Topos Theory in Free Logic

Lucca Tiemens

July 26, 2020

Let me start by stating the axioms for a category again for the sake of completeness.

Definition 1. A structure $\mathbf{C} = (C, dom, cod, \cdot)$ is called a category if and only if it satisfies the following axioms where x, y and $z \in \mathbf{C}$:

- $E(\operatorname{dom} x) \longrightarrow E(x)$
- $E(\operatorname{cod} y) \longrightarrow E(y)$
- $E(x \cdot y) \longleftrightarrow \operatorname{dom} x \simeq \operatorname{cod} y$
- $x \cdot (y \cdot z) \cong (x \cdot y) \cdot z$
- $x \cdot (\operatorname{dom} x) \cong x$
- $(\operatorname{cod} y \cdot y) \cong y.$

1 Basic Definitions

In the following let $\mathbf{C}$ always denote a category. We have the convenient abbreviation *type* denoting an *existing* identity arrow.

Definition 2 (type)**.** $x \in \mathbf{C}$ is called a type if and only if

$$x \simeq \operatorname{dom} x$$

Furthermore, we adopt notation from category theory for *existing* arrows.

Definition 3 (existing arrow)**.** For $x, a, b \in \mathbf{C}$ writing $x : a \to b$ abbreviates

$$\operatorname{dom} x \simeq a \wedge \operatorname{cod} x \simeq b$$

For an arrow which does not necessarily exist, we also introduce notation.

Definition 4 (general arrow)**.** For $x, a, b \in \mathbf{C}$ writing $x : a \Rightarrow b$ abbreviates

$$\operatorname{dom} x \cong a \wedge \operatorname{cod} x \cong b$$

A commutative square is easy to write down. The biggest challenge is to remember the order in which the arguments are stated. I have adopted the habit of starting at the lower right corner and then moving counter-clockwise.

Definition 5 (commutative square)**.** For $f, g, p, q \in \mathbf{C}$ we define

$$\text{commSquare}\ (g,\ p,\ q,\ f) \Longleftrightarrow g \cdot p \cong f \cdot q$$

Note, that we do not explicitly need to say what the domains and codomains are. This can be inferred by using the functions *dom* and *cod*.

At this point I will recall the definition of a product. $p1$ and $p2$ should be thought of as the projection maps. Also note, that $\forall_f$ ($\exists_f$) denotes the *free* universal (existential) quantifier.

Definition 6. For $a, b, c, p1, p2 \in \mathbf{C}$ we define

$$\begin{aligned}\text{product}\ (a,\ b,\ c,\ p1,\ p2) \Longleftrightarrow\ & p1 : c \Rightarrow a \wedge p2 : x \Rightarrow b \wedge \\ & \forall_f xfg.\,(f : x \to a \wedge g : x \to b) \longrightarrow \\ & \exists_f!h.\,(h : x \to c \wedge f \simeq p1 \cdot h \wedge g \simeq p2 \cdot h)\end{aligned}$$

2 Category with binary products

We are now in the position to write down the axioms for a category that has binary products. Taking the product between two elements of a category is defined as a primitive operation. It is defined between types as well as between arrows. Furthermore, the projection arrows are skolemized.

Definition 7. A category with binary products is a category $\mathbf{C}$ together with maps $\otimes : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$, $p1 : \mathbf{C} \to \mathbf{C}$ and $p2 : \mathbf{C} \to \mathbf{C}$ such that

1. $E\,(a \otimes b) \longrightarrow E\,(a) \wedge E\,(b)$
2. $E\,(p1(a)) \longrightarrow E\,(a)$
3. $E\,(p2(b)) \longrightarrow E\,(b)$
4. type a $\wedge$ type b $\longrightarrow$ product $(a,\ b,\ (a \otimes b),\ (p1(a \otimes b)),\ (p2(a \otimes b)))$
5. commSquare $(a,\ p1(\text{dom}\, a \otimes \text{dom}\, b),\ a \otimes b,\ p1(\text{cod}\, a \otimes \text{cod}\, b))$
6. commSquare $(b,\ p2(\text{dom}\, a \otimes \text{dom}\, b),\ a \otimes b,\ p2(\text{cod}\, a \otimes \text{cod}\, b))$

Axiom 4 makes $a \otimes b$ the product of a and b. Axioms 5 and 6 enable us to form the product between arrows as well. Let $a : x \to y$ and $b : v \to w$ be arrows, $a \otimes b$ then denotes the unique arrow $< a, b >: x \otimes v \to y \otimes w$.

Note, that this definition does not force existence of elements. There are models of only non-existing elements - the easiest one being just one non-existing element. Therefore, we have not excluded the empty category in our axiomatization.

3 Cartesian Category

The next particular category I want to axiomatize is a Cartesian category. By such a category I mean a category with *binary* products, that has a *final* type as well as *equalizers*. I will define the required notions below. A different way to view a Cartesian category is thus a category with all *finite* limits. However, I have not formulated the notion of a general limit in our formalism. It seems to me that this will not work well with theorem proving because of the *cone* constructions involved. Do you think we need the general definition for Topos Theory or are we okay with using finite products and equalizers?

Let's get started with the definition of a final type. We did this before, however, there was a small mistake in the original formulation. The correct definition is

Definition 8 (final type)**.** Let $z \in \mathbf{C}$. We then define

$$\text{final } z \iff \forall_f t.\, (\text{type } t) \longrightarrow (\exists_f ! f.\, f : t \to z)$$

In the earlier definition the condition of t being a type was missing. Without this condition a category with a final type does not have any arrows except identity-arrows.

Definition 9 (equalizer between two arrows)**.** Let $f, g, e \in \mathbf{C}$. Then we define

$$\begin{aligned}\text{equalizer } (f, g, e) \iff f \cdot e \simeq g \cdot e \wedge \\ \forall_f z.\, (f \cdot z \simeq g \cdot z) \longrightarrow (\exists_f ! u.\, u : \operatorname{dom} z \to \operatorname{dom} e \wedge e \cdot u \simeq z)\end{aligned}$$

Note that if f and g are not parallel, i.e. they do not agree on the domain or on the codomain, then equalizer (f, g, e) will always be false and hence there is no equalizer.

Now we are in the position to formulate a Cartesian category.

Definition 10 (Cartesian category)**.** A Cartesian category $\mathbf{C}$ is a category with binary products, maps, called $\doteq : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ and $!_1 : \mathbf{C} \to \mathbf{C}$, and a constant $\mathbf{1} \in \mathbf{C}$ such that

1. $(f : \operatorname{dom} g \to \operatorname{cod} g) \longrightarrow \text{equalizer } (f, g, (f \doteq g))$
2. final $\mathbf{1}$
3. type $t \longrightarrow (!_1\, t) : t \to 1$
4. $E(!_1\, t) \longrightarrow \text{type } t$
5. $E(f \doteq g) \longrightarrow f : \operatorname{dom} g \to \operatorname{cod} g$