The Category of Categories

Roger Bishop Jones

Abstract

Explorations into the possibility of contructing non-well-founded foundations systems which are ontologically category theoretic and include a category of all categories.

Created 2006/09/11

Last Checkin Date: 2010/08/13 10:57:22

http://www.rbjones.com/rbjpub/pp/doc/t018.pdf

Id: t018.doc,v 1.13 2010/08/13 10:57:22 rbj Exp

Contents

To Do

- This document was started when I was working with NF and NFU, and is for the sake of simplitіcy based directly on the formalisation of NFU following Holmes [4]. I have now come to a better understanding of the difficulties in working with NFU in ProofPower and am inclined to adopt a different non-well-founded set theory.

The next time I do any work on this, I am therefore likely to decouple it from any specific set theory, and define it as a construction on an arbitrary set theory.

Another aspect of this work which I now regard as obsolete is my attempt to make something of pseudo-induction (which I never really understood). In my subsequent work on non-well-founded set theories I have abandoned pseudo-induction. In particular my work on infinitarily definable non-well-founded sets yeilds its own kinds of induction over non-well-founded sets.

References

[1] T.E. Forster. *Set Theory with a Universal Set - Exploring an Untyped Universe.* Oxford University Press, 1992.

[2] M.R. Holmes. *Elementary Set Theory with a Universal Set*, volume 10 of *Cahier du Centre de Logique.* 1998.

[3] Roger Bishop Jones. Inductive, Co-Inductive and Psuedo-(Co-)Inductive Definitions in ProofPower. *RBJones.com*, 2010. `http://www.rbjones.com/rbjpub/pp/doc/t007.pdf`.

[4] Roger Bishop Jones. NF and NFU in ProofPower-HOL. *RBJones.com*, 2010. `http://www.rbjones.com/rbjpub/pp/doc/t020.pdf`.

[5] Roger Bishop Jones. PolySet Theory. *RBJones.com*, 2010. `http://www.rbjones.com/rbjpub/pp/doc/t020.pdf`.

1 Prelude

In well-founded set theories there is no category of (all) categories. This is one of a wider collection of issues which might be raised against the foundation of mathematics exclusively on well-founded sets. More commonly perhaps, the failure in well-founded set theories of the notion of cardinal numbers as equivalence classes under equipollence may be thought a primary motivation for considering non-well-founded set theories. Here we investigate non-well-founded ontologies without seeking new ways of representing numbers, but rather in order to have richer ontological support for abstract mathematics.

There are two aims for this document concerned with the category of categories. The more ambitious and exotic, perhaps even frivolous, is the construction of a category theoretic foundational ontology which might serve as an alternative to the established ontology of well-founded sets. This itself does not demand a departure from well-foundedness, but in our case we add the additional requirement that there be a category of all categories and more generally that the distinction between large and small categories which is necessary in well-founded set theories are rendered nugatory by the existence of categories encompassing the totality of relevant kinds of mathematical structures (e.g. a category corresponding to each kind of algebra).

Though in this we aim for a category theoretic foundation which stands on its own (rather than being based on a set theory) we need a meta-language to describe this system, and the ontology of that meta-language will include a non-well-founded set theory. This set theory will also be expected to have a category of categories.

The exploration involves more than one set theoretic starting point, and more than one way of constructing concrete category theoretic structures from the chosen set theories. The interest is primarily in systems which combine a substantial collection of well-founded entities with a non-well-founded ontology including a universal set or category.

2 INTRODUCTION

The material here concerns non-well-founded formal foundations for mathematics.

It builds on formalisations of non-well-founded set theories in [4, 5] some based on books by Forster [1] and Holmes [2] and engages in a process of ontological transformation, whereby the ontology for an established foundation system is used to construct an alternative ontology, for which a suitable axiomatisation is formally established.

The rationale here is as follows:

- It is natural, even in an untyped universe, to abstract away from the detailed coding of functions as sets by treating both sets and functions as primitive.
- In set theory we find a foundational ontology which is as simple as can be (i.e. sets are completely unstructured collections, all other ontologies involve working with structures which have more information in them than the collection of things from which they are built. This presumably has many advantages in the investigation of foundational problems. It may be however, that ontologies of more structured entities are pragmatically superior in applications. Sets and functions are special cases of Categories (which are sets with some structure) and functors (more generally arrows) which are functions which preserve some structure. This more complex ontology subsumes in a very direct way the more primitive set/function ontology. A set is a discrete category, a function is a functor whose domain is discrete.

- all the above has nothing to do with universal sets or well-foundedness. The kind of construction described is easily done in a well-founded set theory, but I have found the prospective advantages in a well-founded context insufficient motivation to carry through from the construction to development of the resulting theory. Partly this is because category theory more conspicuously than set theory screams against the constraint of well-foundedness. When done in a non=well-founded context, the additional merit of a category theory in which talk about size becomes less pervasive is added to the incentive to see how the theory works out.

There are several approaches to this problem addressed in this document. Some are based on the axiomatisation of NFU in [4], and some on the model for a theory of poly-sets in [5]. The former appear in section 2.1 the latter in section 2.2.

2.1 Approaches Based on NFU

Two approaches to this construction are presented, the first based on the notion of pseudo-well-foundedness presented in [1] and the second a "Co-Inductive" definition using methods from [3], one other is defined, *en passant* i.e. the well-founded inductive definition (which differs from the co-inductive case only in taking a least instead of a greatest fixed point), but taken not further. One more case might merit consideration if we had decided to take pseudo-induction more seriously, and that is pseudo-co-induction (again taking a greatest rather than a least fixed point).

I will spell this out informally a little more fully.

Let us begin considering the hereditarily finite sets. A hereditarily finite set is a finite set whose *members* are hereditarily finite set, nothing else is. This is an inductive definition, in higher order set theory we can define them as the intersection of all collections of sets which are closed under the formation of finite sets. One can imagine doing this for any property of sets, there is nothing special about "finite".

Now consider the "hereditarily functional sets". A set is functional if it is a set of ordered pairs which considered as a relation is many-one. Too close a parallel with the description of "hereditarily finite" fails in this case. If we say: "the hereditarily functional sets are those functional sets all of whose members are hereditarily functional sets" we are immediately in trouble, because the members of a function are ordered pairs, not functions.

So in this case we need to generalise from "member" to constituent, the notion of constituent being a parameter along with the property. In this case the constituents are the things in the domain and range of the function.

So now we say "the 'hereditarily functional sets' are those many-one relations whose field consists entirely of hereditarily functional sets". This elaboration of the idea of 'hereditarily P' sets provides a semantic way of getting new foundation systems from old, the theory of hereditarily functional sets would make a foundation system similar in strength to the set theory on which its definition is based, and can be independently axiomatised, throwing away all reference to the original set theory. To suggest that the resulting theory is a peer to set theory rather than remaining in some way parasitic upon it, I observe that by a broadly similar construction the original set theory can be constructed from the new function theory, as the 'hereditarily empty set valued functions" (i.e. the functions which always return the empty set, differing only in their domain, and which can therefore be taken as representatives of their domains).

Though there is some attraction in a foundation system based on functions rather than sets, this particular one (an untyped theory involving only well-founded functions) has attracted little interest. Unfortunately the only "problem" it appears to fix is the awkwardness of coding up functions as graphs using sets, and this it replaces with the awkwardness of representing sets by special kinds of function.

We really need both sets and functions, ideally perhaps without coding either, so one might attempt an inductive construction which involved both concepts and yielded a two-sorted theory. However, its more interesting to go one step further and consider sets and functions as special cases of concrete categories and functors. This I have previously done in the well-founded case (at least, so far as defining the domains is concerned). The objective here is to do something similar, without the constraint to well-foundedness, so that we get a foundation system which is categorical and in which there is a universal category.

The well-founded case has been addressed in a web page, at:

http://www.rbjones.com/rbjpub/pp/gst/pcf-defns.html

using a pair of rather cumbersome constructions. The material here offers an alternative presentation of essentially the same material as well as different constructions not confined to well-founded ontologies.

Here I propose to take from that only the manner of representing categories and functors, and to recode the construction following Forster's definition of pseudo-well-foundedness. Thus, a functor will be a triple consisting of a set which is the domain, a set which is the codomain and a function which is a many one relation between the domain and the codomain, total on the domain. A category is a set of functors. The left and right identity operations yeild the domain and codomain of the functors, composition is relational composition on the graphs.

In the well-founded case the construction is a liberalisation of the notion of 'hereditarily P' set. which is closely coupled to the notion of well-foundedness. The two liberalisations are, firstly that the notion of constituent replaces that of member and secondly that we have two sorts of entity involved. In the non well-founded case we begin with the idea of pseudo-hereditary set which is coupled with Forster's notion of pseudo-wellfoundedness. This already is two sorted, so the hope is that these two sorts can be the categories and functors. So we perform the same liberalisation as before (from talk of members to talk of constituents) and then we have a notion of two sets being 'pseudo-C-hereditarily (P,Q)' (the 'C' being the notion of constituent at stake, of which strictly there are two). Then we plumb in the properies specific to the category theoretic application.

All of this would be to no avail if done in the context of a well-founded set theory, though it would be useful to know whether it yields the same result as 'C-heredicarily (P,Q)' in such a context. To get a result which is not well-founded, we need to start with a non-well-founded collection of sets. For this purpose we have axiomatisations in ProofPower HOL of NF and NFU. The work is done by defining operators on set membersbip relations, so that the same construction can be applied to more than one set theory.

2.2 Approaches Based on Poly-Sets

The most distinctive feature of this material, in contrast with the material based on NFU is not the particular set theory conceived of as its content, but rather that the material is mainly presented as a construction over a more or less arbitrary set theory and only at the end instantiated to a particular set theory. In fact, the theory of Poly-Sets is given in the same manner. This makes it possible to demonstrate more fully the relationship between properties in the underlying set theories and properties in the resulting category theoretic system.

One this approach has been completed it should be possible to use the material in it to restate the definition based on NFU, until that is done this document will appear to have two superficial different treatments of the same category theoretic material.

3 PRE-FUNCTORS

This is a theory of the kind of functions which are suitable for use as the arrows in a concrete category. Since the concrete categories which are of interest here are all categories of categories, the arrows will be functors and it is therefore not inappropriate to call this special kind of function a "pre-functor".

The sole difference between a prefunctor and a many-one relation in set theory is simply that the pre-functors have a defined codomain which may be distinct from its range. It does not have a domain which is distinct from the set of values for which it is defined, since we expect the functions in a concrete category and the functors in a category of categories to be total functions.

This theory is a child of the formalisation of NFU in [4] (roughly) following Holmes [2].

SML

```
open_theory "nfu_f";
force_new_theory "pre_func";
set_merge_pcs ["hol1", "'savedthm_cs_∃_proof", "'nfu_f1"];
```

HOL Constant

$\boldsymbol{PreFunctor} : SET_{nf} \to BOOL$

$\forall p \bullet\ PreFunctor\ p \Leftrightarrow \exists x\ y \bullet\ p = op(x,\ y) \land Rel\ x \land ManyOne_{nf}\ x \land rng\ x \subseteq_{nf}\ y$

We define functions giving the graph, domain and codomain and field of a pre-functor and for application and composition of prefunctors.

HOL Constant

$\boldsymbol{PFgraph} : SET_{nf} \to SET_{nf}$

$PFgraph = fst$

HOL Constant

$\boldsymbol{PFdom} : SET_{nf} \to SET_{nf}$

$\forall p \bullet\ PFdom\ p = dom\ (PFgraph\ p)$

HOL Constant

$\boldsymbol{PFcod} : SET_{nf} \to SET_{nf}$

$PFcod = snd$

HOL Constant

$\boldsymbol{PFfield} : SET_{nf} \to SET_{nf}$

$\forall p \bullet\ PFfield\ p = PFdom\ p \cup_{nf}\ PFcod\ p$

SML

```
declare_infix(300, "⨾p");
```

This composition operator does not check that the pre-functors being compose have matching ‘types’.

HOL Constant

$\$\mathbf{{}_{9}^{o}p} : SET_{nf} \to SET_{nf} \to SET_{nf}$

$\forall p\ q\bullet\ p\ {}_{9}^{o}{}_{p}\ q = op((PFgraph\ p)\ {}_{9}^{o}{}_{nf}\ (PFgraph\ q),\ PFcod\ q)$

The identity pre-functor over some set is:

HOL Constant

$\boldsymbol{PFid} : SET_{nf} \to SET_{nf}$

$\forall a\bullet\ PFid\ a = op(id\ a,\ a)$

We can now define the left and right identities for a pre-functor.

HOL Constant

$\boldsymbol{PFleft} : SET_{nf} \to SET_{nf}$

$\forall a\bullet\ PFleft\ a = PFid\ (PFdom\ a)$

HOL Constant

$\boldsymbol{PFright} : SET_{nf} \to SET_{nf}$

$\forall a\bullet\ PFright\ a = PFid\ (PFcod\ a)$

Pre-functor composition is an infix suffix *p*.

SML

```
declare_infix (320, "p");
```

HOL Constant

$\$\boldsymbol{{}_{p}} : SET_{nf} \to SET_{nf} \to SET_{nf}$

$\forall p\ c\ d\bullet\ op(c,\ d) \in_{nf} (PFgraph\ p) \Rightarrow p\ {}_{nf}\ c = d$

4 CATEGORIES AND FUNCTORS

More than one approach is considered for defining collections of categories and functors in terms of the selected axiomatisation of set theory.

The first approach considered was inspired by the notion of pseudo-well-foundedness discussed in [1], the second on co-inductive methods described in [3]. Some initial material describing the categories and functors under discussion is used by both approached and is therefore presented first.

4.1 Common Material on Categories and Functors

Now we define the properties and content functions corresponding to concrete categories and functors. This could be done by defining a function which takes a membership relation as an argument and returns a full set of four values for use with the function *PseudoCDPQHeredirary*. However, this involves additional effort which would only be repaid if the construction were used over multiple set theories so I shall begin with a more direct definition.

The intention is that when the two collections have been defined they will be used to create two new types, and that theorems would then be proven about those types which would be suitable for an independent axiomatisation of the theories (for use as a foundation system). This all involves quite a bit of reasoning which strictly belongs to the metatheory and would best not stored in the same theories as contain then object theory. I will therefore create a theory *meta_cf* which will contain the 'metatheory' and put the theory itself in theory *cf* (for Categories and Functors). I propose to base the construction on NFU.

SML
```
force_new_theory "catfun";
(*new_parent "fixp";*)
```

A concrete category (here) is a set of functors (the arrows) which is closed under composition (where the domain and codomains match) and includes the identity functors on the domain and codomain of each functor. Composition is the same operation in all categories, and is associative as here defined.

A concrete functor is an ordered pair of which:

- the first element is a many-one relation
- the second element is a set which includes the right field of the relation
- the mapping defined by the first element over its domain respects composition on that domain

Note that the codomain is explicit in the right hand element of the functor, the domain is recovered from the graph on the left, hence all functors are total.

HOL Constant

$\boldsymbol{CatProp} : SET_{nf} \to BOOL$

$\forall a \bullet\ CatProp\ a \Leftrightarrow$
$\quad (\forall b \bullet\ b \in_{nf} a \Rightarrow PFleft\ b \in_{nf} a \land PFright\ b \in_{nf} a)$
$\land \quad (\forall b\ c{:}SET_{nf} \bullet\ b \in_{nf} a \land c \in_{nf} a \land PFdom\ c = PFcod\ b \Rightarrow (b \fatsemi_p c) \in_{nf} a)$

This function returns the set of functors in a category.

HOL Constant

$\boldsymbol{CatCon} : SET_{nf} \to SET_{nf}\ SET$

$\forall a \bullet\ CatCon\ a = \{f \mid f \in_{nf} a\}$

HOL Constant

$\boldsymbol{FuncProp} : SET_{nf} \to BOOL$

$\forall a \bullet\ FuncProp\ a \Leftrightarrow$
$\quad rng\ (PFgraph\ a) \subseteq_{nf} (PFcod\ a)$
$\land \quad (\forall b\ c \bullet\ b \in_{nf} PFdom\ a \land c \in_{nf} PFdom\ a \land PFcod\ b = PFdom\ c$
$\qquad \Rightarrow op\ (b \mathbin{⨾_p} c,\ (a\ _p\ b) \mathbin{⨾_{nf}} (a\ _p\ c)) \in_{nf} a)$

This function returns the set of categories in the field of a functor.

HOL Constant

$\boldsymbol{FuncCon} : SET_{nf} \to SET_{nf}\ SET$

$\forall f \bullet\ FuncCon\ f = \{PFdom\ f;\ PFcod\ f\}$

The following is a deeper category property, which asserts that the members of the category are all functors.

HOL Constant

$\boldsymbol{CatFuncProp} : SET_{nf} \to BOOL$

$\forall a \bullet\ CatFuncProp\ a \Leftrightarrow CatProp\ a \land \forall f \bullet\ f \in CatCon\ a \Rightarrow FuncProp\ f$

If the deeper property is in use, a deep content function may also be needed. This gives for each category the set of categories which appear as the domain or codomain of one of the functors in the category.

HOL Constant

$\boldsymbol{CatFuncCon} : SET_{nf} \to SET_{nf}\ SET$

$\forall c \bullet\ CatFuncCon\ c = \bigcup\{fc \mid \exists f \bullet\ f \in CatCon\ c \land fc = FuncCon\ f\}$

Next we have the corresponding deeper functor property which asserts that the domain and codomain of the functor are categories.

HOL Constant

$\boldsymbol{FuncCatProp} : SET_{nf} \to BOOL$

$\forall a \bullet\ FuncCatProp\ a \Leftrightarrow FuncProp\ a \land \forall c \bullet\ c \in FuncCon\ a \Rightarrow CatProp\ c$

Similarly the deeper functor content:

HOL Constant

$\boldsymbol{FuncCatCon} : SET_{nf} \to SET_{nf}\ SET$

$\forall f \bullet\ FuncCatCon\ f = \bigcup\{cf \mid \exists c \bullet\ c \in FuncCon\ f \land cf = CatCon\ c\}$

For use in co-inductive definitions the property and the content function are combined into a single content function which effectively incorporates the property. This function maps sets to sets. It maps each set to the set of objects which can be constructed from the members in the argument set.

4.2 Meta-Theory by Pseudo-Induction

SML

```
force_new_theory "metapi";
new_parent "nfu_f";
(*new_parent "fixp";*)
set_merge_pcs["hol1", "'savedthm_cs_∃_proof"];
```

Since these sets (the categories and functors) are expected to be disjoint, at most (hopefully exactly) one of them will contain the empty set. In our scheme the empty set is a category (but not a functor, functors will all be ordered pairs). Looking at our definition of hereditarily above we see that the empty set can only be a member of the left hand collection, and so (c,P) must be the content function and characterising property for the categories and (d,Q) are those for functors.

We are now in a position to define the sets of pseudo-well-founded categories and functors.

HOL Constant

$\boldsymbol{PWFcf} : SET_{nf}\ SET \times SET_{nf}\ SET$

$PWFcf = PseudoHereditarilyPQ\ (CatCon,\ FuncCon,\ CatProp,\ FuncProp)$

We need them as properties for introducing types.

HOL Constant

$\boldsymbol{PWFcategory} : SET_{nf} \to BOOL$

$PWFcategory = \lambda c \bullet\ c \in (Fst\ PWFcf)$

HOL Constant

$\boldsymbol{PWFfunctor} : SET_{nf} \to BOOL$

$PWFfunctor = \lambda f \bullet\ f \in (Snd\ PWFcf)$

In order to use these properties to introducing new types we have to prove that they are non-empty. Its useful to keep the theorems for the specific witnesses, the empty categort and the trivial functor.

$pwf\_cat\_\varnothing\_thm = \quad \vdash PWFcategory\ \varnothing$

$\exists\_pwf\_cat\_thm = \quad \vdash \exists\ a \bullet\ PWFcategory\ a$

$\varnothing\_pwf\_func\_thm = \quad \vdash PWFfunctor\ (op(\varnothing,\ \varnothing))$

$\exists\_pwf\_func\_thm = \quad \vdash \exists\ a \bullet\ PWFfunctor\ a$

4.3 Object Theory by Pseudo-Induction

Now we introduce a new theory in which the types of pseudo-well-founded categories and functors are defined, and whose theories are intended to be a suitable independent axiomatisation for a two-sorted foundation system for which these two types supply a model.

SML

```
open_theory "metapi";
force_new_theory "cfpi";
force_new_pc "cfpi";
new_type_defn(["CAT"], "CAT", [], ∃_pwf_cat_thm);
new_type_defn(["FUNC"], "FUNC", [], ∃_pwf_func_thm);
```

4.4 Meta-Theory by (Co-)Induction

The one we want here is co-induction, but we also show elements of the treatment by induction.

SML

```
open_theory "catfun";
force_new_theory "metaci";
new_parent "nfu_f";
(*new_parent "fixp";*)
set_merge_pcs["hol1", "'savedthm_cs_∃_proof"];
```

We must first chose one of the kinds of representation which are supported by the theory 'fixp' and convert our available functions to the chosen format. In this case a 'content relation' will suffice. This is the relation between an object and those things of which it is an immediate constituent. In this case we take two of these, one giving the relation between a category and those categories of which it is the domain or codomain of a functor, the other that between a functor and those functors of whose domain or codomain the functor is a member.

HOL Constant

$\boldsymbol{CatRel} : SET_{nf} \to SET_{nf} \to BOOL$

$\forall c\ d \bullet\ CatRel\ c\ d \Leftrightarrow CatFuncProp\ d \land c \in CatFuncCon\ d$

HOL Constant

$\boldsymbol{FuncRel} : SET_{nf} \to SET_{nf} \to BOOL$

$\forall f\ g \bullet\ FuncRel\ f\ g \Leftrightarrow FuncCatProp\ g \land f \in FuncCatCon\ g$

We may now use these two functions to obtain the least and greatest fixed points which are the respective inductive and co-inductively defined classes.

First the inductive case:

HOL Constant

$\boldsymbol{WFcf} : SET_{nf}\ SET \times SET_{nf}\ SET$

$WFcf = (HeredRel\ CatRel,\ HeredRel\ FuncRel)$

Then the co-inductive case:

HOL Constant

$\boldsymbol{CWFcf} : SET_{nf}\ SET \times SET_{nf}\ SET$

$CWFcf = (CoHeredRel\ CatRel,\ CoHeredRel\ FuncRel)$

For the purpose of introducing new types we need this as two properties:

HOL Constant

$\boldsymbol{CWFcategory} : SET_{nf} \rightarrow BOOL$

$\forall c \bullet\ CWFcategory\ c = c \in Fst\ CWFcf$

HOL Constant

$\boldsymbol{CWFfunctor} : SET_{nf} \rightarrow BOOL$

$\forall f \bullet\ CWFfunctor\ f = f \in Snd\ CWFcf$

5 The Theory pre_func

5.1 Parents

nfu_f

5.2 Children

catfun

5.3 Constants

PreFunctor	$SET_{nf} \rightarrow BOOL$
PFgraph	$SET_{nf} \rightarrow SET_{nf}$
PFdom	$SET_{nf} \rightarrow SET_{nf}$
PFcod	$SET_{nf} \rightarrow SET_{nf}$
PFfield	$SET_{nf} \rightarrow SET_{nf}$
$\$\mathbin{⨾}_{\boldsymbol{p}}$	$SET_{nf} \rightarrow SET_{nf} \rightarrow SET_{nf}$
PFid	$SET_{nf} \rightarrow SET_{nf}$
PFleft	$SET_{nf} \rightarrow SET_{nf}$
PFright	$SET_{nf} \rightarrow SET_{nf}$
$\$_{\boldsymbol{p}}$	$SET_{nf} \rightarrow SET_{nf} \rightarrow SET_{nf}$

5.4 Fixity

Right Infix 300:

$⨾_{\boldsymbol{p}}$

Right Infix 320:

$_{\boldsymbol{p}}$

5.5 Definitions

PreFunctor	$\vdash \forall\ p$ $\bullet\ PreFunctor\ p$ $\Leftrightarrow (\exists\ x\ y$ $\bullet\ p = op\ (x,\ y)$ $\wedge\ Rel\ x$ $\wedge\ ManyOne_{nf}\ x$ $\wedge\ rng\ x \subseteq_{nf}\ y)$
PFgraph	$\vdash PFgraph = fst$
PFdom	$\vdash \forall\ p\bullet\ PFdom\ p = dom\ (PFgraph\ p)$
PFcod	$\vdash PFcod = snd$
PFfield	$\vdash \forall\ p\bullet\ PFfield\ p = PFdom\ p \cup_{nf} PFcod\ p$
$⨾_{\boldsymbol{p}}$	$\vdash \forall\ p\ q$ $\bullet\ p ⨾_p q = op\ (PFgraph\ p ⨾_{nf} PFgraph\ q,\ PFcod\ q)$
PFid	$\vdash \forall\ a\bullet\ PFid\ a = op\ (id\ a,\ a)$
PFleft	$\vdash \forall\ a\bullet\ PFleft\ a = PFid\ (PFdom\ a)$
PFright	$\vdash \forall\ a\bullet\ PFright\ a = PFid\ (PFcod\ a)$
$_{\boldsymbol{p}}$	$\vdash ConstSpec$ $(\lambda\ _p{}'$ $\bullet\ \forall\ p\ c\ d$ $\bullet\ op\ (c,\ d) \in_{nf} PFgraph\ p \Rightarrow p\ _{nf}\ c = d)$ $\$_p$

6 The Theory catfun

6.1 Parents

pre_func

6.2 Children

metaci *metapi*

6.3 Constants

CatProp	$SET_{nf} \rightarrow BOOL$
CatCon	$SET_{nf} \rightarrow SET_{nf}\ \mathbb{P}$
FuncProp	$SET_{nf} \rightarrow BOOL$
FuncCon	$SET_{nf} \rightarrow SET_{nf}\ \mathbb{P}$
CatFuncProp	$SET_{nf} \rightarrow BOOL$
CatFuncCon	$SET_{nf} \rightarrow SET_{nf}\ \mathbb{P}$
FuncCatProp	$SET_{nf} \rightarrow BOOL$
FuncCatCon	$SET_{nf} \rightarrow SET_{nf}\ \mathbb{P}$

6.4 Definitions

CatProp

$$\begin{array}{l} \vdash \forall\ a \\ \quad \bullet\ CatProp\ a \\ \qquad \Leftrightarrow (\forall\ b \\ \qquad\quad \bullet\ b \in_{nf} a \\ \qquad\qquad \Rightarrow PFleft\ b \in_{nf} a \wedge PFright\ b \in_{nf} a) \\ \qquad\quad \wedge\ (\forall\ b\ c \\ \qquad\quad \bullet\ b \in_{nf} a \wedge c \in_{nf} a \wedge PFdom\ c = PFcod\ b \\ \qquad\qquad \Rightarrow b \fatsemi_p c \in_{nf} a) \end{array}$$

CatCon

$$\vdash \forall\ a \bullet\ CatCon\ a = \{f | f \in_{nf} a\}$$

FuncProp

$$\begin{array}{l} \vdash \forall\ a \\ \quad \bullet\ FuncProp\ a \\ \qquad \Leftrightarrow rng\ (PFgraph\ a) \subseteq_{nf} PFcod\ a \\ \qquad\quad \wedge\ (\forall\ b\ c \\ \qquad\quad \bullet\ b \in_{nf} PFdom\ a \\ \qquad\qquad\quad \wedge\ c \in_{nf} PFdom\ a \\ \qquad\qquad\quad \wedge\ PFcod\ b = PFdom\ c \\ \qquad\qquad \Rightarrow op\ (b \fatsemi_p c,\ a\ {}_p\ b \fatsemi_{nf} a\ {}_p\ c) \in_{nf} a) \end{array}$$

FuncCon

$$\vdash \forall\ f \bullet\ FuncCon\ f = \{PFdom\ f;\ PFcod\ f\}$$

CatFuncProp

$$\begin{array}{l} \vdash \forall\ a \\ \quad \bullet\ CatFuncProp\ a \\ \qquad \Leftrightarrow CatProp\ a \wedge (\forall\ f \bullet\ f \in CatCon\ a \Rightarrow FuncProp\ f) \end{array}$$

CatFuncCon

$$\begin{array}{l} \vdash \forall\ c \\ \quad \bullet\ CatFuncCon\ c \\ \qquad = \bigcup\ \{fc | \exists\ f \bullet\ f \in CatCon\ c \wedge fc = FuncCon\ f\} \end{array}$$

FuncCatProp

$$\begin{array}{l} \vdash \forall\ a \\ \quad \bullet\ FuncCatProp\ a \\ \qquad \Leftrightarrow FuncProp\ a \wedge (\forall\ c \bullet\ c \in FuncCon\ a \Rightarrow CatProp\ c) \end{array}$$

$\boldsymbol{FuncCatCon} \quad \vdash \forall\ f$
$\quad \bullet\ FuncCatCon\ f$
$\quad\quad = \bigcup\ \{cf \mid \exists\ c \bullet\ c \in FuncCon\ f\ \wedge\ cf\ =\ CatCon\ c\}$

7 The Theory metapi

7.1 Parents

nfu_f catfun

7.2 Children

cfpi

7.3 Constants

PWFcf	$SET_{nf}\ \mathbb{P} \times SET_{nf}\ \mathbb{P}$
PWFcategory	$SET_{nf} \rightarrow BOOL$
PWFfunctor	$SET_{nf} \rightarrow BOOL$

7.4 Definitions

PWFcf	$\vdash PWFcf = PseudoHereditarilyPQ\ (CatCon,\ FuncCon,\ CatProp,\ FuncProp)$
PWFcategory	$\vdash PWFcategory = (\lambda\ c \bullet\ c \in Fst\ PWFcf)$
PWFfunctor	$\vdash PWFfunctor = (\lambda\ f \bullet\ f \in Snd\ PWFcf)$

7.5 Theorems

pwf_cat_∅_thm

$\vdash PWFcategory\ \varnothing$

∃_***pwf_cat_thm***

$\vdash \exists\ a \bullet\ PWFcategory\ a$

∅_***pwf_func_thm***

$\vdash PWFfunctor\ (op\ (\varnothing,\ \varnothing))$

∃_***pwf_func_thm***

$\vdash \exists\ a \bullet\ PWFfunctor\ a$

8 The Theory cfpi

8.1 Parents

metapi

8.2 Types

CAT
FUNC

8.3 Definitions

CAT	$\vdash \exists\ f \bullet\ \mathit{TypeDefn}\ \mathit{PWFcategory}\ f$
FUNC	$\vdash \exists\ f \bullet\ \mathit{TypeDefn}\ \mathit{PWFfunctor}\ f$

9 The Theory metaci

9.1 Parents

nfu_f catfun

9.2 Constants

CatRel	$SET_{nf} \rightarrow SET_{nf} \rightarrow BOOL$
FuncRel	$SET_{nf} \rightarrow SET_{nf} \rightarrow BOOL$
WFcf	$SET_{nf}\ \mathbb{P} \times SET_{nf}\ \mathbb{P}$
CWFcf	$SET_{nf}\ \mathbb{P} \times SET_{nf}\ \mathbb{P}$
CWFcategory	$SET_{nf} \rightarrow BOOL$
CWFfunctor	$SET_{nf} \rightarrow BOOL$

9.3 Definitions

CatRel	$\vdash \forall\ c\ d \bullet\ CatRel\ c\ d \Leftrightarrow CatFuncProp\ d \wedge c \in CatFuncCon\ d$
FuncRel	$\vdash \forall\ f\ g \bullet\ FuncRel\ f\ g \Leftrightarrow FuncCatProp\ g \wedge f \in FuncCatCon\ g$
WFcf	$\vdash\ WFcf = (HeredRel\ CatRel,\ HeredRel\ FuncRel)$
CWFcf	$\vdash\ CWFcf = (CoHeredRel\ CatRel,\ CoHeredRel\ FuncRel)$
CWFcategory	$\vdash \forall\ c \bullet\ CWFcategory\ c \Leftrightarrow c \in Fst\ CWFcf$
CWFfunctor	$\vdash \forall\ f \bullet\ CWFfunctor\ f \Leftrightarrow f \in Snd\ CWFcf$

9.4 Theorems

∃_***CWFcategory_thm***

$\vdash \exists\ c \bullet\ CWFcategory\ c$

10 INDEX