Equational Logical Frameworks as Locally Cartesian Closed Categories

Joseph Hua

April 16, 2023

Contents

1 Overview

- Introduce LF with equality. Closely follow [Har21]
- Introduce the categorical way of looking at the type formers (motivational)
- Construct category of contexts formally
- General model of LF in an LCCC
- Lawvere style correspondence

In the first section, we present an *equational logical framework* as in [Har21], which can be used to define dependent type theories such as Martin Löf type theory or ModTT [SH21].

1.1 Notation

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

In this section we recall the equational logical framework, as described in [Har21] (with some minor changes). We want this to be minimalistic but sufficiently expressive so that we can define type theories by asserting the rules the type theory via a collection of generators, called a signature and denoted Σ. We state the syntax and judgment rules for convenience and refer to [Har21] for the details.

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

O indicates this is meant to represent an object (though they are all expressions at this point), similarly S for sorts and K for classes. The agnostic expression symbol e should represent a sort or a class.

The judgments we define are

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

The legal judgments given by the closure under the judgment rules in figures 1, 2, 3, 4, and those for judgmental equality (not included).

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

Figure 1: Context formation

Note that unlike in [Har21] we have split the equality formation into two rules: EQ-CLS and EQ-SORT, since this will give us cleaner semantics.

Judgmental equality for classes $\Gamma \vdash K_0 = K_1$ 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. See [Har21] for details.

Finally, signatures are defined as contexts $\Sigma \vdash$ 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

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

$$\frac{\Gamma \vdash O_0\ :\ K \quad \Gamma \vdash O_1\ :\ K}{\Gamma \vdash O_0 =_S O_1\ \mathsf{cls}}\ \textsc{eq-cls}$$

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

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.

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

3 Categorical definitions

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

- $\mathcal{C}$ has finite products
- 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 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.

The following definition is an adaptation of *universes in a topos* from [SD04].

$$\frac{\Gamma \vdash \mathsf{ctx} \quad X : K \in \Gamma}{\Gamma \vdash X \,:\, K}\ \text{VAR-OBJ} \qquad \frac{\Gamma \vdash S \,:\, \mathsf{Sort} \quad \Gamma, X : S \vdash O \,:\, K}{\Gamma \vdash [X \,:\, S]\, O \,:\, \{X \,:\, S\}\, K}\ \text{PI-LAM-OBJ}$$

$$\frac{\Gamma \vdash O_0 \,:\, \{X \,:\, S\}\, K \quad \Gamma \vdash O_1 \,:\, S}{\Gamma \vdash O_0\, O_1 \,:\, [O_1/X]K}\ \text{PI-APP-OBJ} \qquad \frac{\Gamma \vdash O \,:\, K}{\Gamma \vdash \mathsf{self} \,:\, O =_K O}\ \text{EQ-SELF-OBJ}$$

$$\frac{\Gamma \vdash O \,:\, K_0 \quad \Gamma \vdash K_0 = K_1\ \mathsf{cls}}{\Gamma \vdash O \,:\, K_1}\ \text{ID-OBJ}$$

Figure 4: Object formation

Definition – LCCC-universe

Let $\mathcal{C}$ be a category, and $\mathcal{S}$ a set of morphisms in $\mathcal{C}$. We may denote morphisms in $\mathcal{S}$ by $S : \Gamma, S \to \Gamma$ or $\sigma : \Delta \to \Gamma$, though this is just notation. We say $\mathcal{S}$ is an LCCC-universe when

1. $\mathcal{S}$ is stable under $\mathcal{C}$-pullbacks, i.e. if $\delta : \Delta \to \Gamma$ is in $\mathcal{C}$ and $S : \Gamma, S \to \Gamma$ is in $\mathcal{S}$ and we have pullback

$$\begin{tikzcd} \Delta, \delta^* S \arrow[r] \arrow[d, "\delta^* S"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Gamma, S \arrow[d, "S"] \\ \Delta \arrow[r, "\delta"'] & \Gamma \end{tikzcd} \qquad \text{then} \qquad \delta^* S \in \mathcal{S}$$

2. $\mathcal{S}$ is closed under identities, i.e. if $\sigma : \Delta \to \Gamma$ is in $\mathcal{S}$ then so are $\mathbb{1}_\Delta$ and $\mathbb{1}_\Gamma$.
3. $\mathcal{S}$ is closed under composition (aka Σs), i.e. if

$$\Gamma, S, T \xrightarrow[\in \mathcal{S}]{T} \Gamma, S \xrightarrow[\in \mathcal{S}]{S} \Gamma \qquad \text{then} \qquad \Sigma_S T := S \circ T \in \mathcal{S}$$

4. $\mathcal{S}$ is closed under Πs, i.e. if $S : \Gamma, S \to \Gamma$ is in $\mathcal{S}$ then pullback $S^* : \mathcal{C}/\Gamma \to \mathcal{C}/\Gamma, S$ exists and has right adjoint $\Pi_S : \mathcal{C}/\Gamma, S \to \mathcal{C}/\Gamma$, and

$$\Gamma, S, T \xrightarrow[\in \mathcal{S}]{T} \Gamma, S \xrightarrow[\in \mathcal{S}]{S} \Gamma \qquad \text{then} \qquad \Pi_S T \in \mathcal{S}$$

Note that the first four conditions combined result in $\mathcal{S}$ forming a subcategory of $\mathcal{C}$ that is an LCCC. Note that 2 is different from and an extra condition 5 is given in the definition of *universe in a topos* given in [SD04], We address this later.

4 The syntactic category

We want to find categorical semantics for our type theory. This amounts to one of two equivalent things

- View each aspect of the type theory in a category $\mathcal{C}$
- Construct a syntactic category which will be the "category of contexts" and take semantics to be anything that imitates it (in a Platonic form sense).

We will take the latter approach and construct the category of contexts. We will call this category $\mathcal{C}$ for the rest of this section.

We begin this section by recalling $\lambda^{\times\to}$, a 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 $\lambda^{\times\to}$ 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* as $\mathcal{C}$-objects, and *as a consequence* dependent classes as $\mathcal{C}$-objects.

$$\Gamma \vdash \mathsf{ctx} \qquad\qquad [\![\Gamma]\!] \in \mathcal{C}$$

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

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

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

$$\Gamma \vdash \sigma \ : \ \Delta \qquad\qquad [\![\sigma]\!] : [\![\Gamma]\!] \to [\![\Delta]\!]$$

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

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

For context extension, given $\Gamma \vdash K$ 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 $\mathcal{C}/[\![\Gamma]\!]$-object, where $\mathcal{C}/[\![\Gamma]\!]$ is the slice category.

$$[\![\Gamma \vdash K \ \mathsf{cls}]\!] := [\![\Gamma, X : K]\!] \text{ and } [\![\Gamma, X : K \vdash \sigma \ : \ \Gamma]\!]$$

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

$$\frac{\Gamma \vdash K\ \mathsf{cls}}{\Gamma, X : K \vdash \mathsf{ctx}}\ \textsc{ctx-ext} \qquad\qquad [\![\Gamma, X : K]\!] \xrightarrow{[\![\Gamma \vdash K\ \mathsf{cls}]\!]} [\![\Gamma]\!]$$

4.2 Objects

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

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

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

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

A section of the bundle then picks out for any such collection of objects, an object

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

$$\frac{\Gamma \vdash \mathsf{ctx} \quad X : K \in \Gamma}{\Gamma \vdash X : K}\ \textsc{var-obj} \qquad\qquad \begin{tikzcd} {[\![\Gamma]\!]} \arrow[rr, "{[\![\Gamma \vdash X : K]\!]}"] \arrow[dr, "\mathbb{1}_{[\![\Gamma]\!]}"'] & & {[\![\Gamma, Y : K]\!]} \arrow[dl, "{[\![\Gamma \vdash K\ \mathsf{cls}]\!]}"] \\ & {[\![\Gamma]\!]} & \end{tikzcd}$$

We may also write $[\![K]\!]$ to mean $[\![\Gamma \vdash K\ \mathsf{cls}]\!]$ for brevity.

4.3 Substitution

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 can induce substitutions on dependent classes, which we will show corresponds to pullback of bundles. They can induce substitution on objects (more generally substitution on 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 $\Delta \vdash K\ \mathsf{cls}$. Then we can form $\Gamma \vdash \sigma K\ \mathsf{cls}$, 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

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

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

To check the diagram commutes, we use a notion of judgmental equality between substitutions, which we can define component-wise by judgmental equality between objects. 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

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

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

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

$$\Gamma_0 \vdash \sigma \gg \rho \,:\, \Gamma_2 \qquad\qquad \llbracket \rho \rrbracket \circ \llbracket \sigma \rrbracket : \llbracket \Gamma_0 \rrbracket \to \llbracket \Gamma_2 \rrbracket$$

There is a techincal issue here about substitution not working out in a general LCCC, because of pullback only being unique up to isomorphism. We delay this issue until later, and assume strict pullbacks, i.e. all pullbacks in $\mathcal{C}$ of the same diagram are equal.

4.4 Dependent function classes

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

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

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 $\Gamma \vdash O : \{X : S\}\, K$.

$$\mathcal{C}/?\,(\,?\,, \llbracket K \rrbracket) \underset{\textsc{pi-app-obj}}{\overset{\textsc{pi-lam-obj}}{\rightleftarrows}} \mathcal{C}/\llbracket \Gamma \rrbracket(\mathbb{1}_{\llbracket \Gamma \rrbracket}, \llbracket \{X : S\}\, K \rrbracket)$$

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

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

$$\mathcal{C}/\llbracket \Gamma, X : S \rrbracket\,(\mathbb{1}_{\Gamma, X : S}, \llbracket K \rrbracket) \underset{\textsc{pi-app-obj}}{\overset{\textsc{pi-lam-obj}}{\rightleftarrows}} \mathcal{C}/\llbracket \Gamma \rrbracket(\mathbb{1}_{\llbracket \Gamma \rrbracket}, \llbracket \{X : S\}\, K \rrbracket)$$

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 $\mathbb{1}_{\llbracket \Gamma \rrbracket}$ with a more general $\mathcal{C}/\llbracket \Gamma \rrbracket$-object, a context morphism $\llbracket E \vdash \varepsilon : \Gamma \rrbracket$. The object is then replaced with a morphism in the slice $\llbracket \varepsilon :: f \rrbracket \in \mathcal{C}/\llbracket \Gamma \rrbracket(\llbracket \varepsilon \rrbracket, \llbracket \{X : S\}\, K \rrbracket)$.

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

This in particular gives some object

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

Weakening and applying PI-APP-OBJ we obtain

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

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

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.

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

4.5 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 syntactic category 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.

4.6 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 $\mathbf{Set}$)

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

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.

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

4.7 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:

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

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.

4.8 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.

4.9 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.

4.10 Strict pullbacks

References

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

[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.