Abstract sets

Signature $\boxed{L_{absset}}$:

$$\begin{tikzcd}
\text{Equality} \arrow[dr, shift right, "d_1"'] \arrow[dr, shift left, "c_1"] & & \text{Application} \arrow[dl, shift right, "a_3"'] \arrow[dl, shift left, "a_4"] \arrow[d, "a_2"] \\
& \text{Element} \arrow[d, "e"'] & \text{Map} \arrow[dl, shift right, "d"'] \arrow[dl, shift left, "c"] \\
& \text{Set} &
\end{tikzcd}
\qquad
\begin{tikzcd}
\text{Equality} \arrow[dr, "e_0"'] & & \text{App} \arrow[dl, shift right, "a_0"'] \arrow[dl, shift left, "a_1"] \\
& \text{Set} &
\end{tikzcd}$$

$$e_0 = ed_1 = ec_1$$

$$a_0 = ea_3 = da_2$$

$$a_1 = ea_4 = ca_2$$

A <u>concrete</u> category (c-category) is a pair $\underline{A} = (A, a)$ where 'a' is a faithful functor $a : A \to \mathbf{Set}$. $\underline{A}$ is a <u>complete</u> c-category if 'a' is a discrete isofibration. Equivalence for c-cats is more straightforward than for general c-cats.

($\underline{Ab}$, underlying: $\underline{Ab} \to \mathbf{Set}$) is complete, for instance.

Every c-cat can be 'completed'; the underlying category of the completion is equivalent to the original.

Let $\underline{G} = (G, \in_{\underline{G}})$ be a Set-valued model of (a fragment of) ZFC. For instance, $G = L =$ the constructible sets — but if there <u>is</u> an inaccessible cardinal, there is a <u>countable</u> model of ZFC. $\underline{G}$ gives rise to the c-cat

$$\underline{A} = \boxed{\underline{A}[\underline{G}]} = (A, a) \quad \text{where}$$

$$\mathrm{Ob}(A) = G$$

$$\mathrm{Arr}(A) = \{(x, y, f) : x, y, \text{“}f : x \to y\text{” in the sense of } \underline{G}\}$$

L_{absset}

for $x \in G$, $a(x) = \{ y \in G \mid y \in_{\underline{G}} x \}$

(etc)

Re-coding c-cat $\underline{A} = (A, a)$ into

$\rightsquigarrow \boxed{M(\underline{A})} : L_{absset} \to$ Set
SET?

$M = M(\underline{A})$ ::

$M(\mathrm{Set}) = \mathrm{Ob}(A)$

$M(\mathrm{El}) = \{ (U, u) \mid U \in \mathrm{Ob}(A),\ u \in a(U) \}$

$M(\mathrm{Map}) = \{ (U, V, f) \mid U, V \in \mathrm{Ob}(A);\ f : U \to V \}$

$M(\mathrm{Eq}) = \{ (U, u, v) \mid U \in \mathrm{Ob}(A);\ u, v \in a(U), u = v \}$

$M(\mathrm{App}) = \{ \underbrace{(U, V, f, u, v)}_{a_0\ a_1\ a_2\ a_3\ a_4} \mid U, V \in \mathrm{Ob}(A);\ f : U \to V,$

$u \in a(U),\ v \in a(V)$

$\boxed{v = (a(f))(u)} \}$

For the syntax, we need the five type-formation rules:

1 $\quad :: \; U : \mathrm{Set}$

2 $\quad U : \mathrm{Set} \;\; :: \;\; u : \mathrm{El}(U)$

3 $\quad U : \mathrm{Set}.\; u, v : \mathrm{El}(U) \;\; :: \;\; e : \mathrm{Eq}(U, u, v)$

4 $\quad U, V : \mathrm{Set} \;\; :: \;\; f : \mathrm{Map}(U, V)$

5 $\quad U, V : \mathrm{Set}.\; u : \mathrm{El}(U).\; v : \mathrm{El}(V).\; f : \mathrm{Map}(U, V)$

$$:: \;\; a : \mathrm{App}(U, V, u, v, f)$$

Defined concepts (syntactic sugar):

examples: $\boxed{\text{Example 1:}}$

$X : \mathrm{Set}.\; x : \mathrm{El}(X).\; y \in \mathrm{El}(X) \;\; ::$

$$\boxed{x = y} \;\; \underset{\text{def}}{\equiv} \;\; \exists e : \mathrm{Eq}(X, x, y) \,.\, \mathrm{TRUE}$$

(free variables: X, x, y)

a proposition (formula)

Example 2:

$X, Y : Set.\ x : El(X).\ y \in El(Y)$

$f : Map(X, Y)$ ::

$f(x) = y \ \underset{def}{\equiv} \ \exists a : App(X, Y, $ [illegible] $, y).\ TRUE$

Example 3:

Context $X \xleftarrow{\pi_0} Z \xrightarrow{\pi_1} Y$ (right?)

:: $\boxed{Product(X, Y, Z, \pi_0, \pi_1)} \ \underset{def}{\equiv}$

$\underbrace{\forall x \in X}_{\forall x : El(X)}.\ \forall y \in Y.\ \exists z \in Z.\ \forall w \in Z$

$[(\pi_0(w) = x \ \&\ \pi_1(w) = y) \Leftrightarrow w = z]$

Example 4: axiom of existence of binary product

$\boxed{ExProd} \ \underset{def}{\equiv} \ \forall (X, Y)\ \exists (Z, \pi_0, \pi_1)\ Product(X, Y, Z, \pi_0, \pi_1)$

For $\underline{G}$ model of set-theory

$$M(\underline{G}) \underset{def}{=} M(\underline{A}[\underline{G}]) \models \text{ExProd}$$

is <u>true</u> since for $X \in G$, $Y \in G$

we can take $Z = \{\underbrace{(x,y)_{\underline{G}}}_{\text{ordered pair in the sense of } \underline{G}} \mid x \in_{\underline{G}} X,\ y \in_{\underline{G}} Y\}$

The universal property of the product can be expressed and deduced from suitable instances of an

"arrow-existence" comprehension

axiom schema in FOLDS(L_{absset})

which is true in $M(\underline{G})$ if $\underline{G} \models ZFC$

A global umbrella theory of structures (such as Bourbaki's Set Theory) is <u>Bourbakian</u> (new word!) if all its definable theoretical predicates $P(A_0, A_1, \ldots)$ referring to structures $A_0, A_1, \ldots$ of respective species $S_1, S_2, \ldots$ are <u>invariant</u> under the identities $=_{S_1}, =_{S_2}, \ldots$ appropriate (adopted to be appropriate) for the species $S_1, S_2, \ldots$:

$$A_0 =_{S_0} B_0,\ A_1 =_{S_1} B_1, \ldots\ P(A_0, A_1, \ldots) \Rightarrow P(B_0, B_1, \ldots)$$

Bourbaki's own Set Theory is <u>not</u> Bourbakian; Lawvere's first-order theory of the category of sets is <u>not</u> Bourbakian <u>if</u> "isomorphism" is the adopted notion of identity for the usual species of structures (groups, topological spaces, etc). Reason:

Consider the species-of-structures of <u>bare sets</u>, both as S_0 and as S_1,

or, objects of the category Set in Lawvere's case.

The language of the umbrella theory allows the predicate

$$P(A_0, A_1) \underset{def}{\equiv} A_0 = A_1$$

since equality, $=$, is a theoretical predicate in the first-order theory of a universe of sets, and also, in the first-order theory of a category. But, of course, invariance fails:

$$A_0 \cong B_0 , \quad A_1 \cong B_1 , \quad A_0 = A_1$$

isomorphism; equipollence

$$\not\Rightarrow \quad B_0 = B_1$$

Abstract Set Theory $\underset{\sim}{\text{is}}$ Bourbakian in the new sense since in it all theoretical predicates are invariant under isomorphism (intrinsic equivalence).

Internal topos theory .

becomes Bourbakian if for category theory the FOLDS language over L_{cat} is adopted

— but not without such a move.

$\boxed{\text{Sanafunctors}}$

Signature $\boxed{L_{sanafun}}$:

($L_{sanafun}$ is the same as L_{anafun} used before; since the intended models now are sanafunctors = saturated anafunctors, a subclass of anafunctors in general, I changed the notation)

$L = L_{sanafun}$ contains two copies of L_{cat}, for the domain and the codomain categories of the functor being considered. L contains two additional kinds, App_0 and App_1. These are like App in L_{absset}, for function-applications, on objects in case of App_0, and on arrows in case App_1. The twist is that for a functor $F: \mathbb{X} \to \mathbb{A}$, the application instance $(X, A, F(X)=A)$, a would-be element of App_0, is "undesirable" because it uses equality of objects of the category $\mathbb{A}$. We change this by replacing $F(X) = A$ by $F(X) \cong A$ and even better, by a any particular isomorphism $i : F(X) \xrightarrow{\cong} A$. Once this done the concept of sanafunctor will be readily obtained:

a free-living sanafunctor $\mathbb{F} : \mathbb{X} \xrightarrow{sana} \mathbb{A}$ is a model of the FOLDS($L_{sanafun}$)-theory of the class of of all $L_{sanafun}$-structures of the form $M(\mathbb{F} : \mathbb{X} \to \mathbb{A})$ where $M(\mathbb{F})$ is, of course, the $L_{sanafun}$-recoding of $\mathbb{F}$. To make this reasonable and explicit, we establish the analog's of the propositions stated in the treatment of categories in L_{cat}.

A difference to the L_{cat} case is that coming back from a sanafunctor to an ordinary functor needs an essential use of the axiom of choice

$\boxed{L_{sanafun}}$:

$$\begin{tikzcd}
I_0 \arrow[dr] & E_0 \arrow[d, shift left] \arrow[d, shift right] & T_0 \arrow[dl] \arrow[dl, shift left] \arrow[dl, shift right] & & I_1 \arrow[dr] & E_1 \arrow[d, shift left] \arrow[d, shift right] & T_1 \arrow[dl] \arrow[dl, shift left] \arrow[dl, shift right] \\
& A_0 \arrow[d, shift right, "d_0"'] \arrow[d, shift left, "c_0"] & & App_1 \arrow[ll, "a_2"'] \arrow[rr, "a_3"] \arrow[d, shift right, "d_2"'] \arrow[d, shift left, "c_2"] & & A_1 \arrow[d, shift right, "d_1"'] \arrow[d, shift left, "c_1"] & \\
& O_0 & & App_0 \arrow[ll, "a_0"] \arrow[rr, "a_1"'] & & O_1 &
\end{tikzcd}$$

$$d_0 a_2 = a_0 d_2 \qquad d_1 a_3 = a_1 d_2$$

$$c_0 a_2 = a_0 c_2 \qquad c_1 a_3 = a_1 c_2$$

$F : \mathbb{X} \to \mathbb{A}$ functor $\longmapsto$ $\boxed{M(F)}: L_{sanfun} \to SET$

as follows. The domain & codomain parts by what happens for L_{cat}. Remains: for $M = M(F$

$M(App_0)$, $M(App_1)$ and the related arrows

$$\boxed{M(App_0)} = \{ (X, A, i) \mid X \in ob(\mathbb{X}),\ A \in ob(\mathbb{A}),\ i : F(X) \xrightarrow{\cong} A \}$$

$$\boxed{M(App_1)} = \{ ((X,A,i), (Y,B,j), f : X \to Y \text{ in } \mathbb{X}, g : A \to B \text{ in } \mathbb{A}) \mid$$

$$\begin{tikzcd}
FX \arrow[r, "i", "\cong"'] \arrow[d, "Ff"'] \arrow[dr, phantom, "\equiv"] & A \arrow[d, "g"] \\
FY \arrow[r, "j", "\cong"'] & B
\end{tikzcd}
\text{ commutes} \}$$

The action on the lower (dim ≤ 1) part of L_{sana} is:

$$\begin{tikzcd}
f \arrow[d, mapsto] \arrow[dd, mapsto, bend left] & ((X,A,i), (Y,B,j), f, g) \arrow[l, mapsto] \arrow[r, mapsto] \arrow[d, mapsto] \arrow[dd, mapsto, bend left] & g \arrow[d, mapsto] \arrow[dd, mapsto, bend left] \\
X & (X, A, i) \arrow[l, mapsto] \arrow[r, mapsto] & A \\
Y & (Y, B, j) \arrow[l, mapsto] \arrow[r, mapsto] & B
\end{tikzcd}$$