Category Theory

Roger Jones

Abstract

Formalisation of some of the concepts of category theory in ProofPower-HOL.

Created 2006/04/09

Last Changed Date: 2011/08/15 17:00:38

http://www.rbjones.com/rbjpub/pp/doc/t017.pdf

Id: t017.doc,v 1.7 2011/08/15 17:00:38 rbj Exp

© Roger Bishop Jones; Licenced under Gnu LGPL

Contents

References

[1] Roger Bishop Jones. Category Theory. *RBJones.com*, 2010.
`http://www.rbjones.com/rbjpub/pp/doc/t017.pdf`.

[2] Roger Bishop Jones. Universal Algebra and Lattice Theory. *RBJones.com*, 2010.
`http://www.rbjones.com/rbjpub/pp/doc/t039.pdf`.

1 Introduction

This is minor exercise to help some of the elementary concepts of category theory into my brain, and is not intended to show offer any enlightenment which cannot be found in any elementary text on category theory. In fact I am following Saunders Mac Lane (roughly).

Create new theory "cat", parent "rbjmisc".

SML

```
open_theory "rbjmisc";
open PreConsisProof; open UnifyForwardChain; open Trawling;
force_new_theory "cat";
force_new_pc "'cat";
merge_pcs ["'savedthm_cs_∃_proof"] "'cat";
new_parent "sum";
new_parent "one";
set_merge_pcs["rbjmisc", "'cat"];
set_flag ("pp_use_alias", true);
```

2 Categories Functors and Natural Transformations

2.1 Definition of Category

We model a category by its set of arrows. It is therefore a set (of arrows), together with:

- a partial associative operation over the arrows
- left and right operators which return the domain and codomain of each arrow

HOL Labelled Product

CAT

```
Arrows :'a SET;
Compose  :'a → 'a → 'a;
Left Right: 'a → 'a
```

Before giving the definition of a category it will be convenient to define some auxiliary concepts.

HOL Constant

$$\textbf{\textit{Obj}} : \ 'a\ CAT \to 'a\ SET$$

$$\forall\ C\ a\bullet\ a \in Obj\ C \Leftrightarrow a \in Arrows\ C \wedge a = Left\ C\ a$$

We need often to state explicitly that two arrows are composable and therefore introduce a notation for this. We use an infix operator which yields a property of categories.

SML

```
declare_infix(400, "ϒ");
```

HOL Constant

$$\$\Upsilon : \ 'a \to 'a \to 'a\ CAT \to BOOL$$

$$\begin{aligned}&\forall\ f\ g\ C\bullet\ (f\ \ \Upsilon\ g)\ C \Leftrightarrow\\ &\quad f \in Arrows\ C \wedge g \in Arrows\ C\\ &\quad \wedge\ Right\ C\ f = Left\ C\ g\end{aligned}$$

HOL Constant

$$\textbf{\textit{Identity}} : \ 'a\ CAT \to 'a\ SET$$

$$\begin{aligned}&\forall\ C\ a\bullet\ a \in Identity\ C \Leftrightarrow a \in Arrows\ C\\ &\quad \wedge\ Left\ C\ a = a \wedge Right\ \ C\ a = a\\ &\quad \wedge\ \forall b\bullet\ b \in Arrows\ C \Rightarrow (Left\ C\ b = a \Rightarrow Compose\ C\ a\ b = b)\\ &\qquad \wedge\ (Right\ C\ b = a \Rightarrow Compose\ C\ b\ a = b)\end{aligned}$$

It will also be convenient to cite concisely the domain and codomain of an arrow, i.e. a notation for homsets.

SML

```
declare_infix(400, ">->");
```

HOL Constant

$$\$>-> : \ 'a \to 'a \to 'a\ CAT \to 'a\ SET$$

$$\begin{aligned}&\forall\ f\ a\ b\ C\bullet\ f \in (a >-> b)\ C \Leftrightarrow\\ &\quad f \in Arrows\ C \wedge a \in Arrows\ C \wedge b \in Arrows\ C\\ &\quad \wedge\ Left\ C\ f = a \wedge Right\ C\ f = b\end{aligned}$$

SML

```
declare_alias ("↦", ⌜$>->⌝);
```

An infix notation for composition:

SML

```
declare_infix(500, "o_c");
```

HOL Constant

$$\$o_c : 'a \to 'a \to 'a\ CAT \to 'a$$

$$\forall\ f\ g\ C\bullet\ (f\ o_c\ g)\ C = Compose\ C\ f\ g$$

HOL Constant

$$Cat : 'a\ CAT \to BOOL$$

$$\begin{array}{l}
\forall\ {}_c\bullet\ Cat\ {}_c \Leftrightarrow \\
\quad (\forall\ x\bullet\ x \in Arrows\ {}_c \\
\qquad \Rightarrow Left\ {}_c\ x \in Identity\ {}_c \wedge Right\ {}_c\ x \in Identity\ {}_c) \\
\wedge\ (\forall\ x\ y\bullet\ (x\ \curlyvee\ y)_c \Rightarrow (x\ o_c\ y)_c \in Arrows\ {}_c \\
\qquad \wedge\ ((x\ o_c\ y)_c) \in ((Left\ {}_c\ x) \rightarrowtail (Right\ {}_c\ y))_c) \\
\wedge\ (\forall\ x\ y\ z\bullet\ (x\ \curlyvee\ y)_c \wedge (y\ \curlyvee\ z)_c \\
\qquad \Rightarrow (x\ o_c\ (y\ o_c\ z)_c)_c = (((x\ o_c\ y)_c)\ o_c\ z)_c)
\end{array}$$

2.1.1 Notation for Commutative Diagrams

I have no idea whether this is any use.

SML

```
declare_infix (310, "Ξ");
declare_infix (310, "Λ");
```

HOL Constant

$$\$\Lambda : 'a\ CAT \to 'a\ LIST \to 'a$$

$$\begin{array}{l}
\forall\ C\ la\ a\bullet\ C\ \Lambda\ [a] = a \\
\qquad \wedge\ (la = [] \vee C\ \Lambda\ (Cons\ a\ la) = Compose\ C\ a\ (C\ \Lambda\ la))
\end{array}$$

HOL Constant

$$\$\Xi : 'a\ CAT \to 'a\ LIST\ LIST \to BOOL$$

$$\begin{array}{l}
\forall\ C\ lla\bullet\ C\Xi\ lla \Leftrightarrow \\
\qquad \exists x\bullet\ Fold\ Insert\ (Map\ (\$\Lambda\ C)\ lla)\ \{\} = \{x\}
\end{array}$$

2.2 Categories

HOL Constant

$$\varnothing_c : 'a\ CAT$$

$$Arrows\ \varnothing_c = \{\}$$

HOL Constant

$\mathbf{1_c} : \ 'a \to 'a \ CAT$

$\forall x{:}'a\bullet \ Cat \ (1_c \ x) \ \wedge \ Arrows \ (1_c \ x) = \{x\}$

Of course, we have:

$\boldsymbol{Cat\_\varnothing_c\_thm} \vdash \ Cat \ \varnothing_c$

Both of the above are special cases of *discrete* categories. In a discrete category all arrows are identity arrows. The following function allows the construction of arbitrary discrete categories.

HOL Constant

$\boldsymbol{Discrete_c} : \ 'a \ SET \to 'a \ CAT$

$\forall s{:}'a \ SET\bullet \ Discrete_c \ s = MkCAT \ s \ (\lambda x \ y\bullet \ x) \ (\lambda x\bullet x) \ (\lambda x\bullet x)$

Any set can be made into a discrete category in this way:

$discrete\_cat\_thm \vdash \forall s\bullet \ Cat \ (Discrete_c \ s)$

2.3 Functors

HOL Constant

$\boldsymbol{Functor} : \ 'a \ CAT \times 'b \ CAT \to ('a \to 'b) \ SET$

$\forall \ A \ B \ f\bullet \ f \in Functor \ (A,B) \Leftrightarrow$
$\quad (\forall a\bullet \ a \in Arrows \ A \Rightarrow f \ a \in Arrows \ B)$
$\wedge \quad \forall a \ b\bullet \ (a \ \curlyvee \ b) \ A$
$\quad \Rightarrow Right \ B \ (f \ a) = f \ (Right \ A \ a)$
$\quad \wedge \ Left \ B \ (f \ b) = f \ (Left \ A \ b)$
$\quad \wedge \ Compose \ B \ (f \ a) \ (f \ b) = f \ (Compose \ A \ a \ b)$

HOL Constant

$\boldsymbol{Full} : \ 'a \ CAT \times 'b \ CAT \to ('a \to 'b) \ SET$

$\forall \ A \ B \ f\bullet \ f \in Full \ (A,B) \Leftrightarrow$
$\quad \forall b\bullet \ b \in Arrows \ B \Rightarrow \exists a\bullet \ a \in Arrows \ A \wedge b = f \ a$

HOL Constant

$\boldsymbol{Faithful} : \ 'a \ CAT \times 'b \ CAT \to ('a \to 'b) \ SET$

$\forall \ A \ B \ f\bullet \ f \in Faithful \ (A,B) \Leftrightarrow$
$\quad \forall a \ b\bullet \ a \in Arrows \ A \wedge b \in Arrows \ A \wedge f \ a = f \ b \Rightarrow a = b$

Since functors are functions, they compose, and the composition is also a functor. Composition also preserves fullness and faithfulness.

$$\vdash \forall A{:}'a\ CAT;\ B{:}'b\ CAT;\ C{:}'c\ CAT;\ ab{:}\ 'a \to 'b;\ bc{:}\ 'b \to 'c \bullet$$
$$ab \in Functor\ (A,B) \wedge bc \in Functor\ (B,C) \Rightarrow (bc\ o\ ab) \in Functor\ (A,C)$$

$$\vdash \forall A{:}'a\ CAT;\ B{:}'b\ CAT;\ C{:}'c\ CAT;\ ab{:}\ 'a \to 'b;\ bc{:}\ 'b \to 'c \bullet$$
$$ab \in Full\ (A,B) \wedge bc \in Full\ (B,C) \Rightarrow (bc\ o\ ab) \in Full\ (A,C)$$

$$\vdash \forall A{:}'a\ CAT;\ B{:}'b\ CAT;\ C{:}'c\ CAT;\ ab{:}\ 'a \to 'b;\ bc{:}\ 'b \to 'c \bullet$$
$$ab \in Functor\ (A,B) \wedge ab \in Faithful\ (A,B) \wedge bc \in Faithful\ (B,C)$$
$$\Rightarrow (bc\ o\ ab) \in Faithful\ (A,C)$$

2.3.1 Subcategories

SML

$$declare\_infix\ (310,\ "\subseteq_c");$$

HOL Constant

$$\$\subseteq_{\mathbf{c}} : 'a\ CAT \to 'a\ CAT \to BOOL$$

$$\forall\ A1\ A2 \bullet\ A1 \subseteq_c A2 \Leftrightarrow$$
$$(\forall a \bullet\ a \in Arrows\ A1 \Rightarrow a \in Arrows\ A2$$
$$\wedge\ Left\ A1\ a \in Arrows\ A1 \wedge Right\ A1\ a \in Arrows\ A1$$
$$\wedge\ Left\ A1\ a = Left\ A2\ a \wedge Right\ A1\ a = Right\ A2\ a)$$
$$\wedge\ (\forall a\ b \bullet\ a \in Arrows\ A1 \wedge b \in Arrows\ A1 \Rightarrow$$
$$Compose\ A1\ a\ b \in Arrows\ A1$$
$$\wedge\ Compose\ A1\ a\ b = Compose\ A2\ a\ b)$$

A subcategory of a category is of course a category.

$$subcat\_cat\_thm \vdash \forall a\ b{:}'a\ CAT \bullet\ Cat\ a \wedge b \subseteq_c a \Rightarrow Cat\ b$$

2.4 Natural Transformations

HOL Constant

$$\$\mathbf{NatTran} :\ ('a\ CAT \times 'b\ CAT) \to (('a \to 'b) \times ('a \to 'b)) \to ('a \to 'b)\ SET$$

$$\forall\ A\ B\ R\ S\ \tau \bullet\ \tau \in NatTran\ (A,B)\ (R,S) \Leftrightarrow$$
$$(\forall a \bullet\ a \in Obj\ A$$
$$\Rightarrow \tau\ a \in Arrows\ B \wedge Left\ B(\tau\ a) = R\ a \wedge Right\ B(\tau\ a) = S\ a)$$
$$\wedge\ (\forall f \bullet\ f \in Arrows\ A$$
$$\Rightarrow Left\ B(\tau\ (Right\ A\ f)) = R\ f \wedge Right\ B(\tau\ (Right\ A\ f)) = S\ f$$
$$\wedge\ Compose\ B\ (\tau\ (Left\ A\ f))\ (S\ f) = Compose\ B\ (R\ f)\ (\tau\ (Right\ A\ f)))$$

HOL Constant

$\$\boldsymbol{Id_{NT}} : ('a\ CAT \times 'b\ CAT) \to ('a \to 'b) \to ('a \to 'b)$

$\forall\ A\ B\ R\ f \bullet\ Id_{NT}\ (A,B)\ R\ f = Left\ B\ (R\ f)$

3 Constructions on Categories

3.1 Opposites

SML

```
declare_postfix (400, "op");
```

HOL Constant

$\$\boldsymbol{op} : 'a\ CAT \to 'a\ CAT$

$\forall\ C \bullet\ C^{op} = MkCAT\ (Arrows\ C)\ (\lambda x\ y \bullet Compose\ C\ y\ x)\ (Right\ C)\ (Left\ C)$

3.2 Functor Categories

4 The Yoneda Embedding

I conjecture that the Yoneda embedding can be defined without resort to "The" category of sets, and without constraints on size.

5 Categorial Structures

This doesn't belong here, and if it gets anywhere it will be moved elsewhere. Its an experiment in doing a kind of categorical foundation system by methods analogous to the study of foundations via the study of membership structures.

In the domain of discourse there exist only arrows. Some of these are identity arrows and such arrows are themselves categories.

The structure which constitutes a category is taken as defined in *Category Theory* [1]. The categorial structure is given by a type of arrows one element of which is distinguished as the universal category together with a function which assignes to each identity arrow in that category some other category.

SML

```
open_theory "cat";
force_new_theory "fcat";
new_parent "membership";
```

HOL Labelled Product

FCAT

$Ucat : 'a\ CAT;$

$Catmap : 'a \to 'a\ CAT$

The idea is then to define the properties of FCAT which are desirable for if they are to serve as the model of a foundation system, as an ontology for mathematics.

As in set theory we have some axioms which tell us what kind of things sets are, and then axioms which tell us which objects of that kind actually exist.

HOL Constant

$$\boldsymbol{Fcat} : {'a}\ FCAT \to BOOL$$

$$\forall fc{:}\,{'a}\ FCAT\bullet\ Cat\ (Ucat\ fc)$$
$$\qquad \wedge\ \forall obj{:}{'a}\bullet\ obj \in Obj\ (Ucat\ fc) \Rightarrow Cat\ (Catmap\ fc\ obj)$$

6 Categories in Universal Algebra

In this section I hope to explore the development of category theory as an application of the Universal Algebra under development in [2].

The motivations for this are not very strong. The greatest of these is my desire now to find a way to explore some possible philosophical applications of category theory. Why this should favour this particular approach is not obvious. However, I am working on the Universal Algebra and wanting to probe category theory, so doing the latter in the former makes some sense.

Whether a category is an algebra is moot. It depends on what you mean by algebra. Universal algebra normally deals with systems in which there are only total first order operations over a single domain, extended by some theoretical computer scientists (most prominently Goguen) to many sorted algebras. Typically there is also a presumption that algebra is done in or with a first order equational logic (at least for some purposes, e.g. for the presentation of particular algebras).

In Category theory one of the required operations, composition, is partial, and no particular logical institition is presumed. For these reasons Category theory may not be thought of as an algebra, but some of the results of our Universal Algebra may nevertheless be applicable, and crucially, the general manner of representing structures used in our Universal Algebra looks suitable for use in Category theory (a category being such a structure).

I propose to begin by replaying some of the former material in a new theory using the Universalm Algebra.

SML

```
open_theory "unalg";
force_new_theory "uacat";
force_new_pc "'uacat";
merge_pcs ["'prove_∃_⇒_conv", "'savedthm_cs_∃_proof"] "'uacat";
set_merge_pcs ["unalg", "'uacat"];
```

6.1 Definition of Category

We model a category by its set of arrows. It is therefore a set (of arrows), together with:

- a partial associative operation over the arrows
- left and right operators which return the domain and codomain of each arrow

We will represent categories using objects of type *STRUCT* having a suitable signature. The signature of a category is defined:

HOL Constant

```
| CatSig : SIG
|-----------------------------------------------------------
| CatSig = IxPack [("⦾9c" , 2); ("Left_c" , 1); ("Right_c" , 1)]
```

SML

```
|declare_infix (300, "⦾9c");
```

We need a convenient notation for categories and therefore define the following constructor:

HOL Constant

```
| MkCat : 'a SET → ('a → 'a → 'a) → ('a → 'a) → ('a → 'a) → 'a STRUCT
|-----------------------------------------------------------
| ∀D $⦾9c Left_c Right_c• MkCat D $⦾9c Left_c Right_c =
|     MkSTRUCT D (IxPack [("⦾9c" , pack2op $⦾9c); ("Left_c" , pack1op Left_c); ("Right_c" , pack1op Right_c)
```

To enable pattern matching in definitions we prove and install an appropriate existential theorem.

SML

```
|val MkCat_∃_lemma = make_alg_∃_thm ⌜MkCat⌝;
```

```
|MkCat_∃_lemma =
|     ⊢ ∀ cf• ∃ f• ∀ D $⦾9c Left_c Right_c•
|         f (MkCat D ⦾9c Left_c Right_c) = cf D ⦾9c Left_c Right_c
```

SML

```
|add_∃_cd_thms [MkCat_∃_lemma] "'uacat";
|set_merge_pcs ["unalg", "'uacat"];
```

Before giving the definition of a category it will be convenient to define some auxiliary concepts.

HOL Constant

```
| Obj : 'a STRUCT → 'a SET
|-----------------------------------------------------------
| ∀D $⦾9c Left_c Right_c a• let C = MkCat D $⦾9c Left_c Right_c in
|     a ∈ Obj C ⇔ a ∈ SCar C ∧ a = Left_c a
```

7 The Theory cat

7.1 Parents

one sum *rbjmisc*

7.2 Children

fcat

7.3 Constants

Right	$'a\ CAT \to 'a \to 'a$
Left	$'a\ CAT \to 'a \to 'a$
Compose	$'a\ CAT \to 'a \to 'a \to 'a$
Arrows	$'a\ CAT \to 'a\ \mathbb{P}$
MkCAT	$'a\ \mathbb{P} \to ('a \to 'a \to 'a) \to ('a \to 'a) \to ('a \to 'a) \to 'a\ CAT$
Obj	$'a\ CAT \to 'a\ \mathbb{P}$
$\$\Upsilon$	$'a \to 'a \to 'a\ CAT \to BOOL$
Identity	$'a\ CAT \to 'a\ \mathbb{P}$
$\$>->$	$'a \to 'a \to 'a\ CAT \to 'a\ \mathbb{P}$
$\$\boldsymbol{o_c}$	$'a \to 'a \to 'a\ CAT \to 'a$
Cat	$'a\ CAT \to BOOL$
$\$\Lambda$	$'a\ CAT \to 'a\ LIST \to 'a$
$\$\Xi$	$'a\ CAT \to 'a\ LIST\ LIST \to BOOL$
$\varnothing_c$	$'a\ CAT$
$\mathbf{1_c}$	$'a \to 'a\ CAT$
$\boldsymbol{Discrete_c}$	$'a\ \mathbb{P} \to 'a\ CAT$
Functor	$'a\ CAT \times 'b\ CAT \to ('a \to 'b)\ \mathbb{P}$
Full	$'a\ CAT \times 'b\ CAT \to ('a \to 'b)\ \mathbb{P}$
Faithful	$'a\ CAT \times 'b\ CAT \to ('a \to 'b)\ \mathbb{P}$
$\$\subseteq_c$	$'a\ CAT \to 'a\ CAT \to BOOL$
NatTran	$'a\ CAT \times 'b\ CAT \to ('a \to 'b) \times ('a \to 'b) \to ('a \to 'b)\ \mathbb{P}$
$\boldsymbol{Id_{NT}}$	$'a\ CAT \times 'b\ CAT \to ('a \to 'b) \to 'a \to 'b$
$\$^{op}$	$'a\ CAT \to 'a\ CAT$

7.4 Aliases

$\rightarrowtail$	$\$>-> : 'a \to 'a \to 'a\ CAT \to 'a\ \mathbb{P}$

7.5 Types

$'1\ \boldsymbol{CAT}$

7.6 Fixity

Right Infix 310:
$\subseteq_c$ Λ Ξ

Right Infix 400:
$>->$ Υ

Right Infix 500:
$\boldsymbol{o_c}$

Postfix 400:
^{op}

7.7 Definitions

CAT $\vdash \exists\, f\bullet\ \mathit{TypeDefn}\ (\lambda\ x\bullet\ T)\ f$

MkCAT

Arrows

Compose

Left

Right $\vdash \forall\ t\ x1\ x2\ x3\ x4$
$\bullet\ \mathit{Arrows}\ (\mathit{MkCAT}\ x1\ x2\ x3\ x4) = x1$
$\wedge\ \mathit{Compose}\ (\mathit{MkCAT}\ x1\ x2\ x3\ x4) = x2$
$\wedge\ \mathit{Left}\ (\mathit{MkCAT}\ x1\ x2\ x3\ x4) = x3$
$\wedge\ \mathit{Right}\ (\mathit{MkCAT}\ x1\ x2\ x3\ x4) = x4$
$\wedge\ \mathit{MkCAT}\ (\mathit{Arrows}\ t)\ (\mathit{Compose}\ t)\ (\mathit{Left}\ t)\ (\mathit{Right}\ t)$
$= t$

Obj $\vdash \forall\ C\ a\bullet\ a \in \mathit{Obj}\ C \Leftrightarrow a \in \mathit{Arrows}\ C \wedge a = \mathit{Left}\ C\ a$

Υ $\vdash \forall\ f\ g\ C$
$\bullet\ (f\ \Upsilon\ g)\ C$
$\Leftrightarrow f \in \mathit{Arrows}\ C$
$\wedge\ g \in \mathit{Arrows}\ C$
$\wedge\ \mathit{Right}\ C\ f = \mathit{Left}\ C\ g$

Identity $\vdash \forall\ C\ a$
$\bullet\ a \in \mathit{Identity}\ C$
$\Leftrightarrow a \in \mathit{Arrows}\ C$
$\wedge\ \mathit{Left}\ C\ a = a$
$\wedge\ \mathit{Right}\ C\ a = a$
$\wedge\ (\forall\ b$
$\bullet\ b \in \mathit{Arrows}\ C$
$\Rightarrow (\mathit{Left}\ C\ b = a \Rightarrow \mathit{Compose}\ C\ a\ b = b)$
$\wedge\ (\mathit{Right}\ C\ b = a \Rightarrow \mathit{Compose}\ C\ b\ a = b))$

$>\!\!->$ $\vdash \forall\ f\ a\ b\ C$
$\bullet\ f \in (a \rightarrowtail b)\ C$
$\Leftrightarrow f \in \mathit{Arrows}\ C$
$\wedge\ a \in \mathit{Arrows}\ C$
$\wedge\ b \in \mathit{Arrows}\ C$
$\wedge\ \mathit{Left}\ C\ f = a$
$\wedge\ \mathit{Right}\ C\ f = b$

$\boldsymbol{o_c}$ $\vdash \forall\ f\ g\ C\bullet\ (f\ o_c\ g)\ C = \mathit{Compose}\ C\ f\ g$

Cat $\vdash \forall\ _c$
$\bullet\ \mathit{Cat}\ _c$
$\Leftrightarrow (\forall\ x$
$\bullet\ x \in \mathit{Arrows}\ _c$
$\Rightarrow \mathit{Left}\ _c\ x \in \mathit{Identity}\ _c$
$\wedge\ \mathit{Right}\ _c\ x \in \mathit{Identity}\ _c)$
$\wedge\ (\forall\ x\ y$
$\bullet\ (x\ \Upsilon\ y)\ _c$
$\Rightarrow (x\ o_c\ y)\ _c \in \mathit{Arrows}\ _c$
$\wedge\ (x\ o_c\ y)\ _c$
$\in (\mathit{Left}\ _c\ x \rightarrowtail \mathit{Right}\ _c\ y)\ _c)$
$\wedge\ (\forall\ x\ y\ z$
$\bullet\ (x\ \Upsilon\ y)\ _c \wedge (y\ \Upsilon\ z)\ _c$
$\Rightarrow (x\ o_c\ (y\ o_c\ z)\ _c)\ _c$
$= ((x\ o_c\ y)\ _c\ o_c\ z)\ _c)$

$\mathbf{\Lambda}$ $\vdash \forall\ C\ la\ a$
• $C\ \Lambda\ [a] = a$
$\wedge\ (la = []$
$\vee\ C\ \Lambda\ Cons\ a\ la = Compose\ C\ a\ (C\ \Lambda\ la))$

$\mathbf{\Xi}$ $\vdash \forall\ C\ lla$
• $C\ \Xi\ lla$
$\Leftrightarrow\ (\exists\ x\bullet\ Fold\ Insert\ (Map\ (\$\Lambda\ C)\ lla)\ \{\} = \{x\})$

$\mathbf{\varnothing_c}$ $\vdash Arrows\ \varnothing_c = \{\}$

$\mathbf{1_c}$ $\vdash \forall\ x\bullet\ Cat\ (1_c\ x) \wedge Arrows\ (1_c\ x) = \{x\}$

$\boldsymbol{Discrete_c}$ $\vdash \forall\ s$
• $Discrete_c\ s = MkCAT\ s\ (\lambda\ x\ y\bullet\ x)\ (\lambda\ x\bullet\ x)\ (\lambda\ x\bullet\ x)$

$\boldsymbol{Functor}$ $\vdash \forall\ A\ B\ f$
• $f \in Functor\ (A,\ B)$
$\Leftrightarrow\ (\forall\ a\bullet\ a \in Arrows\ A \Rightarrow f\ a \in Arrows\ B)$
$\wedge\ (\forall\ a\ b$
• $(a\ \Upsilon\ b)\ A$
$\Rightarrow Right\ B\ (f\ a) = f\ (Right\ A\ a)$
$\wedge\ Left\ B\ (f\ b) = f\ (Left\ A\ b)$
$\wedge\ Compose\ B\ (f\ a)\ (f\ b)$
$= f\ (Compose\ A\ a\ b))$

$\boldsymbol{Full}$ $\vdash \forall\ A\ B\ f$
• $f \in Full\ (A,\ B)$
$\Leftrightarrow\ (\forall\ b$
• $b \in Arrows\ B \Rightarrow (\exists\ a\bullet\ a \in Arrows\ A \wedge b = f\ a))$

$\boldsymbol{Faithful}$ $\vdash \forall\ A\ B\ f$
• $f \in Faithful\ (A,\ B)$
$\Leftrightarrow\ (\forall\ a\ b$
• $a \in Arrows\ A \wedge b \in Arrows\ A \wedge f\ a = f\ b$
$\Rightarrow a = b)$

$\mathbf{\subseteq_c}$ $\vdash \forall\ A1\ A2$
• $A1 \subseteq_c A2$
$\Leftrightarrow\ (\forall\ a$
• $a \in Arrows\ A1$
$\Rightarrow a \in Arrows\ A2$
$\wedge\ Left\ A1\ a \in Arrows\ A1$
$\wedge\ Right\ A1\ a \in Arrows\ A1$
$\wedge\ Left\ A1\ a = Left\ A2\ a$
$\wedge\ Right\ A1\ a = Right\ A2\ a)$
$\wedge\ (\forall\ a\ b$
• $a \in Arrows\ A1 \wedge b \in Arrows\ A1$
$\Rightarrow Compose\ A1\ a\ b \in Arrows\ A1$
$\wedge\ Compose\ A1\ a\ b = Compose\ A2\ a\ b)$

$\boldsymbol{NatTran}$ $\vdash \forall\ A\ B\ R\ S\ \tau$
• $\tau \in NatTran\ (A,\ B)\ (R,\ S)$
$\Leftrightarrow\ (\forall\ a$
• $a \in Obj\ A$
$\Rightarrow \tau\ a \in Arrows\ B$
$\wedge\ Left\ B\ (\tau\ a) = R\ a$
$\wedge\ Right\ B\ (\tau\ a) = S\ a)$
$\wedge\ (\forall\ f$
• $f \in Arrows\ A$

$$\Rightarrow \mathit{Left}\ B\ (\tau\ (\mathit{Right}\ A\ f)) = R\ f$$
$$\wedge\ \mathit{Right}\ B\ (\tau\ (\mathit{Right}\ A\ f)) = S\ f$$
$$\wedge\ \mathit{Compose}\ B\ (\tau\ (\mathit{Left}\ A\ f))\ (S\ f)$$
$$= \mathit{Compose}\ B\ (R\ f)\ (\tau\ (\mathit{Right}\ A\ f)))$$

Id_{NT}
$$\vdash \forall\ A\ B\ R\ f\bullet\ Id_{NT}\ (A,\ B)\ R\ f = \mathit{Left}\ B\ (R\ f)$$

op
$$\vdash \forall\ C$$
$$\bullet\ C^{\ op}$$
$$= \mathit{MkCAT}$$
$$(\mathit{Arrows}\ C)$$
$$(\lambda\ x\ y\bullet\ \mathit{Compose}\ C\ y\ x)$$
$$(\mathit{Right}\ C)$$
$$(\mathit{Left}\ C)$$

7.8 Theorems

$Cat\_\varnothing_c\_thm$
$$\vdash \mathit{Cat}\ \varnothing_c$$

$Cat\_1_c\_thm$
$$\vdash \forall\ x\bullet\ \mathit{Cat}\ (1_c\ x)$$

discrete_cat_thm
$$\vdash \forall\ s\bullet\ \mathit{Cat}\ (\mathit{Discrete}_c\ s)$$

functor_composition_thm
$$\vdash \forall\ A\ B\ C\ ab\ bc$$
$$\bullet\ ab \in \mathit{Functor}\ (A,\ B) \wedge bc \in \mathit{Functor}\ (B,\ C)$$
$$\Rightarrow bc\ o\ ab \in \mathit{Functor}\ (A,\ C)$$

full_compose_thm
$$\vdash \forall\ A\ B\ C\ ab\ bc$$
$$\bullet\ ab \in \mathit{Full}\ (A,\ B) \wedge bc \in \mathit{Full}\ (B,\ C)$$
$$\Rightarrow bc\ o\ ab \in \mathit{Full}\ (A,\ C)$$

functor_arrow_thm
$$\vdash \forall\ A\ B\ a\ ab$$
$$\bullet\ a \in \mathit{Arrows}\ A \wedge ab \in \mathit{Functor}\ (A,\ B)$$
$$\Rightarrow ab\ a \in \mathit{Arrows}\ B$$

faithful_thm
$$\vdash \forall\ A\ B\ C\ ab\ bc$$
$$\bullet\ ab \in \mathit{Functor}\ (A,\ B)$$
$$\wedge\ ab \in \mathit{Faithful}\ (A,\ B)$$
$$\wedge\ bc \in \mathit{Faithful}\ (B,\ C)$$
$$\Rightarrow bc\ o\ ab \in \mathit{Faithful}\ (A,\ C)$$

8 Index