott [14] in the 1970s. The problem can also be avoided by restricting the variables in axiom system VII to range only over existing objects and by postulating strictness conditions. This gives us axiom system VIII.

Our exploration has been significantly supported by series of experiments in which automated reasoning tools have been called from within the proof assistant Isabelle/HOL [11] via the Sledgehammer tool [4]. Moreover, we have obtained very useful feedback at various stages from the model finder Nitpick [5] saving us from making several mistakes.

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

To enable our experiments we have exploited an embedding of free logic [13] in classical higher-order logic, which we have recently presented in a related paper [1].

We also want to emphasize that this paper has been written entirely within the Isabelle framework by utilizing the Isabelle "build" tool; cf. [15], Section 2. It is thus an example of a formally verified mathematical document, where the PDF document as presented here has been generated directly from the verified source files mentioned above. We also note that once the proofs have been mechanically checked, they are generally easy to find by hand using paper and pencil.

2 Embedding of Free Logic in HOL

Free logic models partial functions as total functions over a "raw domain" D. A subset E of D is used to characterize the subdomain of "existing" objects; cf. [13] for further details.

The experiments presented in the subsequent sections exploit our embedding of free logic in HOL [1]. This embedding is trivial for the standard Boolean connectives. The interesting aspect is that free logic quantifiers are guarded in the embedding by an explicit existence predicate E (associated with the subdomain E of D), so that quantified variables range only over existing objects, while free variables and arbitrary terms may also denote undefined/non-existing objects outside of E. This way we obtain an elegant treatment of partiality resp. undefinednes as required in category theory. In our related paper [1] we also show how definite description can be appropriately modeled in this approach. However, the definite description is not required for purposes of this paper, so we omit it. Note that the connectives and quantifiers of free logic are displayed below in bold-face fonts. Normal, non-bold-face connectives and quantifiers in contrast belong to the meta-logic HOL. The prefix "f", e.g. in fNot, stands for "free".

```
typedecl i — Type for individuals consts fExistence:: i\Rightarrow bool (E) — Existence/definedness predicate in free logic abbreviation fNot (\neg) — Free negation where \neg \varphi \equiv \neg \varphi abbreviation fImplies (infixr \rightarrow 13) — Free implication where \varphi \rightarrow \psi \equiv \varphi \rightarrow \psi abbreviation fIdentity (infixr = 13) — Free identity where l = r \equiv l = r
```

```
abbreviation fForall (\forall) — Free universal quantification guarded by existence predicate E where \forall \Phi \equiv \forall x. \ E \ x \longrightarrow \Phi \ x abbreviation fForallBinder (binder \forall \ [8] \ 9) — Binder notation where \forall x. \ \varphi \ x \equiv \forall \ \varphi
```

Further free logic connectives can now be defined as usual.

```
abbreviation fOr (infixr \vee 11) where \varphi \vee \psi \equiv (\neg \varphi) \rightarrow \psi abbreviation fAnd (infixr \wedge 12) where \varphi \wedge \psi \equiv \neg(\neg \varphi \vee \neg \psi) abbreviation fImplied (infixr \leftarrow 13) where \varphi \leftarrow \psi \equiv \psi \rightarrow \varphi abbreviation fEquiv (infixr \leftrightarrow 15) where \varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi) 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
```

In this framework partial and total functions are modelled as follows: A function f is total if and only if for all x we have $E x \to E(f x)$. For partial functions f we may have some x such that E x but not E(f x). A function f is strict if and only if for all x we have $E(f x) \to E x$.

3 Preliminaries

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

consts

```
domain:: i \Rightarrow i \pmod{-[108] 109}

codomain:: i \Rightarrow i \pmod{-[110] 111}

composition:: i \Rightarrow i \Rightarrow i \pmod{x} \cdot 110
```

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

$$(x \cdot y)a \cong x(y \ a)$$
 when $dom \ x \simeq cod \ y$

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):

```
abbreviation KlEq (infixe \cong 56) — Kleene equality where x \cong y \equiv (E \ x \lor E \ y) \rightarrow x = y
```

Reasoning tools in Isabelle quickly confirm that \cong is an equivalence relation. But existing identity \cong , in contrast, is only symmetric and transitive, and lacks reflexivity. It is defined as:

```
abbreviation ExId (infixr \simeq 56) — Existing identity where x \simeq y \equiv E \ x \land E \ y \land x = y

We have:

lemma x \cong x \land (x \cong y \to y \cong x) \land ((x \cong y \land y \cong z) \to x \cong z)
by blast

lemma x \simeq x — This does not hold; Nitpick finds a countermodel. 
nitpick [user-axioms, show-all, format = 2, expect = genuine] oops

lemma (x \simeq y \to y \simeq x) \land ((x \simeq y \land y \simeq z) \to x \simeq z)
by blast

lemma x \simeq y \to x \cong y
by simp

lemma x \simeq y \to x \cong y
by simp

lemma x \simeq y \leftrightarrow x \cong y— Nitpick finds a countermodel nitpick [user-axioms, show-all, format = 2, expect = genuine] oops

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

abbreviation I where I i \equiv (\forall x. \ E(i \cdot x) \to i \cdot x \cong x) \land (\forall x. \ E(x \cdot i) \to x \cdot i \cong x)
```

This definition was suggested by an exercise in [9] 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.

4 Axiom Set I

Axiom Set I is our most basic axiom set for category theory generalizing the axioms for a monoid to a partial composition operation. Remember that a monoid is an algebraic structure (S, \circ) , where \circ is a binary operator on set S, satisfying the following properties:

```
Closure: \forall a, b \in S. \ a \circ b \in S

Associativity: \forall a, b, c \in S. \ a \circ (b \circ c) = (a \circ b) \circ c

Identity: \exists id_S \in S. \forall a \in S. \ id_S \circ a = a = a \circ id_S
```

That is, a monoid is a semigroup with a two-sided identity element.

Our first axiom set for category theory employs a partial, strict binary composition operation \cdot , and the existence of left and right identity elements is addressed in the last two axioms. The notions of dom (Domain) and cod (Codomain) abstract from their common meaning in the context of sets. In category theory we work with just a single type of objects (the type i of morphisms) and therefore identity morphisms are employed to suitably characterize their meanings.

```
\begin{array}{lll} S_i \colon & - & \text{Strictness:} & E(x \cdot y) \to (E \ x \land E \ y) \ \text{and} \\ E_i \colon & - & \text{Existence:} & E(x \cdot y) \leftarrow (E \ x \land E \ y \land (\exists \ z. \ z \cdot z \cong z \land x \cdot z \cong x \land z \cdot y \cong y)) \ \text{and} \\ A_i \colon & - & \text{Associativity:} & x \cdot (y \cdot z) \cong (x \cdot y) \cdot z \ \text{and} \\ C_i \colon & - & \text{Codomain:} & \forall \ y . \exists \ i. \ I \ i \land i \cdot y \cong y \ \text{and} \\ D_i \colon & - & \text{Domain:} & \forall \ x . \exists \ j. \ I \ j \land x \cdot j \cong x \end{array}
```

Nitpick confirms that this axiom set is consistent.

¹The keyword "oops" in Isabelle/HOL indicates a failed/incomplete proof attempt; the respective (invalid) conjecture is then not made available for further use. The simplest countermodel for the conjecture given here consists of single, non-existing element.

```
lemma True — Nitpick finds a model nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops
```

Even if we assume there are non-existing objects we get consistency (which is e.g. not the case for Axiom Set VII below).

```
lemma assumes \exists x. \neg (E x) shows True — Nitpick finds a model<sup>2</sup> nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops
```

We may also assume an existing and a non-existing object and still get consistency.

```
lemma assumes (\exists x. \neg(E x)) \land (\exists x. (E x)) shows True — Nitpick finds a model nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops
```

The left-to-right direction of existence axiom E_i is implied.

```
lemma E_iImplied: E(x \cdot y) \to (E \ x \land E \ y \land (\exists \ z. \ z \cdot z \cong z \land x \cdot z \cong x \land z \cdot y \cong y)) by (metis \ A_i \ C_i \ S_i)
```

We can prove that the i in axiom C_i is unique. The proofs can be found automatically by Sledgehammer.³

```
lemma UC_i: \forall y.\exists i. \ I \ i \land i \cdot y \cong y \land (\forall j.(I \ j \land j \cdot y \cong y) \rightarrow i \cong j)
by (smt \ A_i \ C_i \ S_i)
```

Analogously, the provers quickly show that j in axiom D is unique.

```
lemma UD_i: \forall x. \exists j. I j \land x \cdot j \cong x \land (\forall i. (I i \land x \cdot i \cong x) \rightarrow j \cong i) by (smt \ A_i \ D_i \ S_i)
```

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

```
lemma (\exists C D. (\forall y. I (C y) \land (C y) \cdot y \cong y) \land (\forall x. I (D x) \land x \cdot (D x) \cong x) \land \neg (D = C))

nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops — Nitpick finds a model
```

Nitpick finds a model for cardinality i = 2. This model consists of two non-existing objects i_1 and i_2 . C maps both i_1 and i_2 to i_2 . D maps i_1 to i_2 , and vice versa. The composition $i_2 \cdot i_2$ is mapped to i_2 . All other composition pairs are mapped to i_1 .

Even if we require at least one existing object Nitpick still finds a model:

```
lemma (\exists x. E x) \land (\exists C D. (\forall y. I (C y) \land (C y) \cdot y \cong y) \land (\forall x. I (D x) \land x \cdot (D x) \cong x) \land \neg (D = C))
```

nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] **oops** — Nitpick finds a model.

Again the model is of cardinality i = 2, but now we have a non-existing i_1 and and an existing i_2 . Composition \cdot and C are as above, but D is now identity on all objects.

²To display the models or countermodels from Nitpick in the Isabelle/HOL system interface simply put the mouse on the expression "nitpick".

³In our initial experiments proof reconstruction of the external ATP proofs failed in Isabelle/HOL. The SMT reasoner Z3 [7], which is employed in the *smt* tactic by default, was too weak. Therefore we first introduced further lemmata, which helped. However, an alternative way out, which we discovered later, has been to replace Z3 by CVC4 [8] in Isabelle's *smt* tactic (this can be done by stating "declare [[*smt-solver* = cvc4]]" in the source document). In the latest version of the proof document we now suitably switch between the two SMT solvers to obtain best results.

5 Axiom Set II

Axiom Set II is developed from Axiom Set I by Skolemization of i and j in axioms C_i and D_i . We can argue semantically that every model of Axiom Set I has such functions. Hence, we get a conservative extension of Axiom 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: the values of Skolem functions outside E can just be given by the identity function.

```
S_{ii}: — Strictness:
                           (E(x \cdot y) \to (E \times A \times E y)) \wedge (E(dom \times X) \to E \times X) \wedge (E(cod \times Y) \to E \times Y) and
 E_{ii}: — Existence:
                           E(x \cdot y) \leftarrow (E \ x \land E \ y \land (\exists \ z. \ z \cdot z \cong z \land x \cdot z \cong x \land z \cdot y \cong y)) and
 A_{ii}: — Associativity: x \cdot (y \cdot z) \cong (x \cdot y) \cdot z and
                           E y \to (I(cod y) \land (cod y) \cdot y \cong y) and
 C_{ii}: — Codomain:
                           E x \to (I(dom \ x) \land x \cdot (dom \ x) \cong x)
 D_{ii}: — Domain:
As above, we first check for consistency.
  nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops
  lemma assumes \exists x. \neg (E x) shows True — Nitpick finds a model
    nitpick [satisfy, user-axioms, show-all, format = 2, expect = qenuine] oops
  lemma assumes (\exists x. \neg(E x)) \land (\exists x. (E x)) shows True — Nitpick finds a model
    nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops
The left-to-right direction of existence axiom E_{ii} is implied.
  by (metis 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.
lemma domTotal: E x \rightarrow E(dom x)
  by (metis D_{ii} S_{ii})
lemma codTotal: E x \rightarrow E(cod x)
  by (metis C_{ii} S_{ii})
Axiom Set II implies Axiom Set I.<sup>4</sup>
  lemma S_i From II: E(x \cdot y) \to (E \times A \times E y)
    using S_{ii} by blast
  lemma E_iFromII: E(x \cdot y) \leftarrow (E \ x \land E \ y \land (\exists \ z. \ z \cdot z \cong z \land x \cdot z \cong x \land z \cdot y \cong y))
    using E_{ii} by blast
  lemma A_i From II: x \cdot (y \cdot z) \cong (x \cdot y) \cdot z
    using A_{ii} by blast
  lemma C_i From II: \forall y. \exists i. I i \land i \cdot y \cong y
    by (metis C_{ii} S_{ii})
  lemma D_i From II: \forall x. \exists j. I j \land x \cdot j \cong x
    by (metis D_{ii} S_{ii})
```

⁴Axiom Set I also implies Axiom Set II. This can be shown by semantical means on the meta-level. We have also attempted to prove this equivalence within Isabelle/HOL, but so far without final success. However, we succeed to prove that the following holds: $\exists \ Cod \ Dom.\ ((\forall x\ y.\ (E(x\cdot y) \to (E\ x \land E\ y))) \land (\forall x\ y.\ E(x\cdot y) \leftrightarrow (E\ x \land E\ y \land (\exists\ z.\ z\cdot z \cong z \land x\cdot z \cong x \land z\cdot y \cong y))) \land (\forall x\ y\ z.\ x\cdot (y\cdot z) \cong (x\cdot y)\cdot z) \land (\forall\ y.\ I\ (Cod\ y) \land (Cod\ y)\cdot y \cong y) \land (\forall\ x.\ I\ (Dom\ x) \land x\cdot (Dom\ x) \cong x)$). Note that the inclusion of strictness of Cod and Dom is still missing.

6 Axiom Set III

In Axiom Set III the existence axiom E is simplified by taking advantage of the two new Skolem functions dom and cod.

```
S_{iii}: — Strictness:
                             (E(x \cdot y) \to (E \times x \wedge E \times y)) \wedge (E(dom \times x) \to E \times x) \wedge (E(cod \times y) \to E \times y) and
 E_{iii}: — Existence:
                              E(x \cdot y) \leftarrow (dom \ x \cong cod \ y \land E(cod \ y)) and
 A_{iii}: — Associativity: x \cdot (y \cdot z) \cong (x \cdot y) \cdot z and
                              E y \rightarrow (I(cod y) \land (cod y) \cdot y \cong y) and
 C_{iii}: — Codomain:
                              E x \rightarrow (I(dom \ x) \land x \cdot (dom \ x) \cong x)
 D_{iii}: — Domain:
The obligatory consistency check is positive.
  lemma True — Nitpick finds a model
    nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops
 lemma assumes \exists x. \neg (E x) shows True — Nitpick finds a model
    nitpick [satisfy, user-axioms, show-all, format = 2, expect = qenuine] oops
 lemma assumes (\exists x. \neg(E x)) \land (\exists x. (E x)) shows True — Nitpick finds a model
    nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops
The left-to-right direction of existence axiom E_{iii} is implied.
 lemma E_{iii}Implied: E(x \cdot y) \rightarrow (dom \ x \cong cod \ y \land E(cod \ y))
    by (metis (full-types) A_{iii} C_{iii} D_{iii} S_{iii})
Moreover, Axiom Set II is implied.
 lemma S_{ii} From III: (E(x \cdot y) \to (E \times E y)) \land (E(dom \times x) \to E x) \land (E(cod y) \to E y)
    using S_{iii} by blast
 lemma E_{ii}FromIII: E(x \cdot y) \leftarrow (E \ x \land E \ y \land (\exists \ z. \ z \cdot z \cong z \land x \cdot z \cong x \land z \cdot y \cong y))
    by (metis A_{iii} C_{iii} D_{iii} E_{iii} S_{iii})
 lemma A_{ii}FromIII: x \cdot (y \cdot z) \cong (x \cdot y) \cdot z
    using A_{iii} by blast
 lemma C_{ii}FromIII: E y \rightarrow (I(cod y) \land (cod y) \cdot y \cong y)
    using C_{iii} by auto
 lemma D_{ii}FromIII: E x \rightarrow (I(dom x) \land x \cdot (dom x) \cong x)
    using D_{iii} by auto
```

A side remark on the experiments: All proofs above and all proofs in the rest of this paper have been obtained fully automatically with the Sledgehammer tool in Isabelle/HOL. This tool interfaces to prominent first-order automated theorem provers such as CVC4 [8], Z3 [7], E [12] and Spass [3]. Remotely, also provers such as Vampire [10], or the higher-order provers Satallax [6] and LEO-II [2] can be reached. For example, to prove lemma E_{iii} From II we have called Sledgehammer on all postulated axioms of the theory: sledgehammer (S_{ii} E_{ii} A_{ii} C_{ii} D_{ii}). The provers then, via Sledgehammer, suggested to call trusted/verified tools in Isabelle/HOL with the exactly required dependencies they detected. In lemma E_{iii} From II, for example, all axioms from Axiom Set II are required. 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 verified Isabelle/HOL document in which all the proofs have nevertheless been contributed by automated theorem provers.

Axiom Set II also implies Axiom Set III. Hence, both theories are equivalent. The only interesting case is lemma E_{iii} From II, the other cases are trivial.

```
lemma S_{iii} From II: (E(x \cdot y) \to (E \ x \land E \ y)) \land (E(dom \ x) \to E \ x) \land (E(cod \ y) \to E \ y) using S_{ii} by blast lemma E_{iii} From II: E(x \cdot y) \leftarrow (dom \ x \cong cod \ y \land (E(cod \ y))) by (metis \ C_{ii} \ D_{ii} \ E_{ii} \ S_{ii}) lemma A_{iii} From II: x \cdot (y \cdot z) \cong (x \cdot y) \cdot z using A_{ii} by blast lemma C_{iii} From II: E \ y \to (I(cod \ y) \land (cod \ y) \cdot y \cong y) using C_{ii} by auto lemma D_{iii} From II: E \ x \to (I(dom \ x) \land x \cdot (dom \ x) \cong x) using D_{ii} by auto
```

7 Axiom Set IV

Axiom 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.

```
\begin{array}{lll} S_{iv} \colon & - \text{Strictness:} & (E(x \cdot y) \to (E \ x \land E \ y)) \land (E(dom \ x) \to E \ x) \land (E(cod \ y) \to E \ y) \text{ and} \\ E_{iv} \colon & - \text{Existence:} & E(x \cdot y) \leftrightarrow (dom \ x \cong cod \ y \land E(cod \ y)) \text{ and} \\ A_{iv} \colon & - \text{Associativity:} & x \cdot (y \cdot z) \cong (x \cdot y) \cdot z \text{ and} \\ C_{iv} \colon & - \text{Codomain:} & (cod \ y) \cdot y \cong y \text{ and} \\ D_{iv} \colon & - \text{Domain:} & x \cdot (dom \ x) \cong x \end{array}
```

The obligatory consistency check is again positive.

```
lemma True — Nitpick finds a model nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops lemma assumes \exists x. \neg (E x) shows True — Nitpick finds a model nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops lemma assumes (\exists x. \neg (E x)) \land (\exists x. (E x)) shows True — Nitpick finds a model nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops
```

The Axiom Set III is implied. The only interesting cases are lemmata $C_{iii}FromIV$ and $D_{iii}FromIV$. Note that the strengthened axiom E_{iv} is used here.

```
lemma S_{iii}FromIV: (E(x\cdot y) \to (E\ x \land E\ y)) \land (E(dom\ x) \to E\ x) \land (E(cod\ y) \to E\ y) using S_{iv} by blast lemma E_{iii}FromIV: E(x\cdot y) \leftarrow (dom\ x \cong cod\ y \land (E(cod\ y))) using E_{iv} by blast lemma A_{iii}FromIV: x\cdot (y\cdot z) \cong (x\cdot y)\cdot z using A_{iv} by blast lemma C_{iii}FromIV: E\ y \to (I(cod\ y) \land (cod\ y)\cdot y \cong y) by (metis\ C_{iv}\ D_{iv}\ E_{iv}) lemma D_{iii}FromIV: E\ x \to (I(dom\ x) \land x\cdot (dom\ x) \cong x) by (metis\ (full-types)\ C_{iv}\ D_{iv}\ E_{iv})
```

Vice versa, Axiom Set III implies Axiom Set IV. Hence, both theories are equivalent. The interesting cases are lemmata $E_{iv}FromIII$, $C_{iv}FromIII$ and $D_{iv}FromIII$.

```
\begin{array}{l} \mathbf{lemma} \ S_{iv}\mathit{From}\mathit{III} \colon (E(x \cdot y) \to (E \ x \land E \ y)) \land (E(\mathit{dom} \ x \ ) \to E \ x) \land (E(\mathit{cod} \ y) \to E \ y) \\ \mathbf{using} \ S_{iii} \ \mathbf{by} \ \mathit{blast} \\ \mathbf{lemma} \ E_{iv}\mathit{From}\mathit{III} \colon E(x \cdot y) \leftrightarrow (\mathit{dom} \ x \cong \mathit{cod} \ y \land E(\mathit{cod} \ y)) \\ \mathbf{by} \ (\mathit{metis} \ (\mathit{full-types}) \ A_{iii} \ C_{iii} \ D_{iii} \ E_{iii} \ S_{iii}) \\ \mathbf{lemma} \ A_{iv}\mathit{From}\mathit{III} \colon x \cdot (y \cdot z) \cong (x \cdot y) \cdot z \\ \mathbf{using} \ A_{iii} \ \mathbf{by} \ \mathit{blast} \end{array}
```

```
lemma C_{iv} From III: (cod\ y) \cdot y \cong y

using C_{iii} S_{iii} by blast

lemma D_{iv} From III: x \cdot (dom\ x) \cong x

using D_{iii} S_{iii} by blast
```

8 Axiom Set V

Axiom Set V has been proposed by Scott [14] in the 1970s. This set of axioms is equivalent to the axiom set presented by Freyd and Scedrov in their textbook "Categories, Allegories" [9] when encoded in free logic, corrected/adapted and further simplified. Their axiom set is technically flawed when encoded in our given context. This issue has been detected by automated theorem provers with the same technical infrastructure as employed so far. See the subsequent section for more details. We have modified the axioms of [9] by replacing the original Kleene equality \cong in axiom S3 by the non-reflexive, existing identity \cong . Note that the modified axiom S3 is equivalent to E_{iv} ; see the mutual proofs below.

```
\begin{array}{lll} S1: & - & \text{Strictness:} & E(dom \ x) \rightarrow E \ x \ \text{ and} \\ S2: & - & \text{Strictness:} & E(cod \ y) \rightarrow E \ y \ \text{and} \\ S3: & - & \text{Existence:} & E(x \cdot y) \leftrightarrow dom \ x \simeq cod \ y \ \text{and} \\ S4: & - & \text{Associativity:} & x \cdot (y \cdot z) \cong (x \cdot y) \cdot z \ \text{and} \\ S5: & - & \text{Domain:} & x \cdot (dom \ x) \cong x \ \text{and} \\ S6: & - & \text{Codomain:} & (cod \ y) \cdot y \cong y \end{array}
```

The obligatory consistency check is again positive.

```
lemma True — Nitpick finds a model

nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops

lemma assumes \exists x. \neg(E x) shows True — Nitpick finds a model

nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops

lemma assumes (\exists x. \neg(E x)) \land (\exists x. (E x)) shows True — Nitpick finds a model

nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops
```

The Axiom Set IV is implied. The only interesting cases are lemmata $E_{iv}FromV$ and $E_{iv}FromV$.

```
\begin{array}{l} \mathbf{lemma} \ S_{iv}FromV\colon (E(x\cdot y)\to (E\ x\wedge E\ y)) \wedge (E(dom\ x\ )\to E\ x) \wedge (E(cod\ y)\to E\ y) \\ \mathbf{using} \ S1\ S2\ S3\ \mathbf{by} \ blast \\ \mathbf{lemma} \ E_{iv}FromV\colon E(x\cdot y) \leftrightarrow (dom\ x\cong cod\ y\wedge E(cod\ y)) \\ \mathbf{using} \ S3\ \mathbf{by} \ metis \\ \mathbf{lemma} \ A_{iv}FromV\colon x\cdot (y\cdot z)\cong (x\cdot y)\cdot z \\ \mathbf{using} \ S4\ \mathbf{by} \ blast \\ \mathbf{lemma} \ C_{iv}FromV\colon (cod\ y)\cdot y\cong y \\ \mathbf{using} \ S6\ \mathbf{by} \ blast \\ \mathbf{lemma} \ D_{iv}FromV\colon x\cdot (dom\ x)\cong x \\ \mathbf{using} \ S5\ \mathbf{by} \ blast \end{array}
```

Vice versa, Axiom Set IV implies Axiom Set V. Hence, both theories are equivalent.

```
lemma S1FromV: E(dom\ x) \to E\ x

using S_{iv} by blast

lemma S2FromV: E(cod\ y) \to E\ y

using S_{iv} by blast

lemma S3FromV: E(x\cdot y) \leftrightarrow dom\ x \simeq cod\ y

using E_{iv} by metis
```

```
lemma S4FromV\colon x{\cdot}(y{\cdot}z)\cong (x{\cdot}y){\cdot}z

using A_{iv} by blast

lemma S5FromV\colon x{\cdot}(dom\ x)\cong x

using D_{iv} by blast

lemma S6FromV\colon (cod\ y){\cdot}y\cong y

using C_{iv} by blast
```

9 Axiom Sets VI and VII

The axiom set of Freyd and Scedrov from their textbook "Categories, Allegories" [9] becomes inconsistent in our free logic setting if we assume non-existing objects of type i, respectively, if we assume that the operations are non-total. 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 $(f \cdot g)\ x \cong g(f\ x)$ (functional composition from left to right) instead of the set-theoretic definition $(f \cdot g)\ x \cong f(g\ x)$ (functional composition from right to left) used so far.

We leave it to the reader to verify that their axiom system corresponds to the axiom system given below modulo an appropriate conversion of notation.⁵ In Subsection 9.2 we will also analyze their axiom system using their original notation.

A main difference in the system by Freyd and Scedrov to our Axiom Set V from above concerns axiom S3. Namely, instead of the non-reflexive \simeq , they use Kleene equality \cong , cf. definition 1.11 on page 3 of [9].⁶ The difference seems minor, but in our free logic setting it has the effect to cause the mentioned constricted inconsistency issue. This could perhaps be an oversight, or it could indicate that Freyd and Scedrov actually mean the Axiom Set VIII below (where the variables in the axioms range over defined objects only). However, in Axiom Set VIII we had to (re-)introduce explicit strictness conditions to ensure equivalence to the Axiom Set V by Scott.

9.1 Axiom Set VI

```
A1: E(x \cdot y) \leftrightarrow dom \ x \simeq cod \ y \ \text{and}

A2a: cod(dom \ x) \cong dom \ x \ \text{and}

A2b: dom(cod \ y) \cong cod \ y \ \text{and}

A3a: x \cdot (dom \ x) \cong x \ \text{and}

A3b: (cod \ y) \cdot y \cong y \ \text{and}

A4a: dom(x \cdot y) \cong dom((dom \ x) \cdot y) \ \text{and}

A4b: cod(x \cdot y) \cong cod(x \cdot (cod \ y)) \ \text{and}

A5: x \cdot (y \cdot z) \cong (x \cdot y) \cdot z
```

The obligatory consistency checks are again positive. But note that this only holds when we use \simeq instead of \cong in A1.

```
lemma True — Nitpick finds a model nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops lemma assumes \exists x. \neg (E x) shows True — Nitpick finds a model
```

⁵A recipe for this translation is as follows: (i) replace all $x \cdot y$ 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. $dom x \simeq cod y$).

⁶Def. 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. ..."

```
nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops lemma assumes (\exists x. \neg(E x)) \land (\exists x. (E x)) shows True — Nitpick finds a model nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops
```

Axiom Set VI implies Axiom Set V.

```
lemma S1FromVI: E(dom\ x) \to E\ x by (metis\ A1\ A2a\ A3a) lemma S2FromVI: E(cod\ y) \to E\ y using A1\ A2b\ A3b by metis lemma S3FromVI: E(x\cdot y) \leftrightarrow dom\ x \simeq cod\ y by (metis\ A1) lemma S4FromVI: x\cdot (y\cdot z) \cong (x\cdot y)\cdot z using A5 by blast lemma S5FromVI: x\cdot (dom\ x) \cong x using A3a by blast lemma S6FromVI: (cod\ y)\cdot y \cong y using A3b by blast
```

Note, too, that Axiom Set VI is redundant. For example, axioms $A \not = a$ and $A \not = b$ are implied from the others. This kind of flaw in presenting axioms in our view is a more serious oversight. The automated theorem provers can quickly reveal such redundancies.

```
lemma A4aRedundant: dom(x \cdot y) \cong dom((dom \ x) \cdot y)

using A1\ A2a\ A3a\ A5 by metis

lemma A4bRedundant: cod(x \cdot y) \cong cod(x \cdot (cod\ y))

using A1\ A2b\ A3b\ A5 by metis
```

Our attempts to further reduce the axioms set $(A1 \ A2a \ A2b \ A3a \ A3b \ A5)$ were not successful. Alternatively, we can e.g. keep A4a and A4b and show that axioms A2a and A2b are implied.

```
lemma A2aRedundant: cod(dom\ x) \cong dom\ x

using A1\ A3a\ A3b\ A4a\ A4b by smt

lemma A2bRedundant: dom(cod\ y) \cong cod\ y

using A1\ A3a\ A3b\ A4a\ A4b by smt
```

Again, attempts to further reduce the set (A1 A3a A3b A4a A4b A5) were not successful. Other reduced sets of axioms we identified in experiments are (A1 A2a A3a A3b A4b A5) and (A1 A2b A3a A3b A4a A5). 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 ($A1 \ A3a \ A3b \ A5$) as a minimal axiom set, which then exactly matches the Axiom Set V from above.⁷

Axiom Set V implies Axiom Set VI. Hence, both theories are equivalent.

```
lemma A1FromV \colon E(x \cdot y) \leftrightarrow dom \ x \simeq cod \ y using S3 by blast lemma A2aFromV \colon cod(dom \ x) \cong dom \ x by (metis \ S1 \ S2 \ S3 \ S5) lemma A2bFromV \colon dom(cod \ y) \cong cod \ y using S1 \ S2 \ S3 \ S6 by metis
```

⁷This minimal set of axioms is also mentioned by Freyd in [?] and attributed to Martin Knopman. However, the proof sketch presented there seems to fail when the adapted version of A1 (with \simeq) is employed.

```
lemma A3aFromV \colon x \cdot (dom\ x) \cong x

using S5 by blast

lemma A3bFromV \colon (cod\ y) \cdot y \cong y

using S6 by blast

lemma A4aFromV \colon dom(x \cdot y) \cong dom((dom\ x) \cdot y)

by (metis\ S1\ S3\ S4\ S5\ S6)

lemma A4bFromV \colon cod(x \cdot y) \cong cod(x \cdot (cod\ y))

by (metis\ S2\ S3\ S4\ S5\ S6)

lemma A5FromV \colon x \cdot (y \cdot z) \cong (x \cdot y) \cdot z

using S4 by blast
```

9.2 Axiom Set VII

We now study the constricted inconsistency in Axiom Set VI when replacing \simeq in A1 by \cong . We call this Axiom Set VII. This set corresponds modulo representational transformation to the axioms as presented by Freyd and Scedrov. Remember, however, that the free variables are ranging here over all objects, defined or undefined. Below, when we study Axiom Set VIII, we will restrict the variables to range only over existing objects.

```
A1: E(x \cdot y) \leftrightarrow dom \ x \cong cod \ y \ and

A2a: cod(dom \ x) \cong dom \ x \ and

A2b: dom(cod \ y) \cong cod \ y \ and

A3a: x \cdot (dom \ x) \cong x \ and

A3b: (cod \ y) \cdot y \cong y \ and

A4a: dom(x \cdot y) \cong dom((dom \ x) \cdot y) \ and

A4b: cod(x \cdot y) \cong cod(x \cdot (cod \ y)) \ and

A5: x \cdot (y \cdot z) \cong (x \cdot y) \cdot z
```

A model can still be constructed if we do not make assumptions about non-existing objects. In fact, the model presented by Nitpick consists of a single, existing morphism.

```
\mathbf{lemma} \ \mathit{True}
```

nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] **oops** — Nitpick finds a model

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.

We shall see that 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 [9], the category reduces to a monoid provided that it is not empty.)

```
lemma assumes \exists x. \neg(E x) shows True — Nitpick does *not* find a model nitpick [satisfy, user-axioms, show-all, format = 2, expect = none] oops
```

In fact, the automated theorem provers quickly prove falsity when assuming a non-existing object of type i. The provers identify the axioms A1, A2a and A3a to cause the problem under this assumption.

lemma $InconsistencyAutomaticVII: (\exists x. \neg(E x)) \rightarrow False$

```
by (metis A1 A2a A3a)
```

Hence, all morphisms must be defined in theory of Axiom Set VII, or in other words, all operations must be total.

lemma $\forall x. \ E \ x \ using \ Inconsistency Automatic VII \ by \ auto$

The constricted inconsistency proof can be turned into an interactive mathematical argument:

```
lemma InconsistencyInteractiveVII:
   assumes NEx: \exists x. \neg (E x) shows False
 proof -
    — Let a be an undefined object
  obtain a where 1: \neg(E \ a) using NEx by auto
     — We instantiate axiom A3a with a.
  have 2: a \cdot (dom \ a) \cong a  using A3a by blast
    By unfolding the definition of \cong we get from 1 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.
   have 3: \neg(E(a \cdot (dom\ a))) using 1 2 by metis
     – We instantiate axiom A1 with a and dom a.
   have 4: E(a \cdot (dom \ a)) \leftrightarrow dom \ a \cong cod(dom \ a) using A1 by blast
      We instantiate axiom A2a with a.
   have 5: cod(dom \ a) \cong dom \ a \ using \ A2a \ by \ blast
    — We use 5 (and symmetry and transitivity of \cong) to rewrite the right-hand of the equivalence 4
into dom \ a \cong dom \ a.
   have 6: E(a \cdot (dom \ a)) \leftrightarrow dom \ a \cong dom \ a \text{ using 4 5 by } auto
     By reflexivity of \cong we get that a \cdot (dom \ a) must be defined.
   have 7: E(a \cdot (dom \ a)) using 6 by blast
   — We have shown in 7 that a \cdot (dom\ a) is defined, and in 3 that it is undefined. Contradiction.
  then show ?thesis using 7 3 by blast
 qed
```

We present the constricted inconsistency argument once again, but this time in the original notation of Freyd and Scedrov.

consts

```
source:: i \Rightarrow i \ (\Box - [108] \ 109)
target:: i \Rightarrow i \ (\neg \Box [110] \ 111)
compositionF:: i \Rightarrow i \Rightarrow i \ (infix \cdot 110)

A1: E(x \cdot y) \leftrightarrow (x \Box \cong \Box y) \text{ and}
A2a: ((\Box x)\Box) \cong \Box x \text{ and}
A2b: \Box (x \Box) \cong \Box x \text{ and}
A3b: \Box (x \Box) \cong x \text{ and}
A3b: x \cdot (x \Box) \cong x \text{ and}
A4a: \Box (x \cdot y) \cong \Box (x \cdot (\Box y)) \text{ and}
A4b: (x \cdot y)\Box \cong ((x \Box) \cdot y)\Box \text{ and}
A5: x \cdot (y \cdot z) \cong (x \cdot y) \cdot z
```

Again, the automated theorem provers via Sledgehammer find the constricted inconsistency very quickly and they identify the exact dependencies.

```
lemma InconsistencyAutomatic: (\exists x. \neg(E x)) \rightarrow False by (metis\ A1\ A2a\ A3a)
```

The following alternative interactive proof is slightly shorter than the one presented above.

```
lemma InconsistencyInteractive: assumes NEx: \exists x. \neg (E x) shows False
      Let a be an undefined object
  obtain a where 1: \neg(E \ a) using assms by auto
      We instantiate axiom A3a with a.
  have 2: (\Box a) \cdot a \cong a using A3a by blast
       By unfolding the definition of \cong we get from 1 that (\Box a) \cdot a is not defined. This is easy to see,
since if (\Box a) \cdot a were defined, we also had that a is defined, which is not the case by assumption.
  have 3: \neg (E((\Box a) \cdot a)) using 1 2 by metis
      We instantiate axiom A1 with \Box a and a.
  have A: E((\Box a) \cdot a) \leftrightarrow (\Box a) \Box \cong \Box a using A1 by blast
    — We instantiate axiom A2a with a.
  have 5: (\Box a)\Box \cong \Box a using A2a by blast
    — From 4 and 5 we obtain (E((\Box a) \cdot a)) by propositional logic.
  have \theta: E((\Box a) \cdot a) using 4 5 by blast
   — We have \neg(E((\Box a) \cdot a)) and E((\Box a) \cdot a), hence Falsity.
  then show ?thesis using 6 3 by blast
 qed
```

Obviously Axiom Set VII is also redundant, and we have previously reported on respective redundancies [1]. However, this was before the discovery of the above constricted inconsistency issue, which tells us that the system (in our setting) can even be reduced to A1, A2a and A3a (when we additionally assume NEx).

10 Axiom Set VIII

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. By employing the free logic universal quantifier \forall we thus modify Axiom Set VII as follows:

```
B1: \forall x. \forall y. \ E(x\cdot y) \leftrightarrow dom \ x \cong cod \ y \ and
B2a: \forall x. \ cod(dom \ x) \cong dom \ x \ and
B2b: \forall y. \ dom(cod \ y) \cong cod \ y \ and
B3a: \forall x. \ x\cdot (dom \ x) \cong x \ and
B3b: \forall y. \ (cod \ y) \cdot y \cong y \ and
B4a: \forall x. \forall y. \ dom(x\cdot y) \cong dom((dom \ x)\cdot y) \ and
B4b: \forall x. \forall y. \ cod(x\cdot y) \cong cod(x\cdot (cod \ y)) \ and
B5: \forall x. \forall y. \forall z. \ x\cdot (y\cdot z) \cong (x\cdot y)\cdot z
Now, the two consistency checks succeed.

lemma True — Nitpick finds a model
nitpick [satisfy, \ user-axioms, \ show-all, \ format = 2, \ expect = genuine] oops
lemma assumes \exists x. \ \neg(Ex) \ shows \ True — Nitpick finds a model
nitpick [satisfy, \ user-axioms, \ show-all, \ format = 2, \ expect = genuine] oops
lemma assumes (\exists x. \ \neg(Ex)) \land (\exists x. \ (Ex)) \ shows \ True — Nitpick finds a model
nitpick [satisfy, \ user-axioms, \ show-all, \ format = 2, \ expect = genuine] oops
```

However, this axiom set is obviously weaker than our Axiom Set V. In fact, none of the V-axioms are implied:

```
lemma S1: E(dom\ x) \to E\ x — Nitpick finds a countermodel nitpick [user-axioms, show-all, format = 2] oops
```

```
lemma S2: E(cod\ y) \rightarrow E\ y — Nitpick finds a countermodel
    nitpick [user-axioms, show-all, format = 2] oops
 lemma S3: E(x \cdot y) \leftrightarrow dom \ x \simeq cod \ y — Nitpick finds a countermodel
    nitpick [user-axioms, show-all, format = 2] oops
 lemma S4: x \cdot (y \cdot z) \cong (x \cdot y) \cdot z — Nitpick finds a countermodel
   nitpick [user-axioms, show-all, format = 2] oops
 lemma S5: x \cdot (dom \ x) \cong x — Nitpick finds a countermodel
   nitpick [user-axioms, show-all, format = 2] oops
 lemma S6: (cod\ y) \cdot y \cong y — Nitpick finds a countermodel
   nitpick [user-axioms, show-all, format = 2] oops
The situation changes when we explicitly postulate strictness of dom, cod and . We thus
obtain our Axiom Set VIII:
 B0a: E(x \cdot y) \rightarrow (E \ x \land E \ y) and
 B0b: E(dom\ x) \rightarrow E\ x and
 B0c: E(cod\ x) \to E\ x and
 B1: \forall x. \forall y. \ E(x \cdot y) \leftrightarrow dom \ x \cong cod \ y \ and
 B2a: \forall x. \ cod(dom \ x) \cong dom \ x \  and
 B2b: \forall y. \ dom(cod \ y) \cong cod \ y \ and
 B3a: \forall x. \ x \cdot (dom \ x) \cong x \ \text{and}
 B3b: \forall y. (cod y) \cdot y \cong y \text{ and }
 B4a: \forall x. \forall y. \ dom(x \cdot y) \cong dom((dom \ x) \cdot y) and
 B4b: \forall x. \forall y. \ cod(x \cdot y) \cong cod(x \cdot (cod y)) and
  B5: \forall x. \forall y. \forall z. \ x \cdot (y \cdot z) \cong (x \cdot y) \cdot z
Again, the two consistency checks succeed
 lemma True — Nitpick finds a model
    nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops
 lemma assumes \exists x. \neg (E x) shows True — Nitpick finds a model
   nitpick [satisfy, user-axioms, show-all, format = 2, expect = genuine] oops
 lemma assumes (\exists x. \neg(E x)) \land (\exists x. (E x)) shows True — Nitpick finds a model
    nitpick [satisfy, user-axioms, show-all, format = 2, expect = qenuine] oops
Now Axiom Set V is implied.
 lemma S1From VIII: E(dom\ x) \rightarrow E\ x using B0b by blast
 lemma S2FromVIII: E(cod\ y) \rightarrow E\ y using B0c by blast
 lemma S3FromVIII: E(x \cdot y) \leftrightarrow dom \ x \simeq cod \ y by (metis B0a B0b B0c B1 B3a)
 lemma S4FromVIII: x \cdot (y \cdot z) \cong (x \cdot y) \cdot z by (meson B0a B5)
 lemma S5FromVIII: x \cdot (dom \ x) \cong x using B0a B3a by blast
 lemma S6FromVIII: (cod\ y)\cdot y\cong y using B0a B3b by blast
Vive versa, Axiom Set V implies Axiom Set VIII. Hence, both theories are equivalent.
                          E(dom \ x) \rightarrow E \ x \ and
 S1: — Strictness:
 S2: — Strictness:
                          E(cod\ y) \rightarrow E\ y and
 S3: — Existence:
                          E(x \cdot y) \leftrightarrow dom \ x \simeq cod \ y and
 S4: — Associativity: x \cdot (y \cdot z) \cong (x \cdot y) \cdot z and
 S5: — Domain:
                          x \cdot (dom \ x) \cong x \ \mathbf{and}
 S6: — Codomain:
                          (cod\ y)\cdot y\cong y
 lemma B0a: E(x \cdot y) \to (E \times A \times E y) using S1 S2 S3 by blast
```

lemma B0b: $E(dom\ x) \to E\ x$ using S1 by blast lemma B0c: $E(cod\ x) \to E\ x$ using S2 by blast

```
lemma B1: \forall x. \forall y. \ E(x \cdot y) \leftrightarrow dom \ x \cong cod \ y \ by \ (metis \ S3 \ S5)
lemma B2a: \forall x. \ cod \ (dom \ x) \cong dom \ x \ by \ (metis \ S3 \ S5)
lemma B2b: \forall y. \ dom \ (cod \ y) \cong cod \ y \ by \ (metis \ S3 \ S6)
lemma B3a: \forall x. \ x \cdot (dom \ x) \cong x \ using \ S5 \ by \ auto
lemma B3b: \forall y. \ (cod \ y) \cdot y \cong y \ using \ S6 \ by \ blast
lemma B4a: \forall x. \forall y. \ dom \ (x \cdot y) \cong dom \ ((dom \ x) \cdot y) \ by \ (metis \ S1 \ S3 \ S4 \ S5)
lemma B4b: \forall x. \forall y. \ cod \ (x \cdot y) \cong cod \ (x \cdot (cod \ y)) \ by \ (metis \ S2 \ S3 \ S4 \ S6)
lemma B5: \forall x. \forall y. \forall z. \ x \cdot (y \cdot z) \cong (x \cdot y) \cdot z \ using \ S4 \ by \ blast
```

Axiom Set VIII is redundant (as expected from previous observations). The theorem provers quickly confirm that axioms B2a, B2b, B4a, B4b are implied.

```
B0a: E(x \cdot y) \rightarrow (E \ x \land E \ y) \ \text{and}
B0b: E(dom \ x) \rightarrow E \ x \ \text{and}
B0c: E(cod \ x) \rightarrow E \ x \ \text{and}
B1: \forall x . \forall y . E(x \cdot y) \leftrightarrow dom \ x \cong cod \ y \ \text{and}
B3a: \forall x . \ x \cdot (dom \ x) \cong x \ \text{and}
B3b: \forall y . (cod \ y) \cdot y \cong y \ \text{and}
B5: \forall x . \forall y . \forall z . \ x \cdot (y \cdot z) \cong (x \cdot y) \cdot z
\mathbf{lemma} \ B2aRedundant: \forall x . \ cod(dom \ x) \cong dom \ x \ \mathbf{by} \ (metis \ B0a \ B1 \ B3a)
\mathbf{lemma} \ B2bRedundant: \forall x . \ dom(cod \ y) \cong cod \ y \ \mathbf{by} \ (metis \ B0a \ B1 \ B3b)
\mathbf{lemma} \ B4aRedundant: \forall x . \forall y . \ dom(x \cdot y) \cong dom((dom \ x) \cdot y) \ \mathbf{by} \ (metis \ B0a \ B0b \ B1 \ B3a \ B5)
\mathbf{lemma} \ B4bRedundant: \forall x . \forall y . \ cod(x \cdot y) \cong cod(x \cdot (cod \ y)) \ \mathbf{by} \ (metis \ B0a \ B0c \ B1 \ B3b \ B5)
```

Again, note the relation and similarity of the reduced Axiom Set VIII to Axiom Set V by Scott, which we prefer, since it avoids a mixed use of free and bound variables in the encoding and since it is smaller.

Acknowledgements We thank Günter Rote and Lutz Schröder for their valuable comments to earlier drafts of this paper.

References

- [1] Christoph Benzmüller and Dana Scott. Automating free logic in Isabelle/HOL. In G.-M. Greuel, T. Koch, P. Paule, and A. Sommese, editors, *Mathematical Software ICMS 2016, 5th International Congress, Proceedings*, volume 9725 of *LNCS*, pages 43–50, Berlin, Germany, 2016. Springer.
- [2] Christoph Benzmüller, Nik Sultana, Lawrence C. Paulson, and Frank Theiss. The higher-order prover Leo-II. *J. Autom. Reasoning*, 55(4):389–404, 2015.
- [3] Jasmin Christian Blanchette, Andrei Popescu, Daniel Wand, and Christoph Weidenbach. More SPASS with isabelle superposition with hard sorts and configurable simplification. In *Interactive Theorem Proving Third International Conference, ITP 2012, Princeton, NJ, USA, August 13-15, 2012. Proceedings*, volume 7406 of *Lecture Notes in Computer Science*, pages 345–360. Springer, 2012.
- [4] J.C. Blanchette, S. Böhme, and L.C. Paulson. Extending Sledgehammer with SMT solvers. *J. of Automated Reasoning*, 51(1):109–128, 2013.

- [5] J.C. Blanchette and T. Nipkow. Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In *ITP 2010*, number 6172 in LNCS, pages 131–146. Springer, 2010.
- [6] Chad E. Brown. Satallax: An automatic higher-order prover. In Automated Reasoning 6th International Joint Conference, IJCAR 2012, Manchester, UK, June 26-29, 2012. Proceedings, volume 7364 of Lecture Notes in Computer Science, pages 111–117. Springer, 2012.
- [7] Leonardo Mendonça de Moura and Nikolaj Bjørner. Z3: an efficient SMT solver. In Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings, volume 4963 of Lecture Notes in Computer Science, pages 337–340. Springer, 2008.
- [8] Morgan Deters, Andrew Reynolds, Tim King, Clark W. Barrett, and Cesare Tinelli. A tour of CVC4: how it works, and how to use it. In *Formal Methods in Computer-Aided Design*, FMCAD 2014, Lausanne, Switzerland, October 21-24, 2014, page 7. IEEE, 2014.
- [9] Peter J. Freyd and Andre Scedrov. Categories, Allegories. North Holland, 1990.
- [10] Laura Kovács and Andrei Voronkov. First-order theorem proving and vampire. In Computer Aided Verification 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings, volume 8044 of Lecture Notes in Computer Science, pages 1-35. Springer, 2013.
- [11] T. Nipkow, L.C. Paulson, and M. Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Number 2283 in LNCS. Springer, 2002.
- [12] Stephan Schulz. System description: E 1.8. In Logic for Programming, Artificial Intelligence, and Reasoning 19th International Conference, LPAR-19, Stellenbosch, South Africa, December 14-19, 2013. Proceedings, volume 8312 of Lecture Notes in Computer Science, pages 735–743. Springer, 2013.
- [13] Dana 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).
- [14] Dana Scott. Identity and existence in intuitionistic logic. In Michael Fourman, Christopher Mulvey, and Dana Scott, editors, Applications of Sheaves: Proceedings of the Research Symposium on Applications of Sheaf Theory to Logic, Algebra, and Analysis, Durham, July 9–21, 1977, volume 752 of Lecture Notes in Mathematics, pages 660–696. Springer Berlin Heidelberg, 1979.
- [15] Makarius Wenzel. The isabelle system manual. https://www.cl.cam.ac.uk/research/hvg/Isabelle/dist/Isabelle2016/doc/system.pdf, February 2016.