Equational Logical Frameworks as Locally Cartesian Closed Categories

Joseph Hua

April 22, 2023

Contents

1 Overview

In section 2, we present the syntax for an *equational logical framework* (LF for short). In section 3, we set out the main categorical definitions that are relevant to the discussion. In section 4, we describe the category of contexts for the LF. The main goal is to identify two display map subcategories of the category of contexts, S_π.

1.1 Potential points of confusion

It is *not true* that the category of contexts $\mathcal{C}$ for the LF is locally Cartesian closed. However, we can find two display map subcategories $\mathcal{S}_\Pi \hookrightarrow \mathcal{S}_{\Pi,=} \hookrightarrow \mathcal{C}$ that are locally Cartesian closed, and these correspond to the Sort or universe for the LF. It is *not true* that general locally closed Cartesian categories are suitable general semantics for the LF, or even for the subcategory, due to coherence issues for pullback. For general semantics, we should use categories with families [CCD20], categories with attributes [Jac93], or natural models [Awo17]. One can verify that the category of contexts satisfies any of these semantics.

1.2 Terminology

We follow [Har21] and use the word "class" for what is usually called "type" and the word "object" for what is usually called "term". This is to distinguish between the meta-level class in the logical framework and types internal to the type theory defined in the logical framework. To not confuse this with the categorical notion of object, we will write $\mathcal{C}$-object for an object in a category $\mathcal{C}$.

2 An Equational Logical Framework

$$\frac{}{\bullet \vdash \mathsf{ctx}}\ \textsc{ctx-emp} \qquad \frac{\Gamma \vdash K\ \mathsf{cls}}{\Gamma, X : K \vdash \mathsf{ctx}}\ \textsc{ctx-ext} \qquad \frac{}{X : K \in \Gamma, X : K}\ \textsc{ctx-hd} \qquad \frac{X_0 : K_0 \in \Gamma}{X_0 : K_0 \in \Gamma, X_1 : K_1}\ \textsc{ctx-tl}$$

Figure 1: Context formation

$$\frac{\Gamma \vdash \mathsf{ctx}}{\Gamma \vdash \mathsf{Sort}\ \mathsf{cls}}\ \textsc{sort-cls} \qquad \frac{\Gamma \vdash S\ :\ \mathsf{Sort}}{\Gamma \vdash S\ \mathsf{cls}}\ \textsc{incl-cls} \qquad \frac{\Gamma \vdash S\ :\ \mathsf{Sort} \quad \Gamma, X : S \vdash K\ \mathsf{cls}}{\Gamma \vdash \{X\ :\ S\}\, K\ \mathsf{cls}}\ \textsc{pi-cls}$$

Figure 2: Class formation

$$\frac{\Gamma \vdash S_0\ :\ \mathsf{Sort} \quad \Gamma, X : S_0 \vdash S_1\ :\ \mathsf{Sort}}{\Gamma \vdash \{X\ :\ S_0\}\, S_1\ :\ \mathsf{Sort}}\ \textsc{pi-sort} \qquad \frac{\Gamma \vdash S\ :\ \mathsf{Sort} \quad \Gamma \vdash O_0\ :\ S \quad \Gamma \vdash O_1\ :\ S}{\Gamma \vdash O_0 =_S O_1\ :\ \mathsf{Sort}}\ \textsc{eq-sort}$$

Figure 3: Sort formation

In this section we recall the equational logical framework, as described in [Har21] (with some minor changes). This LF can be used to define a wide variety of type theories, including dependent type theories such as Martin Löf type theory or ModTT [SH21]. Within the LF we define type theories by asserting the rules of the type theory via a collection of generators, called a signature and denoted Σ. We state only state the syntax and judgment rules and refer to [Har21] for details and examples.

$$\begin{array}{lrl} \textit{variable} & & X \\ \textit{expression} & O, S, K, e ::= & X \mid \mathsf{Sort} \mid \{X\ :\ S\}\, e \mid O_0 =_S O_1 \mid \mathsf{self} \mid [X\ :\ S]\, O \mid O_0\, O_1 \\ \textit{context} & \Gamma, \Delta ::= & \cdot \mid \Gamma, X : K \end{array}$$

O indicates expressions representing objects (though just an expression at this point), similarly S represents sorts and K represents classes. The agnostic expression symbol e represents sorts or a classes.

$$\frac{\Gamma \vdash \mathsf{ctx} \quad X : K \in \Gamma}{\Gamma \vdash X \,:\, K}\ \textsc{var-obj} \qquad \frac{\Gamma \vdash S \,:\, \mathsf{Sort} \quad \Gamma, X : S \vdash O \,:\, K}{\Gamma \vdash [X \,:\, S]\, O \,:\, \{X \,:\, S\}\, K}\ \textsc{pi-lam-obj}$$

$$\frac{\Gamma \vdash O_0 \,:\, \{X \,:\, S\}\, K \quad \Gamma \vdash O_1 \,:\, S}{\Gamma \vdash O_0\, O_1 \,:\, [O_1/X]K}\ \textsc{pi-app-obj} \qquad \frac{\Gamma \vdash O \,:\, K}{\Gamma \vdash \mathsf{self} \,:\, O =_K O}\ \textsc{eq-self-obj}$$

$$\frac{\Gamma \vdash O \,:\, K_0 \quad \Gamma \vdash K_0 = K_1\ \mathsf{cls}}{\Gamma \vdash O \,:\, K_1}\ \textsc{je-obj}$$

Figure 4: Object formation

The judgments we define are

$\Gamma \vdash \mathsf{ctx}$	Γ is a context
$X : K \in \Gamma$	$X \,:\, K$ appears in context Γ
$\Gamma \vdash K\ \mathsf{cls}$	K is a class in context Γ
$\Gamma \vdash O \,:\, K$	O is an object of class K in context Γ
$\Gamma \vdash K_0 = K_1\ \mathsf{cls}$	K_0, K_1 are equal classes in context Γ
$\Gamma \vdash O_0 = O_1 \,:\, K$	O_0, O_1 are equal objects of class K in context Γ

The legal judgments given by the closure under the judgment rules in figures 1, 2, 3, 4, 5.

Note that unlike in [Har21] we have EQ-SORT creating a sort rather than a class. This will ensure one of our display categories is an LCCC.

Judgmental equality for classes $\Gamma \vdash K_0 = K_1\ \mathsf{cls}$ and objects $\Gamma \vdash O_0 = O_1 \,:\, K$ are defined to be congruences, i.e. equivalence relations that respect formation of classes, sorts, and objects. Furthermore, we have β and η rules for PI.

Finally, signatures are defined as contexts $\Sigma \vdash \mathsf{ctx}$, and the type theory generated by a signature Σ consists of legal judgments over Σ. Note that before introducing a signature we cannot make any sorts, nor can we make any interesting objects. The only rules for their formation are via PI-SORT and EQ-SORT, which require more sorts and objects as premises. It is the signature that populates Sort with sorts and populates those sorts with objects. Given a signature Σ, we write $\lambda^{\Pi\mathsf{Eq}}[\Sigma]$ to denote the generated type theory.

$\Gamma \vdash_\Sigma \mathsf{ctx}$	:=	$\Sigma, \Gamma \vdash \mathsf{ctx}$
$\Gamma \vdash_\Sigma K\ \mathsf{cls}$	:=	$\Sigma, \Gamma \vdash K\ \mathsf{cls}$
$\Gamma \vdash_\Sigma O \,:\, K$	:=	$\Sigma, \Gamma \vdash O \,:\, K$
$\Gamma \vdash_\Sigma K_0 = K_1\ \mathsf{cls}$	:=	$\Sigma, \Gamma \vdash K_0 = K_1\ \mathsf{cls}$
$\Gamma \vdash_\Sigma O_0 = O_1 \,:\, K$	:=	$\Sigma, \Gamma \vdash O_0 = O_1 \,:\, K$

3 Categorical definitions

Definition – Cartesian closed category (CCC)
We say a category $\mathcal{C}$ is cartesian closed when

- $\mathcal{C}$ has finite products

$$\frac{\Gamma \vdash K \ \mathsf{cls}}{\Gamma \vdash K = K \ \mathsf{cls}} \ \textsc{cls-rfl} \qquad \frac{\Gamma \vdash K_0 = K_2 \ \mathsf{cls} \qquad \Gamma \vdash K_1 = K_2 \ \mathsf{cls}}{\Gamma \vdash K_0 = K_1 \ \mathsf{cls}} \ \textsc{cls-st} \qquad \frac{\Gamma \vdash O \ : \ K}{\Gamma \vdash O = O \ : \ K} \ \textsc{obj-rfl}$$

$$\frac{\Gamma \vdash O_0 = O_2 \ : \ K \qquad \Gamma \vdash O_1 = O_2 \ : \ K}{\Gamma \vdash O_0 = O_1 \ : \ K} \ \textsc{obj-st} \qquad \frac{\Gamma \vdash O_0 = O_1 \ : \ K_0 \qquad \Gamma \vdash K_0 = K_1 \ \mathsf{cls}}{\Gamma \vdash O_0 = O_1 \ : \ K_1} \ \textsc{obj-je-cls}$$

$$\frac{\Gamma, X : S \vdash O_0 \ : \ K \qquad \Gamma \vdash O_1 \ : \ S}{\Gamma \vdash ([X \ : \ S]\, O_0) O_1 = [O_1/X] O_0 \ : \ [O_1/X] K} \ \textsc{app-lam} \qquad \frac{\Gamma \vdash O \ : \ \{X \ : \ S\}\, K}{\Gamma \vdash O = [X \ : \ S]\,(O X) \ : \ \{X \ : \ S\}\, K} \ \textsc{lam-app}$$

$$\frac{\Gamma \vdash O_0 \ : \ O_1 =_S O_2}{\Gamma \vdash O_1 = O_2 \ : \ S} \ \textsc{reflection} \qquad \frac{\Gamma \vdash O_0 \ : \ O_1 =_S O_2 \qquad \Gamma \vdash O_3 \ : \ O_1 =_S O_2}{\Gamma \vdash O_0 = O_3 \ : \ O_1 =_S O_2} \ \textsc{unicity}$$

$$\frac{\Gamma \vdash S_0 = S_1 \ : \ \mathsf{Sort}}{\Gamma \vdash S_0 = S_1 \ \mathsf{cls}} \ \textsc{je-incl-cls} \qquad \frac{\Gamma \vdash S_0 = S_1 \ : \ \mathsf{Sort} \quad \Gamma, X : S_0 \vdash K_0 = K_1 \ \mathsf{cls}}{\Gamma \vdash \{X \ : \ S_0\}\, K_0 = \{X \ : \ S_1\}\, K_1 \ \mathsf{cls}} \ \textsc{je-pi-cls}$$

$$\frac{\Gamma \vdash S_0 = S_1 \ : \ \mathsf{Sort} \quad \Gamma, X : S_0 \vdash S_2 = S_3 \ : \ \mathsf{Sort}}{\Gamma \vdash \{X \ : \ S_0\}\, S_2 = \{X \ : \ S_1\}\, S_3 \ : \ \mathsf{Sort}} \ \textsc{je-pi-sort}$$

$$\frac{\Gamma \vdash S_0 = S_1 \ : \ \mathsf{Sort} \quad \Gamma \vdash O_0 = O_1 \ : \ S_0 \quad \Gamma \vdash O_2 = O_3 \ : \ S_1}{\Gamma \vdash O_0 =_{S_0} O_2 = O_1 =_{S_0} O_3 \ : \ \mathsf{Sort}} \ \textsc{je-eq-sort}$$

$$\frac{\Gamma \vdash S_0 = S_1 \ : \ \mathsf{Sort} \quad \Gamma, X : S_0 \vdash O_0 = O_1 \ : \ K}{\Gamma \vdash [X \ : \ S_0]\, O = [X \ : \ S_1]\, O_1 \ : \ \{X \ : \ S_0\}\, K} \ \textsc{je-pi-lam-obj}$$

$$\frac{\Gamma \vdash O_0 = O_1 \ : \ \{X \ : \ S\}\, K \quad \Gamma \vdash O_2 = O_3 \ : \ S}{\Gamma \vdash O_0\, O_2 = O_1\, O_3 \ : \ [O_2/X] K} \ \textsc{je-pi-app-obj}$$

Figure 5: Judgmental equality

- For each $\mathcal{C}$-object A, the product with A functor $\times A : \mathcal{C} \to \mathcal{C}$ has a right adjoint.

We denote the right adjoint by $[A, -]$ and call it internal hom.

Definition – Locally cartesian closed category (LCCC)
We say a category $\mathcal{C}$ is locally cartesian closed when either of the following equivalent definitions hold

1. Every slice category $\mathcal{C}/A$ over a $\mathcal{C}$-object A is cartesian closed.
2. $\mathcal{C}$ has pullbacks, and for each morphism $f : B \to A$ in $\mathcal{C}$, the base change (pullback along f) $f^* : \mathcal{C}/A \to \mathcal{C}/B$ has a right adjoint.

We denote the right adjoint to base change by Π_f.

We do not show that these two definitions are equivalent. The idea is that existence of pullbacks correspond to existence of products in slices and Πs correspond to internal homs in slices.

$$B \xrightarrow{f} A \qquad \begin{tikzcd}
\mathcal{C}/A \arrow[d, shift right=2, "f^*"'] \arrow[dd, bend right=60, "\times f"'] \\
\mathcal{C}/B \arrow[u, shift right=2, "\Pi_f"'] \arrow[u, phantom, "\dashv"] \arrow[d, shift right=2, "\Sigma_f"'] \\
\mathcal{C}/A \arrow[u, shift right=2, "f^*"'] \arrow[uu, bend right=60, "-^f"']
\end{tikzcd}$$

Definition – Display map category
Let $\mathcal{S}$ be a set of morphisms in category $\mathcal{C}$. We say $\mathcal{S}$ is a set of display maps when for any compatible pair of morphisms $f : A \to B$ in $\mathcal{C}$ and $g : C \to B$ in $\mathcal{S}$ the pullback exists and $f^*g \in \mathcal{S}$.

4 The category of contexts

I order to find categorical semantics for a type theory it is useful to study the category of contexts. In this section we construct the category of contexts. We will call this category $\mathcal{C}$ for the rest of this section.

We begin by recalling $\lambda^{\times\to}$, simply typed λ-calculus with only unit, product and function classes. And use intuition of its categorical semantics to motivate those for logical frameworks. Recall that $\lambda^{\times\to}$ can be interpreted in a CCC $\mathcal{D}$ by taking classes K as $\mathcal{D}$-objects $[\![K]\!]$, contexts $X_1 : K_1, \ldots, X_n : K_n$ as products $[\![K_1]\!] \times \cdots \times [\![K_n]\!]$, and objects $\Gamma \vdash K : O$ as morphisms $[\![O]\!] : [\![\Gamma]\!] \to [\![K]\!]$. This seems quite natural, since product classes become products, and function classes become internal homs in the semantics. If we are to immitate this for $\lambda^{\Pi\mathsf{Eq}}[\Sigma]$, we would only be able to talk about *closed classes* $\bullet \vdash K$ cls, since in $\lambda^{\times\to}$ we don't have dependent classes $\Gamma \vdash K$ cls. However, in the presence of dependent pairs (aka Σ-types) we could view a context $\Gamma = X_1 : K_1, \ldots, X_n : K_n$ as a closed class $\Sigma_\Gamma := \Sigma_{X_1:K_1} \cdots \Sigma_{X_{n-1}:K_{n-1}} K_n$. Then any dependent class can be viewed as a context, which can be viewed as a closed class.

4.1 Contexts and dependent classes

Instead of including dependent pairs, let us view dependent classes $\Gamma \vdash K$ cls as extended contexts $\Gamma, X : K$, and take *contexts* (up to judgmental equality) as $\mathcal{C}$-objects, and *as a consequence* dependent classes as $\mathcal{C}$-objects.

$$\Gamma \vdash \mathsf{ctx} \qquad\qquad \llbracket \Gamma \rrbracket \in \mathcal{C}$$

It is then natural to define morphisms of contexts (aka a substitution) (up to judgmental equality), denoted $\Gamma \vdash \sigma : \Delta$, consisting of the data of objects

$$\begin{aligned} &\Gamma \vdash \sigma_1 : K_1 \\ &\Gamma \vdash \sigma_2 : [\sigma_1/X_1]K_2 \\ &\vdots \\ &\Gamma \vdash \sigma_n : [\sigma_{n-1}/X_{n-1}]\cdots[\sigma_1/X_1]K_n \end{aligned} \qquad \text{when } \Delta = X_1 : K_1, \cdots, X_n : K_n$$

Such a substitution corresponds to a closed object of the closed class $\Sigma_\Gamma \to \Sigma_\Delta$ in the presence of dependent pairs.

$$\Gamma \vdash \sigma : \Delta \qquad\qquad \llbracket \sigma \rrbracket : \llbracket \Gamma \rrbracket \to \llbracket \Delta \rrbracket$$

The empty context gives an object $\llbracket \bullet \rrbracket$, and for any context Γ there is exactly one substitution $\Gamma \vdash \sigma : \bullet$, given by the data of no objects. This makes $\llbracket \bullet \rrbracket$ the terminal object.

$$\frac{}{\bullet \vdash \mathsf{ctx}}\ \textsc{ctx-emp} \qquad\qquad \mathbf{1}_{\mathcal{C}} \in \mathcal{C}$$

For context extension, given $\Gamma \vdash K\ \mathsf{cls}$, we obtain a substitution $\Gamma, X : K \vdash \sigma : \Gamma$ by $\Gamma, X : K \vdash X_i : K_i$ for each $X_i : K_i \in \Gamma$. Then let's take the interpretation of K to be a $\mathcal{C}/\llbracket \Gamma \rrbracket$-object, where $\mathcal{C}/\llbracket \Gamma \rrbracket$ is the slice category.

$$\llbracket \Gamma \vdash K\ \mathsf{cls} \rrbracket := \llbracket \Gamma, X : K \rrbracket \text{ and } \llbracket \Gamma, X : K \vdash \sigma : \Gamma \rrbracket$$

We can informally identify use the slice-object $\llbracket \Gamma \vdash K\ \mathsf{cls} \rrbracket$ to denote the underlying morphism.

$$\frac{\Gamma \vdash K\ \mathsf{cls}}{\Gamma, X : K \vdash \mathsf{ctx}}\ \textsc{ctx-ext} \qquad\qquad \llbracket \Gamma, X : K \rrbracket \xrightarrow{\llbracket \Gamma \vdash K\ \mathsf{cls} \rrbracket} \llbracket \Gamma \rrbracket$$

4.2 Judgmental Equality

Technically we should we taking $\mathcal{C}$-objects to be the set of contexts quotiented by the equivalence relation induced by judgmental equality on classes, i.e. $X_1 : K_1, \cdots, X_n : K_n \sim Y_1 : K_1', \cdots, Y_n : K_n'$ if and only if

$$\begin{aligned} \bullet \vdash K_1 &= K_1'\ \mathsf{cls} \\ X_1 : K_1 \vdash K_2 &= K_2'\ \mathsf{cls} \\ &\vdots \\ X_1 : K_1, \cdots, X_{n-1} : K_{n-1} \vdash K_n &= K_n'\ \mathsf{cls} \end{aligned}$$

We always denote an equivalence class using a representative $\llbracket \Gamma \rrbracket$, and omit the quotient-related proofs.

We should also be taking $\mathcal{C}$-morphisms from $\llbracket\Gamma\rrbracket$ to $\llbracket\Delta\rrbracket$ to be substitutions quotiented by the equivalence relation induced by judgmental equality on terms, i.e. $\Gamma \vdash \sigma \,:\, \Delta \sim E \vdash \rho \,:\, Z$ if and only if $\llbracket\Gamma\rrbracket = \llbracket E\rrbracket$ and $\llbracket\Delta\rrbracket = \llbracket Z\rrbracket$ and

$$\begin{array}{ll} \Gamma \vdash \sigma_1 = \rho_1 \,:\, K_1 & \\ \Gamma \vdash \sigma_2 = \rho_2 \,:\, [\sigma_1/X_1]K_2 & \\ \vdots & \text{when } \Delta = X_1 : K_1, \cdots, X_n : K_n \\ \Gamma \vdash \sigma_n = \rho_n \,:\, [\sigma_{n-1}/X_{n-1}]\cdots[\sigma_1/X_1]K_n & \end{array}$$

Likewise, we always denote an equivalence class using a representative $\llbracket\sigma\rrbracket$, and omit the quotient-related proofs.

4.3 Objects

In the $\lambda^{\times\to}$ case we interpreted objects $\Gamma \vdash O \,:\, K$ as morphisms $\llbracket O\rrbracket : \llbracket\Gamma\rrbracket \to \llbracket K\rrbracket$. Since the classes are now dependent, and the object is situated in some context, we should upgrade this to a morphism in the slice $\mathcal{C}/\llbracket\Gamma\rrbracket$.

$$\Gamma \vdash O \,:\, K$$

$$\begin{tikzcd} \llbracket\Gamma\rrbracket \arrow[rr, "\llbracket\Gamma\vdash O\,:\,K\rrbracket"] \arrow[dr, "\mathbb{1}_{\llbracket\Gamma\rrbracket}"'] & & \llbracket\Gamma, X : K\rrbracket \arrow[dl, "\llbracket\Gamma\vdash K\ \mathsf{cls}\rrbracket"] \\ & \llbracket\Gamma\rrbracket & \end{tikzcd}$$

We can call $\llbracket\Gamma \vdash O \,:\, K\rrbracket$ a *section* of the *bundle* $\llbracket\Gamma \vdash K\ \mathsf{cls}\rrbracket$. This is a bundle in the sense that for any collection of objects $O_1, \cdots, O_n$ in the classes from Γ, there is a closed class

$$\bullet \vdash [O_n/X_n]\cdots[O_1/X_1]K\ \mathsf{cls}$$

which is the *fiber* over that collection of objects. A section of the bundle then picks out for any such collection of objects, an object in the *fiber*

$$\bullet \vdash [O_n/X_n]\cdots[O_1/X_1]O \,:\, [O_n/X_n]\cdots[O_1/X_1]K$$

$$\frac{\Gamma \vdash \mathsf{ctx} \quad X : K \in \Gamma}{\Gamma \vdash X \,:\, K}\ \textsc{var-obj}$$

$$\begin{tikzcd} \llbracket\Gamma\rrbracket \arrow[rr, "\llbracket\Gamma\vdash X\,:\,K\rrbracket"] \arrow[dr, "\mathbb{1}_{\llbracket\Gamma\rrbracket}"'] & & \llbracket\Gamma, Y : K\rrbracket \arrow[dl, "\llbracket\Gamma\vdash K\ \mathsf{cls}\rrbracket"] \\ & \llbracket\Gamma\rrbracket & \end{tikzcd}$$

We may also write $\llbracket K\rrbracket$ to mean $\llbracket\Gamma \vdash K\ \mathsf{cls}\rrbracket$ for brevity.

4.4 Substitution action on classes and objects

Context morphisms (aka substitutions) are dependent tuples of objects, just as contexts are dependent tuples of classes. They can be composed, which we can view as sequential substitution. They have an action on dependent classes, which we will show corresponds to pullback of bundles. They have an action on objects (more generally action on other substitutions), which we will show corresponds to pullback of sections (more generally existence of all pullbacks).

First let us suppose $\Gamma \vdash \sigma : \Delta$ is a substitution and $\Delta \vdash K$ cls. Then we can form $\Gamma \vdash \sigma K$ cls, where

$$\sigma K := [\sigma_n/X_n] \cdots [\sigma_1/X_1]K \quad \text{and} \quad \Delta = X_1 : K_1, \cdots, X_n : K_n$$

This automatically gives us a substitution $\Gamma, X : \sigma K \vdash \sigma :: X : \Delta, X : K$. This says we have a pullback

$$\frac{\Gamma \vdash \sigma : \Delta \qquad \Delta \vdash K \text{ cls}}{\Gamma \vdash \sigma K \text{ cls}}$$

$$\begin{tikzcd}
[\![\Gamma, X : \sigma K]\!] \arrow[r, dashed, "{[\![\sigma::X]\!]}"] \arrow[d, dashed, "{[\![\sigma K]\!]}"'] \arrow[dr, phantom, "\lrcorner", very near start] & [\![\Delta, X : K]\!] \arrow[d, "{[\![K]\!]}"] \\
[\![\Gamma]\!] \arrow[r, "{[\![\sigma]\!]}"'] & [\![\Delta]\!]
\end{tikzcd}$$

To check the diagram commutes, we need to check judgmental equality component-wise (opening up the equivalence class definition). For the universal property we need to apply (syntactic) substitution on objects.

Now suppose $\Delta \vdash O : K$. Then we can form $\Gamma \vdash \sigma O : \sigma K$, where

$$\sigma O := [\sigma_n/X_n] \cdots [\sigma_1/X_1]O$$

This says our pullback diagram extends

$$\frac{\Gamma \vdash \sigma : \Delta \qquad \Delta \vdash O : K}{\Gamma \vdash \sigma O : \sigma K}$$

$$\begin{tikzcd}
[\![\Gamma]\!] \arrow[r, "{[\![\sigma]\!]}"] \arrow[d, dashed, "{[\![\sigma O]\!]}"'] \arrow[dr, phantom, "\lrcorner", very near start] & [\![\Delta]\!] \arrow[d, "{[\![O]\!]}"] \\
[\![\Gamma, X : \sigma K]\!] \arrow[r, "{[\![\sigma::X]\!]}"] \arrow[d, "{[\![\sigma K]\!]}"'] \arrow[dr, phantom, "\lrcorner", very near start] & [\![\Delta, X : K]\!] \arrow[d, "{[\![K]\!]}"] \\
[\![\Gamma]\!] \arrow[r, "{[\![\sigma]\!]}"'] & [\![\Delta]\!]
\end{tikzcd}$$

Now we can use substitution on classes and objects to define composition of context morphisms. Given two substitutions $\Gamma_0 \vdash \sigma : \Gamma_1$ and $\Gamma_1 \vdash \rho : \Gamma_2$ we define the composition $\Gamma_0 \vdash \sigma \gg \rho : \Gamma_2$ by giving

$$\begin{aligned}
&\Gamma_0 \vdash \sigma\rho_1 : \sigma K_1 \\
&\Gamma_0 \vdash \sigma\rho_2 : \sigma[\rho_1/X_1]K_2 \\
&\vdots \\
&\Gamma_0 \vdash \sigma\rho_n : \sigma[\rho_{n-1}/X_{n-1}] \cdots [\rho_1/X_1]K_n
\end{aligned} \qquad \text{where } \Gamma_2 = X_1 : K_1, \cdots, X_n : K_n$$

So we have

$$\Gamma_0 \vdash \sigma \gg \rho : \Gamma_2 \qquad\qquad [\![\rho]\!] \circ [\![\sigma]\!] : [\![\Gamma_0]\!] \to [\![\Gamma_2]\!]$$

There is a techincal issue here about substitution not working out in a general LCCC, because of pullback only being unique up to isomorphism. However, in the syntax category we have "strict pullbacks".

4.5 Strict pullbacks

The previous construction of the pullback $[\![K]\!] \mapsto [\![\sigma]\!] \mapsto ([\![\sigma K]\!], [\![\sigma :: X]\!])$ defines a strictly equal diagram.

$$[\![\sigma]\!]^* [\![\rho]\!]^* [\![K]\!] = [\![\sigma(\rho K)]\!] = [\![(\sigma \gg \rho) K]\!] = ([\![\rho]\!] \circ [\![\sigma]\!])^* \, [\![K]\!]$$

$$\begin{tikzcd}
\Gamma_0, (\sigma \gg \rho) K \arrow[r, dashed, "="] \arrow[d, "(\sigma \gg \rho) K"'] \arrow[dr, phantom, "\lrcorner\quad ="] & \Gamma_0, \sigma\rho K \arrow[r] \arrow[d, "\sigma\rho K"] \arrow[dr, phantom, "\lrcorner"] & \Gamma_1, \rho K \arrow[r] \arrow[d, "\rho K"] \arrow[dr, phantom, "\lrcorner"] & \Gamma_2, K \arrow[d, "K"] \\
\Gamma_0 \arrow[r, "="] & \Gamma_0 \arrow[r, "\sigma"'] & \Gamma_1 \arrow[r, "\rho"'] & \Gamma_2
\end{tikzcd}$$

Note that in general categories (including LCCCs) the pullback along a composition is only necessarily isomorphic to double pullback.

4.6 Dependent function classes

In the $\lambda^{\times\to}$ case, λ-abstraction and function application provide the adjunction isomorphism in the semantics

$$\mathcal{C}([\![\Gamma \times K_0]\!], [\![K_1]\!]) \underset{\mathsf{ap}}{\overset{\lambda}{\rightleftarrows}} \mathcal{C}([\![\Gamma]\!], [\![K_0 \to K_1]\!])$$

Let us try to consider what happens when function classes are replaced with dependent function classes. Instead of morphisms in the ambient category, we need morphisms in slices, since we are concerned with objects $\Gamma \vdash O \,:\, \{X \,:\, S\}\, K$.

$$\mathcal{C}/?\,(\,?\,, [\![K]\!]) \underset{\text{PI-APP-OBJ}}{\overset{\text{PI-LAM-OBJ}}{\rightleftarrows}} \mathcal{C}/[\![\Gamma]\!](\mathbb{1}_{[\![\Gamma]\!]}, [\![\{X \,:\, S\}\, K]\!])$$

The introduction rule PI-LAM-OBJ fill in the missing details. We only use a special case of PI-APP-OBJ here.

$$\dfrac{\Gamma \vdash S \,:\, \mathsf{Sort} \quad \Gamma, X : S \vdash O \,:\, K}{\Gamma \vdash [X \,:\, S]\, O \,:\, \{X \,:\, S\}\, K}\ \text{PI-LAM-OBJ} \qquad \dfrac{\Gamma, X : S \vdash O \,:\, \{X \,:\, S\}\, K \quad \Gamma, X : S \vdash X \,:\, S}{\Gamma, X : S \vdash O\, X \,:\, K}\ \text{PI-APP-OBJ}$$

$$\mathcal{C}/[\![\Gamma, X : S]\!]\,(\mathbb{1}_{\Gamma, X:S}, [\![K]\!]) \underset{\text{PI-APP-OBJ}}{\overset{\text{PI-LAM-OBJ}}{\rightleftarrows}} \mathcal{C}/[\![\Gamma]\!](\mathbb{1}_{[\![\Gamma]\!]}, [\![\{X \,:\, S\}\, K]\!])$$

The β and η rules for PI state that the maps back and forth form a bijection. It is not yet clear what adjunction the above is a special case of.

Following the moves from before and our categorical intuition, we first replace $\mathbb{1}_{[\![\Gamma]\!]}$ with a more general $\mathcal{C}/[\![\Gamma]\!]$-object, a context morphism $[\![E \vdash \varepsilon \,:\, \Gamma]\!]$. The object is then replaced with a morphism in the slice $[\![\varepsilon :: f]\!] \in \mathcal{C}/[\![\Gamma]\!]([\![\varepsilon]\!], [\![\{X \,:\, S\}\, K]\!])$.

$$\begin{tikzcd}
{[\![E]\!]} \arrow[rr, "{[\![\varepsilon::f]\!]}"] \arrow[dr, "{[\![\varepsilon]\!]}"'] & & {[\![\Gamma, \{X \,:\, S\}\, K]\!]} \arrow[dl, "{[\![\{X:S\}\, K]\!]}"] \\
& {[\![\Gamma]\!]} &
\end{tikzcd}$$

This in particular gives some object

$$E \vdash f \,:\, \{X \,:\, \varepsilon S\}\ ((\varepsilon :: X)K)$$

Weakening and applying PI-APP-OBJ we obtain

$$E, X : \varepsilon S \vdash f X \,:\, (\varepsilon :: X)K$$

$$\begin{tikzcd}
{[\![E, X : \varepsilon S, Y : (\varepsilon :: X)K]\!]} \arrow[r, "{[\![(\varepsilon::X)K]\!]}"] \arrow[d, "{[\![\varepsilon::X::Y]\!]}"'] \arrow[dr, phantom, "\lrcorner", very near start] & {[\![E, X : \varepsilon S]\!]} \arrow[r, "{[\![\varepsilon S]\!]}"] \arrow[d, dashed, "{[\![\varepsilon::X]\!]}"'] \arrow[l, bend right, "{[\![f X]\!]}"'] \arrow[dl, dashed] \arrow[dr, phantom, "\lrcorner", very near start] & {[\![E]\!]} \arrow[d, "{[\![\varepsilon]\!]}"] \\
{[\![\Gamma, X : S, Y : K]\!]} \arrow[r, "{[\![K]\!]}"'] & {[\![\Gamma, X : S]\!]} \arrow[r, "{[\![S]\!]}"'] & {[\![\Gamma]\!]}
\end{tikzcd}$$

The dashed arrow is then a morphism in the slice $\mathcal{C}/[\![\Gamma, X \,:\, S]\!]$. So we have (making the same argument the other way round and using β and η)

$$\mathcal{C}/[\![\Gamma, X \,:\, S]\!]\,([\![\varepsilon :: X]\!], [\![K]\!]) \underset{\text{PI-APP-OBJ}}{\overset{\text{PI-LAM-OBJ}}{\rightleftarrows}} \mathcal{C}/[\![\Gamma]\!]([\![\varepsilon]\!], [\![\{X \,:\, S\}\, K]\!])$$

This is closer to an adjunction: we can see that the left adjoint should be pullback along $[\![S]\!] : [\![\Gamma, X \,:\, S]\!] \to [\![\Gamma]\!]$.

$$[\![S]\!]^* [\![\varepsilon]\!] = [\![\varepsilon :: X]\!]$$

Now we make the right adjoint functorial as well. We replace $[\![K]\!]$ with a slice $[\![\delta]\!] : [\![\Delta]\!] \to [\![\Gamma, X \,:\, S]\!]$.

$$\mathcal{C}/[\![\Gamma, X \,:\, S]\!]\,([\![S]\!]^*[\![\varepsilon]\!], [\![\delta]\!]) \underset{\text{PI-APP-OBJ}}{\overset{\text{PI-LAM-OBJ}}{\rightleftarrows}} \mathcal{C}/[\![\Gamma]\!]([\![\varepsilon]\!], \Pi_{[\![S]\!]}[\![\delta]\!])$$

Briefly and roughly: Write context $\Delta = V_1 \,:\, \Delta_1, \cdots, V_n \,:\, \Delta_n$. A slice morphism O on the left consists of objects $E, X : \varepsilon S \vdash O_i \,:\, [O/V]\Delta_i$. Then λ-abstraction gives objects $E \vdash [X \,:\, \varepsilon S]\, O_i \,:\, \{X \,:\, \varepsilon S\}\, [O/V]\Delta_i$ on the right hand side. So we should define $\Pi_{[\![\delta]\!]}[\![\zeta]\!]$ such that its $\mathcal{C}$-object part is context

$$[\![V_1 : \{X \,:\, \varepsilon S\}\, \Delta_0, \cdots, V_n : \{X \,:\, \varepsilon S\}\, \Delta_n]\!]$$

and its morphism part comes from each Z_i being a class in context $\Gamma, X \,:\, S$. The proof that it is an adjunction is in the same vein as before.

$$[\![S]\!]^* \dashv \Pi_{[\![S]\!]}$$

4.7 Sorts (pre-equality)

The category of contexts is *not* a locally cartesian closed category. In particular, we don't have pullbacks for arbitrary morphisms in the category; the closest we got was over a context with only sorts. However, we can find a subcategory that *is* locally cartesian closed, which essentially consists of context extentions, where the extensions only introduce sorts. This subcategory will be an LCCC-universe, which we specify by giving a set of $\mathcal{C}$-morphisms $\mathcal{S}_\Pi$.

Note that although this subcategory of the category of contexts is an LCCC, it is not the case that locally cartesian closed categories are fit for semantic purposes due to the substitution/pullback issue addressed later.

We know that for contexts extended with sorts $[\![S]\!] : [\![\Gamma, X : S]\!] \to [\![\Gamma]\!]$ we have the adjunction $[\![S]\!]^* \dashv \Pi_{[\![S]\!]}$. In particular $[\![\mathsf{El}]\!] : [\![\mathsf{El} : \mathsf{Sort}, X : \mathsf{El}]\!] \to [\![\mathsf{El} : \mathsf{Sort}]\!]$ is such an extension, and there is a correspondence between pullbacks of $[\![\mathsf{El}]\!]$ and sort-judgments

$$\Gamma \vdash S \,:\, \mathsf{Sort} \qquad\qquad \begin{tikzcd} {[\![\Gamma, S]\!]} \arrow[r] \arrow[d, "{[\![S]\!]}"'] \arrow[dr, phantom, "\lrcorner", very near start] & {[\![\mathsf{El} : \mathsf{Sort}, X : \mathsf{El}]\!]} \arrow[d, "{[\![\mathsf{El}]\!]}"] \\ {[\![\Gamma]\!]} \arrow[r] & {[\![\mathsf{El} : \mathsf{Sort}]\!]} \end{tikzcd}$$

So, inspired by the definition of LCCC-universes, we inductively generate $\mathcal{S}_\Pi$ by requiring $[\![\mathsf{El}]\!] \in \mathcal{S}_\Pi$, that $\mathcal{S}_\Pi$ contains identities, taking the closure under pullback against $\mathcal{C}$-morphisms, closure under Πs and Σs (composition).

An equivalent characterization of $\mathcal{S}_\Pi$ is the set of compositions of sort-context-exentions, i.e. $[\![S_1]\!] \circ \cdots \circ [\![S_n]\!]$ (the identity on $[\![\Gamma]\!]$ when $n = 0$) whenever

$$\Gamma \vdash \mathsf{ctx} \qquad \Gamma \vdash S_1 \,:\, \mathsf{Sort} \qquad \Gamma, X_1 : S_1 \vdash S_2 \,:\, \mathsf{Sort} \qquad \cdots \qquad \Gamma, X_1 : S_1, \cdots, X_{n-1} : S_{n-1} \vdash S_n \,:\, \mathsf{Sort}$$

To ensure the two characterizations are equivalent it suffices to check that the second is stable under pullbacks, and closed under Πs.

To check stability under pullback along $[\![\sigma]\!] : [\![\Delta]\!] \to [\![\Gamma]\!]$, first recall that for any sort $[\![S]\!] : [\![\Gamma, X : S]\!] \to [\![\Gamma]\!]$, the pullback is the substition on the sort $[\![\sigma^* S]\!] : [\![\Delta, X : \sigma^* S]\!] \to [\![\Delta]\!]$ which is another sort. This means that the pullback of any composition $[\![S_1]\!] \circ \cdots \circ [\![S_n]\!]$ is just the composition of the pullbacks ("pasting"), and given that each piece of the pullback is a sort, the composition is in $\mathcal{S}_\Pi$. For the case when $n = 0$ note that the pullback of the identity is the identity.

Closure under Πs: by stability under pullback along general context substitutions, we know that for any morphism $g = [\![S_1]\!] \circ \cdots \circ [\![S_n]\!] \in \mathcal{S}_\Pi$, g^* is a well-defined functor. Furthermore, for each $[\![S_i]\!]$ we have an adjunction $[\![S_i]\!]^* \dashv \Pi_{[\![S_i]\!]}$, and the composition of adjunctions yields an adjunction

$$[\![S_n]\!]^* \circ \cdots \circ [\![S_1]\!]^* = [\![g]\!]^* \dashv \Pi_{[\![g]\!]} = \Pi_{[\![S_1]\!]} \circ \cdots \circ \Pi_{[\![S_n]\!]}$$

Thus the two charactizations of $\mathcal{S}_\Pi$ are equivalent.

Remark. In the original definition of a universe in a topos [SD04] the property 2 is the inclusion of all monos from $\mathcal{C}$ (in particular identity morphisms). Including all monos corresponds to including all proposition-context-extensions, i.e. $\Gamma, X : P$ where P is a proposition.

The original 5 said that all morphisms of $\mathcal{S}_\Pi$ are pullbacks of a single morphism $\mathsf{El} : E \to U$. Our most obvious candidate for such a map is the first context extension $[\![\mathsf{El}_1]\!] : [\![\mathsf{El}_1 : \mathsf{Sort}, X_1 : \mathsf{El}_1]\!] \to [\![\mathsf{El}_1 : \mathsf{Sort}]\!]$. This would imply in particular that composition and identity are also certain pullbacks of (El). In the case of composition, this amounts to requiring dependent pair classes (aka Σ-types) in the type theory, which we do not have. In the case of identity, this amounts to requiring that every sort is inhabited with an object, which is not something we want in our LF.

4.8 Equality classes

We showed that pullbacks of context morphisms along context extensions/classes corresponds to substitution on classes. Do we have pullback along any context morphism? The answer is *yes*: in order to construct the pullback, in the presence of an equality class.

$$\begin{tikzcd} [\![\Delta]\!] \times_{[\![\Gamma]\!]} [\![E]\!] \arrow[r, "\pi_E"] \arrow[d, "\pi_\Delta"'] \arrow[dr, phantom, "\lrcorner", very near start] & [\![E]\!] \arrow[d, "[\![\varepsilon]\!]"] \\ [\![\Delta]\!] \arrow[r, "[\![\delta]\!]"'] & [\![\Gamma]\!] \end{tikzcd}$$

Again imagining Δ and E as closed classes, the pullback should be "pairs of objects in $\Delta \times E$" that are equal upon their substitutions into the objects given by δ and ε. In set-theoretic notation (since we are inspired by pullback in **Set**)

$$\{(X, Y) \in \Delta \times E \mid \delta(X) = \varepsilon(Y)\}$$

Translating this idea into a context means that we need

$$\begin{aligned} [\![\Delta]\!] \times_{[\![\Gamma]\!]} [\![E]\!] &= [\![X : \Delta, Y : E, H : \delta =_\Gamma \varepsilon]\!] \\ \pi_\Delta &= [\![X]\!] \\ \pi_E &= [\![Y]\!] \end{aligned}$$

In detail: if $\Delta = X_1 : \Delta_1, \cdots, X_n : \Delta_n$ and $E = Y_1 : E_1, \cdots, Y_m : E_m$ and $\Gamma = Z_1 : \Gamma_1, \cdots, Z_l : \Gamma_l$ then the context is

$$X_1 : \Delta_1, \cdots, X_n : \Delta_n, Y_1 : E_1, \cdots, Y_m : E_m, H_1 : \delta_1 =_{\Gamma_1} \varepsilon_1, \cdots, H_k : \delta_l =_{\Gamma_l} \varepsilon_l$$

and the context morphisms are those that return the relevant variables from this context.

To show the universal property, given cone $[\![Z]\!]$ we use that judgmental equalities $Z \vdash \zeta \gg \delta = \zeta \gg \varepsilon : \Gamma$ lift to $Z \vdash \mathsf{self} : \zeta \gg \delta =_\Gamma \zeta \gg \varepsilon$ for existence (this fact follows from judgmental equality being a congruence); we use UNICITY for uniqueness.

$$\frac{\Gamma \vdash O_0 : K \quad \Gamma \vdash O_1 : K}{\Gamma \vdash O_0 =_S O_1 \ \mathsf{cls}} \ \text{EQ-CLS} \qquad\qquad [\![\Delta]\!] \times_{[\![\Gamma]\!]} [\![E]\!]$$

4.9 Sorts (post-equality)

We want the corresponding semantics for the rule EQ-SORT. This should say that the LCCC-universe is closed under equality. The situation is similar to EQ-CLS, except our assumption should be that we have

$$\begin{tikzcd} [\![Y : E, X : \Delta, H : \delta =_\Gamma \varepsilon]\!] \arrow[r, "\pi_E"] \arrow[d, "\pi_\Delta"'] \arrow[dr, phantom, "\lrcorner", very near start] & [\![E]\!] \arrow[d, "[\![\varepsilon]\!]"] \\ [\![\Delta]\!] \arrow[r, "[\![\delta]\!]"'] & [\![\Gamma]\!] \end{tikzcd}$$

when Γ only contains sorts. This guarantees that each equality class formed $\delta_i =_{\Gamma_i} \varepsilon_i$ is again a sort.

Now, ensuring Γ is sort-only, we naively try to add $[\![\delta]\!] : [\![\Delta]\!] \to [\![\Gamma]\!]$ into the LCCC-universe $\mathcal{S}_\Pi$, and inductively generate another LCCC-universe. The new LCCC-universe is stable under pullbacks from $\mathcal{C}$, so $\pi_E = [\![\varepsilon]\!]^*[\![\delta]\!]$ should be in it. Note that neither is $[\![E]\!]$ sort-only, nor is π_E in $\mathcal{S}_\Pi$, so this LCCC-universe becomes quickly unwieldly. However, if we require that *both* Δ *and* Γ are sort only, then π_E is a sort-context-extension, so it is in $\mathcal{S}_\Pi$ already.

So let us characterize our candidate new LCCC-universe as

$$\mathcal{S}_{\Pi,=} := \mathcal{S}_\Pi \cup \{ [\![\Delta \vdash \delta \;:\; \Gamma]\!] \mid \Delta, \Gamma \text{ sort-only } \}$$

$\mathcal{S}_{\Pi,=}$ is stable under pullback, as noted above. It is clearly closed under identity. To show it is closed under composition, we case on which part of the union the parts came from; each case is easy. To show it is closed under Πs, we just need to generalize the adjunction from before:

$$\mathcal{C}/[\![\Delta]\!]\left([\![\delta]\!]^*[\![\varepsilon]\!], [\![\zeta]\!]\right) \underset{\text{PI-APP-OBJ}}{\overset{\text{PI-LAM-OBJ}}{\rightleftarrows}} \mathcal{C}/[\![\Gamma]\!]\left([\![\varepsilon]\!], \Pi_{[\![\delta]\!]}[\![\zeta]\!]\right)$$

Briefly and roughly, if $[\![\zeta]\!] : [\![Z]\!] \to [\![\Delta]\!]$ for $Z = V_1 : Z_1, \cdots, V_n : Z_n$ then the $\mathcal{C}$-object part of $\Pi_{[\![\delta]\!]}[\![\zeta]\!]$ is

$$[\![V_1 : \{X \;:\; \Delta\}\, \{Y \;:\; \varepsilon =_\Gamma \delta\}\, Z_1, \cdots, V_n : \{X \;:\; \Delta\}\, \{Y \;:\; \varepsilon =_\Gamma \delta\}\, Z_n]\!]$$

This can be formed since Δ is sort-only and Γ is sort only (hence the equality classes are sorts). Furthermore, if $[\![\zeta]\!] \in \mathcal{S}_{\Pi,=}$ then either it is a sort-context-extension of sort-only Δ, hence Z is sort-only, or it Z is sort-only by assumption. Thus $\Pi_\delta[\![\zeta]\!]$ would be sort-only by PI-SORT. So $\mathcal{S}_{\Pi,=}$ is closed under Πs. This concludes that $\mathcal{S}_{\Pi,=}$ is an LCCC-universe.

4.10 Signatures

Given a signature Σ, we can simply take the slice over $[\![\Sigma]\!]$ in the semantics to obtain semantics for the type theory generated by Σ. We know that a slice of a locally closed cartesian category is locally cartesian closed, hence $\mathcal{S}_{\Pi,=}/[\![\Sigma]\!]$ is still an LCCC.

4.11 Definitional equality

Strictly speaking, we should have taken contexts *up to definitional equality* as $\mathcal{C}$-objects by treating definitional equality of classes as equivalence relations. Similarly we should have taken substitutions *up to definitional equality* as $\mathcal{C}$-morphisms by treating definitional equality of objects as equivalence relations. This does not have a significant impact on the arguments we have made.

5 Extensions

Let us return to studying $\mathcal{S}_\Pi$ and $\mathcal{S}_{\Pi,=}$.

References

[Awo17] Steve Awodey. Natural models of homotopy type theory, 2017.

[CCD20] Simon Castellan, Pierre Clairambault, and Peter Dybjer. Categories with families: Unityped, simply typed, and dependently typed, 2020.

[Har21] Robert Harper. An equational logical framework for type theories, 2021.

[Jac93] Bart Jacobs. Comprehension categories and the semantics of type dependency. *Theoretical Computer Science*, 107(2):169–207, 1993.

[SD04] Thomas Streicher and Tu Darmstadt. Universes in toposes. 05 2004.

[SH21] Jonathan Sterling and Robert Harper. Logical relations as types: Proof-relevant parametricity for program modules. *Journal of the ACM*, 68(6):1–47, oct 2021.