Category with families

Frank Tsai

March 7, 2025

These are my personal notes on and expositions of [Hof97]. Any mistake is my own.

1 Introduction

Semantics is a compositional assignment of mathematical objects to syntactic objects; syntactic objects are *interpreted* as objects in the semantic domain. A reasonable question that one may ask is: why do we care about semantics? Besides mathematical curiosity, semantic methods have been applied to show numerous independence results. On a more practical side, semantic methods can be applied to show syntactic properties such as normalization and decidability of type checking. Indeed, one can argue that the latter property is nonnegotiable in any practical computer implementation of type theory.

Hofmann [Hof97] developed an abstract semantic framework — called *category with families* — upon which a single interpretation function can be defined once and for all. Then, to show that a theory can be interpreted in a given semantic domain, one shows that the given semantic domain fits in such a framework.

2 Substitutions

In dependent type theory, judgments have to be made with respect to a context. For example, the judgment

$$\mathrm{Id}(x, 1)\ \mathrm{type}$$

does not make sense without knowing that $x : \mathbb{N}$.

For us, a context is a finite list of variable declarations. Note that the order of variable declarations matters in dependent type theory since the type of a variable may depend on a variable declared earlier. We use the turnstile notation to denote judgments made relative to a context. The example above can be made under the context $x : \mathbb{N}$, denoted as follows:

$$x : \mathbb{N} \vdash \mathrm{Id}(x, 1)\ \mathrm{type}$$

The most fundamental operation that we can perform on variables is *substitution*. Say, we have a term

$$z : \mathbb{R}^+ \vdash \lfloor z \rfloor : \mathbb{N}$$

We can substitute $\lfloor z \rfloor$ for x in $\mathrm{Id}(x, 1)\ \mathrm{type}$, resulting in $\mathrm{Id}(\lfloor z \rfloor, 1)\ \mathrm{type}$. This judgment has to be made in an updated context.

$$z : \mathbb{R}^+ \vdash \mathrm{Id}(\lfloor z \rfloor, 1)\ \mathrm{type}$$

We may regard the substitution along the term $\lfloor z \rfloor$ as a *morphism* from $z : \mathbb{R}^+$ to $x : \mathbb{N}$.

Definition 1. Let Δ and $\Gamma = x_1 : \sigma_1, \dots, x_n : \sigma_n$ be contexts. A *substitution* from Δ to Γ is a sequence of n terms $(t_1, \dots, t_n)$ such that the following n judgments hold:

$$\begin{aligned}
&\Delta \vdash t_1 : \sigma_1 \\
&\Delta \vdash t_2 : \sigma_2[t_1/x_1] \\
&\quad \dots \\
&\Delta \vdash t_n : \sigma_n[t_1/x_1] \dots [t_{n-1}/x_{n-1}]
\end{aligned}$$

Notation 2. Let $\Gamma = x_1 : \sigma_1, \ldots, x_n : \sigma_n$ be a context and $f = (t_1, \ldots, t_n)$ be a substitution. If $\Gamma \vdash \sigma$ type (respectively $\Gamma \vdash t : \sigma$), then we write $\sigma[f]$ (respectively $t[f]$) for the simultaneous substitution $\sigma[t_1/x_1, \ldots, t_n/x_n]$.

Example 3. For any context Γ, there exists a unique substitution () from Γ to the empty context $\diamond$.

Example 4. If $\Gamma = x_1 : \sigma_1, \ldots, x_n : \sigma_n$ is a context and $\Gamma \vdash \sigma$ type and x is a fresh variable, then $(x_1, \ldots, x_n)$ is a substitution $\Gamma, x : \sigma \vdash \mathsf{p}(\Gamma, \sigma) : \Gamma$.

Example 5. For any context $\Gamma = x_1 : \sigma_1, \ldots, x_n : \sigma_n$ and any term $\Gamma \vdash t : \sigma$ we can form a substitution $\Gamma \vdash (x_1, \ldots, x_n, t) : \Gamma, x : \sigma$. We write $\overline{t} := (x_1, \ldots, x_n, t)$.

Example 6. For any context $\Gamma = x_1 : \sigma_1, \ldots, x_n : \sigma_n$, the identity substitution $\Gamma \vdash \mathrm{id}_\Gamma : \Gamma$ is given by $\mathrm{id}_\Gamma = (x_1, \ldots, x_n)$.

Example 7. Substitutions can be composed in the usual way. Let $\Lambda \vdash g : \Delta$ and $\Delta \vdash f : \Gamma$, where $f = (t_1, \ldots, t_n)$. The composition $f \circ g$ is the tuple $(t_1[g], \ldots, t_n[g])$. It is clear that $\Lambda \vdash f \circ g : \Gamma$.

Example 8. Let $\Delta \vdash f : \Gamma$. There is a substitution $\mathsf{q}(f, \sigma)$ in the following configuration:

$$\begin{array}{ccc}
\Delta, x : \sigma[f] & \xrightarrow{\mathsf{q}(f,\sigma)} & \Gamma, x : \sigma \\
{\scriptstyle \mathsf{p}(\Delta,\sigma[f])}\downarrow & & \downarrow{\scriptstyle \mathsf{p}(\Gamma,\sigma)} \\
\Delta & \xrightarrow[f]{} & \Gamma
\end{array}$$

Explicitly, $\mathsf{q}(f, \sigma)$ is given by (f, x). The diagram commutes, meaning $\mathsf{p}(\Gamma, \sigma) \circ \mathsf{q}(f, \sigma) = f \circ \mathsf{p}(\Delta, \sigma[f])$, up to variable renaming.

Proposition 9. *Assume $\Pi \vdash h : \Lambda$, $\Lambda \vdash g : \Delta$, and $\Delta \vdash f : \Gamma$. Furthermore, let $\Gamma \vdash \sigma$ type and $\Gamma \vdash t : \sigma$; then the following equations hold (up to variable renaming).*

$$\begin{aligned}
f \circ \mathrm{id}_\Delta &= \mathrm{id}_\Gamma \circ f = f \\
(f \circ g) \circ h &= f \circ (g \circ h) \\
\sigma[\mathrm{id}_\Gamma] &= \sigma \\
\sigma[f \circ g] &= \sigma[f][g] \\
t[\mathrm{id}_\Gamma] &= t \\
t[f \circ g] &= t[f][g]
\end{aligned}$$

The first two equations suggest that contexts and substitutions form a category, while the remaining equations suggest that substitution is functorial.

3 Category with families

Building upon Proposition 9, we can consider any small category with a terminal object as a category of contexts and substitutions. The terminal object models the empty context.

Scholium 10. A small category with a terminal object contains the data of contexts and substitutions of type theory.

To model the following two judgments. We need to specify a set of types $\mathrm{Ty}(\Gamma)$ for each context Γ and set of terms $\mathrm{Tm}(\Gamma, \sigma)$ for each type $\sigma \in \mathrm{Ty}(\Gamma)$.

$$\Gamma \vdash \sigma \text{ type} \qquad\qquad \Gamma \vdash t : \sigma$$

Moreover, these families of sets vary with substitutions: given a substitution $\Delta \vdash f : \Gamma$, we must have $\sigma[f] \in \mathrm{Ty}(\Delta)$ and $t[f] \in \mathrm{Tm}(\Delta, \sigma[f])$. This models the following judgments:

$$\Delta \vdash \sigma[f] \text{ type} \qquad\qquad \Delta \vdash t[f] : \sigma[f]$$

Hence if C is a category of contexts and substitutions, the set of types and the set of terms form a C-indexed family. To formalize this idea, we define the category Fam.

Definition 11. The *category* Fam *of families* has, as objects, pairs (B^0, B^1), where B^0 is a set and B^1 is an B^0-indexed family of sets $(B^1_b)_{b \in B^0}$; and, as morphisms, $(B^0, B^1) \to (C^0, C^1)$ pairs (f^0, f^1), where $f^0 : B^0 \to C^0$ is a function and f^1 is an B^0-indexed family of functions $(f^1_b : B^1_b \to C^1_{f^0(b)})_{b \in B^0}$.

Scholium 12. Given a category C of contexts and substitutions. The C-indexed family of types and terms is specified by a contravariant functor $F : C^{\mathrm{op}} \to \mathsf{Fam}$.

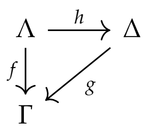
Notation 13. We write $(\mathrm{Ty}, \mathrm{Tm}) : C^{\mathrm{op}} \to \mathsf{Fam}$ to emphasize that for each context Γ, the components of $(\mathrm{Ty}(\Gamma), \mathrm{Tm}(\Gamma))$ are respectively a set of types and an indexed-set of terms. We write $\sigma\{f\}$ and $t\{f\}$ rather than the more cumbersome notations $\mathrm{Ty}(f)(\sigma)$ and $\mathrm{Tm}(f)_{\sigma\{f\}}(t)$.

The last missing ingredient is context extension: given a context Γ and a type $\sigma \in \mathrm{Ty}(\Gamma)$ we need to specify a context $\Gamma.\sigma$ to model the variable declaration $\Gamma, x : \sigma$. We need a substitution $\mathsf{p}(\Gamma, \sigma) : \Gamma.\sigma \to \Gamma$, so we can extend the context Γ with σ. Furthermore, we need a term $v \in \mathrm{Tm}(\Gamma.\sigma, \sigma\{\mathsf{p}(\Gamma, \sigma)\})$ that plays the role of a variable, so we can model the variable rule:

$$\Gamma, x : \sigma \vdash x : \sigma$$

A candidate for context extension is simply a substitution $f : \Delta \to \Gamma$ equipped with a term $t \in \mathrm{Tm}(\Delta, \sigma\{f\})$.

Definition 14. The *category of comprehension candidates* $\mathrm{El}(F_{\Gamma,\sigma})$ has, as objects, pairs (f, s), where $f : \Delta \to \Gamma$ is a substitution and $s \in \mathrm{Tm}(\Delta, \sigma\{f\})$; and, as morphisms, $(f, s) \to (g, t)$ commuting triangles in the following configuration:



such that $t\{h\} = s$.

Among all candidates, we pick the universal one to model context extension.

Definition 15. Let $\sigma \in \mathrm{Ty}(\Gamma)$. A *comprehension for* σ is a choice of a terminal object in $\mathrm{El}(F_{\Gamma,\sigma})$. Explicitly, this consists of a substitution $\mathsf{p}(\Gamma, \sigma) : \Gamma.\sigma \to \Gamma$ and a term $v \in \mathrm{Tm}(\Gamma.\sigma, \sigma\{\mathsf{p}(\Gamma, \sigma)\})$ such that for any substitution $f : \Delta \to \Gamma$ and any term $t \in \mathrm{Tm}(\Delta, \sigma\{f\})$, there is a unique substitution $\langle f, t\rangle : \Delta \to \Gamma.\sigma$ with $\mathsf{p}(\Gamma, \sigma) \circ \langle f, t\rangle = f$ and $v\{\langle f, t\rangle\} = t$.

Notation 16. We write $\mathsf{p}(\sigma)$ for $\mathsf{p}(\Gamma, \sigma)$ when Γ is obvious.

The reader familiar with the notion of a category of elements will recognize that Definition 14 is the category of elements of some contravariant functor.

Exercise 17. Find the contravariant functor $F_{\Gamma,\sigma}$ and deduce that σ has a comprehension if and only if $F_{\Gamma,\sigma}$ is representable.

Scholium 18. The terminal object in $\mathrm{El}(F_{\Gamma,\sigma})$ models context extension.

We are now ready to define the notion of a category with families.

Definition 19. A *category with families* (CwF) is given by the following data:

- a category C with a terminal object;
- a functor $F : C^{\mathrm{op}} \to \mathsf{Fam}$;
- a comprehension for each $\Gamma \in C$ and $\sigma \in \mathrm{Ty}(\Gamma)$.

Terminology 20. We refer to the action on morphisms of F as *substitution* as well.

3.1 Examples of categories with families

Example 21. The category of contexts and substitutions, identified up to definitional equality, has a CwF structure in the following settings:

$$\mathrm{Ty}(\Gamma) := \{\sigma \mid \Gamma \vdash \sigma \text{ type}\}$$
$$\mathrm{Tm}(\Gamma, \sigma) := \{t \mid \Gamma \vdash t : \sigma\}$$

Each substitution $\Delta \vdash f : \Gamma$ is mapped to the usual substitution function $-[f]$.

The comprehension for σ is given by the substitution $\Gamma, x : \sigma \vdash (x_1, \ldots, x_n) : \Gamma$, where $\Gamma = x_1 : \sigma_1, \ldots, x_n : \sigma_n$, and the variable $\Gamma, x : \sigma \vdash x : \sigma$.

Example 22. The category $\mathbb{T} := \{\mathsf{ff} \leq \mathsf{tt}\}$ of truth values has a CwF structure with the following settings:

$$\mathrm{Ty}(\mathsf{ff}) := \mathrm{Ty}(\mathsf{tt}) = \{\mathsf{ff}, \mathsf{tt}\}$$
$$\mathrm{Tm}(\Gamma, \sigma) := \begin{cases} \mathbf{1}_{\mathsf{Set}} & \text{if } \Gamma \leq \sigma, \\ \mathbf{0}_{\mathsf{Set}} & \text{otherwise.} \end{cases}$$

The comprehension for σ is given by $\Gamma \wedge \sigma \leq \Gamma$ and the variable is the trivial element.

Example 23. The category Set of sets and functions has a CwF structure. We can consider a context Γ as a set of *stages*; then a type σ is a Γ-indexed set of terms and a term of type σ at stage γ is an element of σ_γ.

$$\mathrm{Ty}(\Gamma) := \{(\sigma_\gamma)_{\gamma \in \Gamma}, (\tau_\gamma)_{\gamma \in \Gamma}, \ldots\}$$
$$\mathrm{Tm}(\Gamma, \sigma) := \prod_{\gamma \in \Gamma} \sigma_\gamma$$

Here, $\prod_{\gamma \in \Gamma} \sigma_\gamma$ is the set-theoretic dependent function space; an element $t \in \prod_{\gamma \in \Gamma} \sigma_\gamma$ is a function mapping each $\gamma \in \Gamma$ to an element of σ_γ.

Substitution is given by precomposition: for each function $f : \Delta \to \Gamma$, the type $\sigma\{f\}$ is given component-wise by $(\sigma\{f\})_\delta := \sigma_{f(\delta)}$ and the term $t\{f\}$ is given by $t\{f\}(\delta) = t(f(\delta))$.

The comprehension for σ is given by the first project $\{(\gamma, x) \mid \gamma \in \Gamma, x \in \sigma_\gamma\} \to \Gamma$ and the variable $v \in \prod_{(\gamma,x) \in \Gamma.\sigma} \sigma_{f(\gamma,x)}$ is defined by $v(\gamma, x) = x$.

In the set model, there is no non-trivial structure between any two stages because a context, in this case, is *just* a set.

Certain applications require contexts to have additional structures; for example, to study how an object type theory embeds into the Logical Framework, we may take a stage to be a context in the object type theory. In this scenario, we have to accommodate a *category* of stages.

Example 24. Let $\mathcal{T}$ be a small category of stages; the presheaf category $\mathrm{Pr}(\mathcal{T})$ has a CwF structure that generalizes the set-theoretic one in Example 23.

Given a context $\Gamma : \mathcal{T}^{\mathrm{op}} \to \mathsf{Set}$, a type σ is a "Γ-indexed" family. To make sense of this, we compress Γ into its category of elements $\mathrm{El}(\Gamma)$; then a type σ is simply a presheaf of terms over $\mathrm{El}(\Gamma)$.

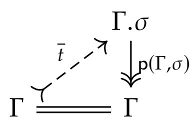
$$\mathrm{Ty}(\Gamma) := \{(\sigma_\gamma)_{\gamma \in \mathrm{El}(\Gamma)}, (\tau_\gamma)_{\gamma \in \mathrm{El}(\Gamma)}, \ldots\}$$

Terms, substitution, and comprehensions are given analogously as in Example 23.

3.2 Terms, sections, and weakening

The discussion in Section 2 gives us the impression that every term determines a substitution and every substitution of a particular kind corresponds to a term. We will expose this idea in greater detail in this section.

Let C be a CwF and $t \in \mathrm{Tm}(\Gamma, \sigma)$. Note that $t \in \mathrm{Tm}(\Gamma, \sigma\{\mathrm{id}_\Gamma\})$ by functoriality. Then comprehension yields a unique section $\bar{t} := \langle \mathrm{id}_\Gamma, t \rangle$ of the projection $\mathsf{p}(\Gamma, \sigma)$ such that $v\{\bar{t}\} = t$.



Conversely, let $f : \Gamma \to \Gamma.\sigma$ be a section of $\mathsf{p}(\Gamma, \sigma)$. This section corresponds to a term $v\{f\} \in \mathrm{Tm}(\Gamma, \sigma)$. Then comprehension yields a section $\overline{v\{f\}} : \Gamma \to \Gamma.\sigma$. By uniqueness, $f = \overline{v\{f\}}$; hence there is a bijective correspondence between terms and sections. In light of this correspondence, we may freely regard every term as a substitution.

Notation 25. We write t to mean both the term t and the substitution $\bar{t}$.

Contexts, types, and terms can be weakened; substitutions are no exceptions. Given a substitution $f : \Delta \to \Gamma$ and a type $\sigma \in \mathrm{Ty}(\Gamma)$, there ought to be a substitution $\mathsf{q}(f, \sigma) : \Delta.\sigma\{f\} \to \Gamma.\sigma$ that leaves the variable declaration $x : \sigma$ unchanged, while behaves like f on the rest of the context. Since we want to leave $x : \sigma$ unchanged, we take the variable $v_{\sigma\{f\}} \in \mathrm{Tm}(\Delta.\sigma\{f\}, \sigma\{f\})$ given by the comprehension for $\sigma\{f\}$; then the only substitution that makes sense is $\langle f \circ \mathsf{p}(\sigma\{f\}), v_{\sigma\{f\}} \rangle$.

Notation 26. To keep the notations clean, we will not spell out substitutions along weakening substitutions, such as $\mathsf{p}(\sigma)$ and $\mathsf{q}(f, \sigma)$, explicitly.

3.3 Interpreting types

So far, we have only defined how to interpret contexts and substitutions, which is insufficient for any interesting type theory, *i.e.*, a type theory with types. In this section, we will define the interpretations of types common found in Martin-Löf type theories.

3.3.1 The empty type

$$\text{0-F}\ \frac{}{\vdash \mathbb{0}\ \text{type}} \qquad \text{0-E}\ \frac{\Gamma, x : \mathbb{0} \vdash \tau\ \text{type}}{\Gamma, x : \mathbb{0} \vdash \mathrm{ind}^{\mathbb{0}}_{\tau}(x) : \tau}$$

Definition 27. A CwF C supports the empty type if the following data are given:

- *formation*: a type $\mathsf{empty} \in \mathrm{Ty}(\mathbf{1}_C)$;
- *elimination*: for each type $\tau \in \mathrm{Ty}(\Gamma.\,\mathsf{empty})$, a term $\mathsf{ind}^{\mathsf{empty}}_{\tau} \in \mathrm{Tm}(\Gamma.\,\mathsf{empty}, \tau)$.

The eliminator is required to be stable under substitution: for any $f : \Delta \to \Gamma$, $\mathsf{ind}^{\mathsf{empty}}_{\tau\{f\}} = \mathsf{ind}^{\mathsf{empty}}_{\tau}\{f\}$.

Example 28. The term model supports the empty type with the following settings:

$$\mathsf{empty} := \mathbb{0}$$
$$\mathsf{ind}^{\mathsf{empty}}_{\tau} := \mathrm{ind}^{\mathbb{0}}_{\tau}(x)$$

where $x : \mathbb{0}$ is a free variable.

Example 29. The truth value model supports the empty type with the following settings:

$$\mathsf{empty} := \mathrm{ff}$$
$$\mathsf{ind}^{\mathsf{empty}}_{\tau} := *$$

Remark 30. This is the only possible interpretation of $\mathbb{0}$ in the truth value model since we need to ensure that $\mathrm{Tm}(\Gamma.\,\mathsf{empty}, \tau)$ is nonempty for any τ.

Example 31. The set model supports the empty type with the following settings:

$$\mathsf{empty}_{\gamma} := \varnothing$$
$$\mathsf{ind}^{\mathsf{empty}}_{\tau} := !$$

where ! is the unique function mapping out of the empty set.

3.3.2 The unit type

$$\frac{}{\vdash \mathbb{1}\ \text{type}}\ \mathbb{1}\text{-F} \qquad \frac{}{\vdash \star : \mathbb{1}}\ \mathbb{1}\text{-I} \qquad \frac{\Gamma, x : \mathbb{1} \vdash \tau\ \text{type} \qquad \Gamma \vdash s : \tau[\star/x]}{\Gamma, x : \mathbb{1} \vdash \text{ind}^{\mathbb{1}}(x, s) : \tau}\ \mathbb{1}\text{-E} \qquad \frac{\Gamma, x : \mathbb{1} \vdash \tau\ \text{type} \qquad \Gamma \vdash s : \tau[\star/x]}{\Gamma \vdash \text{ind}^{\mathbb{1}}(\star, s) = s : \tau[\star/x]}\ \mathbb{1}\text{-}\beta$$

Definition 32. A CwF C supports the unit type if the following data are given:

- *formation*: a type $\mathsf{unit} \in \mathrm{Ty}(\mathbf{1}_C)$;
- *introduction*: a term $\star \in \mathrm{Tm}(\mathbf{1}_C, \mathsf{unit})$;
- *elimination*: for each type $\tau \in \mathrm{Ty}(\Gamma.\,\mathsf{unit})$, a function $\mathsf{ind}^{\mathsf{unit}}_{\tau} : \mathrm{Tm}(\Gamma, \tau\{\star\}) \to \mathrm{Tm}(\Gamma.\,\mathsf{unit}, \tau)$.

These data are subject to the following equations:

- *β-law*: $\mathsf{ind}^{\mathsf{unit}}(s)\{\star\} = s$;
- *stability under substitution*: for any morphism $f : \Delta \to \Gamma$, $\mathsf{ind}^{\mathsf{unit}}_{\tau}(s)\{f\} = \mathsf{ind}^{\mathsf{unit}}_{\tau\{f\}}(s\{f\})$.

Example 33. The term model supports the unit type with the following settings:

$$\begin{aligned} \mathsf{unit} &:= \mathbb{1} \\ \star &:= \star \\ \mathsf{ind}^{\mathsf{unit}}_{\tau}(s) &:= \text{ind}^{\mathbb{1}}(x, s) \end{aligned}$$

where $x : \mathbb{1}$ is a free variable.

Example 34. The truth value model supports the unit type with the following settings:

$$\begin{aligned} \mathsf{unit} &:= \mathsf{tt} \\ \star &:= * \\ \mathsf{ind}^{\mathsf{unit}}_{\tau}(s) &:= \,! \end{aligned}$$

Remark 35. This is the only interpretation of the unit type in the truth value model since we need to ensure that $\mathrm{Tm}(\mathbf{1}_{\mathbb{T}}, \mathsf{unit})$ is nonempty.

Example 36. The set model supports the unit type with the following settings:

$$\begin{aligned} \mathsf{unit}_{\gamma} &:= \mathbf{1}_{\mathsf{Set}} \\ \star &:= \mathrm{id}_{\mathbf{1}_{\mathsf{Set}}} \\ \mathsf{ind}^{\mathsf{unit}}_{\tau}(s)(\gamma, *) &:= * \end{aligned}$$

where $*$ is the unique element of $\mathbf{1}_{\mathsf{Set}}$.

3.3.3 The natural number type

$$\frac{}{\vdash \mathbb{N}\ \text{type}}\ \mathbb{N}\text{-F} \qquad \frac{}{\vdash 0 : \mathbb{N}}\ \mathbb{N}\text{-I}_z \qquad \frac{\Gamma \vdash u : \mathbb{N}}{\Gamma \vdash \text{suc}(u) : \mathbb{N}}\ \mathbb{N}\text{-I}_{\text{suc}}$$

$$\frac{\begin{array}{c}\Gamma, x : \mathbb{N} \vdash \tau\ \text{type} \\ \Gamma \vdash s : \tau[0/x]\end{array} \qquad \Gamma, y : \mathbb{N}, p : \tau[y/x] \vdash t : \tau[\text{suc}(y)/x]}{\Gamma, n : \mathbb{N} \vdash \text{ind}^{\mathbb{N}}(n, s, t) : \tau[n/x]}\ \mathbb{N}\text{-E}$$

$$\frac{\Gamma, x : \mathbb{N} \vdash \tau\ \text{type} \qquad \Gamma \vdash s : \tau[0/x] \qquad \Gamma, y : \mathbb{N}, p : \tau[y/x] \vdash t : \tau[\text{suc}(y)/x]}{\Gamma \vdash \text{ind}^{\mathbb{N}}(0, s, t) = s : \tau[0/x]}\ \mathbb{N}\text{-}\beta_1$$

$$\frac{\Gamma, x : \mathbb{N} \vdash \tau\ \text{type} \qquad \Gamma \vdash s : \tau[0/x] \qquad \Gamma, y : \mathbb{N}, p : \tau[y/x] \vdash t : \tau[\text{suc}(y)/x]}{\Gamma, n : \mathbb{N} \vdash \text{ind}^{\mathbb{N}}(\text{suc}(n), s, t) = t[n/y, \text{ind}^{\mathbb{N}}(n, s, t)/p] : \tau[\text{suc}(n)/x]}\ \mathbb{N}\text{-}\beta_2$$

Definition 37. A CwF C supports the natural number type if the following data are given:

- *formation*: $\mathsf{nat} \in \mathrm{Ty}(\mathbf{1}_C)$;
- *introduction*: a term $\mathsf{zero} \in \mathrm{Tm}(\mathbf{1}_C, \mathsf{nat})$ together with a function $\mathsf{suc}_\Gamma : \mathrm{Tm}(\Gamma, \mathsf{nat}) \to \mathrm{Tm}(\Gamma, \mathsf{nat})$;
- *elimination*: for each $\tau \in \mathrm{Ty}(\Gamma.\,\mathsf{nat})$, a function

$$\mathsf{ind}^{\mathsf{nat}}_\tau : \mathrm{Tm}(\Gamma, \tau\{\mathsf{zero}\}) \times \mathrm{Tm}(\Gamma.\,\mathsf{nat}\,.\tau, \tau\{\mathsf{suc}(v_{\mathsf{nat}})\}) \to \mathrm{Tm}(\Gamma.\,\mathsf{nat}, \tau\{v_{\mathsf{nat}}\})$$

These data are required to be stable under substitution and the following equations must hold:

$$\mathsf{ind}^{\mathsf{nat}}_\tau(s,t)\{\mathsf{zero}\} = s$$
$$\mathsf{ind}^{\mathsf{nat}}_\tau(s,t)\{\mathsf{suc}(n)\} = t\{n\}\{\mathsf{ind}^{\mathsf{nat}}(s,t)\}$$

Example 38. The truth value model supports the natural number type with the following settings:

$$\mathsf{nat} := \mathrm{tt}$$
$$\mathsf{zero} := *$$
$$\mathsf{suc} := \mathrm{id}$$
$$\mathsf{ind}^{\mathsf{nat}}_\tau := !$$

3.3.4 Universes

$$\frac{\mathcal{U}\text{-F}}{\vdash \mathcal{U}\ \mathrm{type}} \qquad \text{El-F}\ \frac{\Gamma \vdash t : \mathcal{U}}{\Gamma \vdash \mathrm{El}(t)\ \mathrm{type}}$$

Definition 39. A universe is *closed under* $\mathbb{0}$ *and* $\mathbb{1}$ if we additionally have the following rules:

$$\mathbb{0}\text{-}\mathcal{U}\ \frac{}{\vdash \hat{\mathbb{0}} : \mathcal{U}} \qquad \mathbb{1}\text{-}\mathcal{U}\ \frac{}{\vdash \hat{\mathbb{1}} : \mathcal{U}} \qquad \mathbb{0}\text{-El}\ \frac{}{\vdash \mathrm{El}(\hat{\mathbb{0}}) = \mathbb{0}\ \mathrm{type}} \qquad \mathbb{1}\text{-El}\ \frac{}{\vdash \mathrm{El}(\hat{\mathbb{1}}) = \mathbb{1}\ \mathrm{type}}$$

Definition 40. Let C be a CwF supporting $\mathbb{0}$ and $\mathbb{1}$. C supports a universe closed under $\mathbb{0}$ and $\mathbb{1}$ if the following data are given:

- *formation*: $\mathcal{U} \in \mathrm{Ty}(\mathbf{1}_C)$;
- *decoding function*: for each Γ, a function $\mathrm{El}_\Gamma : \mathrm{Tm}(\Gamma, \mathcal{U}) \to \mathrm{Ty}(\Gamma)$.
- *the empty type*: $\widehat{\mathsf{empty}} \in \mathrm{Tm}(\mathbf{1}_C, \mathcal{U})$;
- *the unit type*: $\widehat{\mathsf{unit}} \in \mathrm{Tm}(\mathbf{1}_C, \mathcal{U})$.

These data are required to be stable under substitution and the following equations must hold:

$$\mathrm{El}(\widehat{\mathsf{empty}}) = \mathsf{empty}$$
$$\mathrm{El}(\widehat{\mathsf{unit}}) = \mathsf{unit}$$

Proposition 41. *The truth value model does not support a universe closed under* $\mathbb{0}$ *and* $\mathbb{1}$.

Proof. Suppose that it does, then we have $\widehat{\mathsf{empty}} = \widehat{\mathsf{unit}}$ since $\mathrm{Tm}(\mathbf{1}_{\mathbb{T}}, \mathcal{U})$ is a singleton set; hence $\mathsf{empty} = \mathsf{unit}$. By Examples 29 and 34, this implies that $\mathrm{ff} = \mathrm{tt}$, which is not the case. □

The set model supports a universe closed under more than $\mathbb{0}$ and $\mathbb{1}$; for instance, we can take the usual Grothendieck universe.

3.3.5 Dependent product types

Π-F
$$\frac{\Gamma, x : \sigma \vdash \tau \text{ type}}{\Gamma \vdash \Pi x : \sigma.\tau \text{ type}}$$

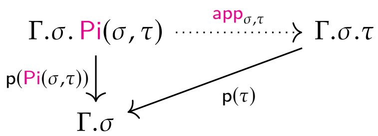
Π-I
$$\frac{\Gamma, x : \sigma \vdash t : \tau}{\Gamma \vdash \lambda x : \sigma.t : \Pi x : \sigma.\tau}$$

Π-E
$$\frac{\Gamma \vdash f : \Pi x : \sigma.\tau}{\Gamma, u : \sigma \vdash f(u) : \tau[u/x]}$$

Π-β
$$\frac{\Gamma, x : \sigma \vdash t : \tau \qquad \Gamma \vdash s : \sigma}{\Gamma \vdash (\lambda x : \sigma.t)(s) = t[s/x] : \tau[s/x]}$$

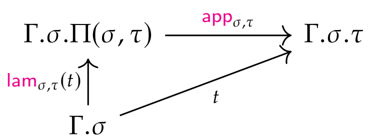
Definition 42. A CwF C supports Π-types if for any two types $\sigma \in \mathrm{Ty}_C(\Gamma)$ and $\tau \in \mathrm{Ty}_C(\Gamma.\sigma)$, we have the following data:

- *formation*: a type $\mathsf{Pi}(\sigma, \tau) \in \mathrm{Ty}_C(\Gamma)$;
- *introduction*: for each term $t \in \mathrm{Tm}_C(\Gamma.\sigma, \tau)$, there is a term $\mathsf{lam}_{\sigma,\tau}(t) \in \mathrm{Tm}_C(\Gamma, \mathsf{Pi}(\sigma, \tau))$;
- *elimination*: a morphism in the following configuration:



These data are subject to the following conditions:

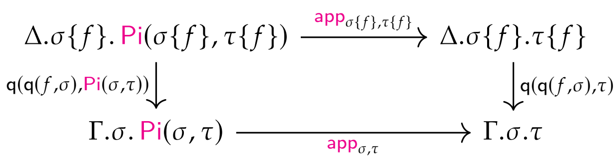
- *β-law*: for every term $t \in \mathrm{Tm}_C(\Gamma.\sigma, \tau)$, the following diagram commutes:



- *stability under substitution*: for every morphism $f : \Delta \to \Gamma$, one has the following equations:

$$\mathsf{Pi}(\sigma, \tau)\{f\} = \mathsf{Pi}(\sigma\{f\}, \tau)$$
$$\mathsf{lam}_{\sigma,\tau}(t)\{f\} = \mathsf{lam}_{\sigma\{f\},\tau}(t)$$

  and the following diagram commutes:



Example 43. The term model of a type theory supports Π-types with the evident settings:

$$\mathsf{Pi}(\sigma, \tau) := \Pi x : \sigma.\tau$$
$$\mathsf{lam}_{\sigma,\tau}(t) := \lambda x : \sigma.t$$

and $\mathsf{app}_{\sigma,\tau}$ is given by the substitution

$$\Gamma, y : \sigma, z : \Pi x : \sigma.\tau \vdash (\gamma, z(y)) : \Gamma, x : \sigma, w : \tau$$

Example 44. The truth value model supports Π-types with the following settings:

$$\mathsf{Pi}(\sigma, \tau) := \sigma \to \tau$$
$$\mathsf{lam}_{\sigma,\tau}(t) := *$$
$$\mathsf{app}_{\sigma,\tau} := \Gamma \wedge \sigma \wedge \mathsf{Pi}(\sigma, \tau) \leq \Gamma \wedge \sigma \wedge \tau$$

Example 45. The set-theoretic model supports Π-types with the following settings:

$$\begin{aligned} \mathsf{Pi}(\sigma,\tau)_\gamma &:= \prod_{x\in\sigma_\gamma} \tau_{(\gamma,x)} \\ \mathsf{lam}_{\sigma,\tau}(t) &:= \lambda\gamma\in\Gamma.\lambda x\in\sigma_\gamma.t(\gamma,x) \\ \mathsf{app}_{\sigma,\tau}(\gamma,x,f) &:= (\gamma,f(x)) \end{aligned}$$

3.3.6 Intensional identity types

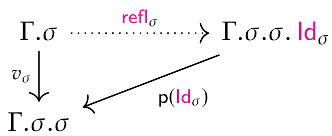
$$\text{Id-F}\quad \frac{\Gamma\vdash a:\sigma \qquad \Gamma\vdash b:\sigma}{\Gamma\vdash \mathrm{Id}_\sigma(a,b)\ \text{type}} \qquad \text{Id-I}\quad \frac{\Gamma\vdash a:\sigma}{\Gamma\vdash \mathrm{refl}_\sigma(a):\mathrm{Id}_\sigma(a,a)}$$

$$\text{Id-E}\quad \frac{\Gamma,x:\sigma,y:\sigma,p:\mathrm{Id}_\sigma(x,y)\vdash\tau\ \text{type} \qquad \Gamma,z:\sigma\vdash s:\tau[z/x,z/y,\mathrm{refl}_\sigma(z)/p]}{\Gamma,x:\sigma,y:\sigma,p:\mathrm{Id}_\sigma(x,y)\vdash \mathrm{ind}^{\mathrm{Id}}(p,s):\tau}$$

$$\text{Id-}\beta\quad \frac{\Gamma\vdash a:\sigma \qquad \Gamma,z:\sigma\vdash s:\tau[z/x,z/y,\mathrm{refl}_\sigma(z)/p]}{\Gamma\vdash \mathrm{ind}^{\mathrm{Id}}(\mathrm{refl}_\sigma(a),s)=s[a/z]:\tau[a/x,a/y,\mathrm{refl}_\sigma(a)/p]}$$

Definition 46. A CwF C supports (intensional) identity types if for every $\sigma\in\mathrm{Ty}_C(\Gamma)$, the following data are given:

- *formation*: a type $\mathsf{Id}_\sigma\in\mathrm{Ty}_C(\Gamma.\sigma.\sigma)$;
- *introduction*: a morphism $\mathsf{refl}_\sigma:\Gamma.\sigma\longrightarrow\Gamma.\sigma.\sigma.\,\mathsf{Id}_\sigma$ in the following configuration:



- *elimination*: for each type $\tau\in\mathrm{Ty}_C(\Gamma.\sigma.\sigma.\,\mathsf{Id}_\sigma)$, a function $\mathsf{ind}^{\mathsf{Id}}_{\sigma,\tau}:\mathrm{Tm}_C(\Gamma.\sigma,\tau\{\mathsf{refl}_\sigma\})\longrightarrow\mathrm{Tm}_C(\Gamma.\sigma.\sigma.\,\mathsf{Id}_\sigma,\tau)$.

These data are required to be stable under substitution and additionally the following equation holds for every term $t\in\mathrm{Tm}_C(\Gamma.\tau,\tau\{\mathsf{refl}_\sigma\})$:

$$\mathsf{ind}^{\mathsf{Id}}_{\sigma,\tau}(t)\{\mathsf{refl}_\sigma\}=t \qquad\qquad \beta\text{-law}$$

Example 47. The term model of intensional type theory supports identity types with the following settings:

$$\begin{aligned} \mathsf{Id}_\sigma &:= \Gamma,x:\sigma,y:\sigma\vdash\mathrm{Id}_\sigma(x,y)\ \text{type} \\ \mathsf{refl}_\sigma &:= \Gamma,x:\sigma\vdash(\gamma,x,x,\mathrm{refl}_\sigma(x)):\Gamma,x:\sigma,y:\sigma,p:\mathrm{Id}_\sigma(x,x) \\ \mathsf{ind}^{\mathsf{Id}}_{\sigma,\tau}(t) &:= \mathrm{ind}^{\mathrm{Id}}(p,t) \end{aligned}$$

where p is a free variable.

Example 48. The truth value model supports identity types with the following settings:

$$\begin{aligned} \mathsf{Id}_\sigma &:= \mathrm{tt} \\ \mathsf{refl}_\sigma &:= \Gamma\wedge\sigma\le\Gamma\wedge\sigma\wedge\sigma\wedge\mathsf{Id}_\sigma \\ \mathsf{ind}^{\mathsf{Id}}_{\sigma,\tau} &:= ! \end{aligned}$$

Example 49. The set model supports identity types with the following settings:

$$\begin{aligned} (\mathsf{Id}_\sigma)_{(\gamma,x,y)} &:= \begin{cases} \mathbf{1}_{\mathsf{Set}} & \text{if } x=y; \\ \mathbf{0}_{\mathsf{Set}} & \text{otherwise} \end{cases} \\ \mathsf{refl}_\sigma(\gamma,x) &:= (\gamma,x,x,*) \\ \mathsf{ind}^{\mathsf{Id}}_{\sigma,\tau}(s)(\gamma,x,y,p) &:= s(\gamma,x) \end{aligned}$$

3.3.7 Dependent sum types

$$\Sigma\text{-F}\quad \frac{\Gamma, x : \sigma \vdash \tau \text{ type}}{\Gamma \vdash \Sigma x : \sigma.\tau \text{ type}} \qquad \Sigma\text{-I}\quad \frac{\Gamma \vdash a : \sigma \qquad \Gamma \vdash b : \tau[a/x]}{\Gamma \vdash \langle a, b\rangle : \Sigma x : \sigma.\tau} \qquad \Sigma\text{-E}\quad \frac{\begin{array}{c}\Gamma, z : \Sigma x : \sigma.\tau \vdash \rho \text{ type}\\ \Gamma, x : \sigma, y : \tau \vdash s : \rho[\langle x, y\rangle/z]\end{array}}{\Gamma, z : \Sigma x : \sigma.\tau \vdash \text{ind}^{\Sigma}(z, s) : \rho}$$

$$\Sigma\text{-}\beta\quad \frac{\Gamma \vdash a : \sigma \qquad \Gamma \vdash b : \tau[a/x] \qquad \Gamma, x : \sigma, y : \tau \vdash s : \rho[\langle x, y\rangle/z]}{\text{ind}^{\Sigma}(\langle a, b\rangle, s) = s[a/x, b/y] : \rho[\langle a, b\rangle/z]}$$

Definition 50. A CwF C supports Σ-types if the following data are given for any two types $\sigma \in \text{Ty}_C(\Gamma)$ and $\tau \in \text{Ty}_C(\Gamma.\sigma)$:

- *formation*: a type $\mathsf{Sig}(\sigma, \tau) \in \text{Ty}_C(\Gamma)$;
- *introduction*: a morphism in the following configuration:

$$\begin{array}{ccc} \Gamma.\sigma.\tau & \xrightarrow{\mathsf{pair}_{\sigma,\tau}} & \Gamma.\mathsf{Sig}(\sigma, \tau) \\ {\scriptstyle \mathsf{p}(\tau)}\downarrow & & \downarrow{\scriptstyle \mathsf{p}(\mathsf{Sig}(\sigma,\tau))} \\ \Gamma.\sigma & \xrightarrow[\mathsf{p}(\sigma)]{} & \Gamma \end{array}$$

- *elimination*: for every type $\rho \in \text{Ty}_C(\Gamma.\mathsf{Sig}(\sigma, \tau))$ and term $t \in \text{Tm}_C(\Gamma.\sigma.\tau, \rho\{\mathsf{pair}_{\sigma,\tau}\})$, a term $\mathsf{ind}^{\mathsf{Sig}}_{\sigma,\tau,\rho}(t) \in \text{Tm}_C(\Gamma.\mathsf{Sig}(\sigma, \tau), \rho)$.

These data are subject to the following conditions:

- *β-law*: for every type $\rho \in \text{Ty}_C(\Gamma.\mathsf{Sig}(\sigma, \tau))$ and term $t \in \text{Tm}_C(\Gamma.\sigma.\tau, \rho\{\mathsf{pair}_{\sigma,\tau}\})$, the equation $\mathsf{ind}^{\mathsf{Sig}}_{\sigma,\tau,\rho}(t)\{\mathsf{pair}_{\sigma,\tau}\} = t$ holds;
- stability under substitution.

Example 51. The term model supports Σ-types with the following settings:

$$\begin{aligned} \mathsf{Sig}(\sigma, \tau) &:= \Sigma x : \sigma.\tau \\ \mathsf{pair}_{\sigma,\tau} &:= \Gamma, x : \sigma, y : \tau \vdash (\gamma, \langle u, s\rangle) : \Gamma, z : \Sigma x : \sigma.\tau \\ \mathsf{ind}^{\mathsf{Sig}}_{\sigma,\tau,\rho}(s) &:= \text{ind}^{\Sigma}(x, s) \end{aligned}$$

where $x : \Sigma x : \sigma.\tau$ is a free variable.

Example 52. The truth value model supports Σ-types with the following settings:

$$\begin{aligned} \mathsf{Sig}(\sigma, \tau) &:= \sigma \wedge \tau \\ \mathsf{pair}_{\sigma,\tau} &:= \Gamma \wedge \sigma \wedge \tau \le \Gamma \wedge \mathsf{Sig}(\sigma, \tau) \\ \mathsf{ind}^{\mathsf{Sig}}_{\sigma,\tau,\rho} &:= ! \end{aligned}$$

Example 53. The set model supports Σ-types with the following settings:

$$\begin{aligned} \mathsf{Sig}(\sigma, \tau)_{\gamma\in\Gamma} &:= \{(x, y) : x \in \sigma_\gamma, y \in \tau_{(\gamma,x)}\} \\ \mathsf{pair}_{\sigma,\tau}(\gamma, x, y) &:= (\gamma, (x, y)) \\ \mathsf{ind}^{\mathsf{Sig}}_{\sigma,\tau,\rho}(s) &:= \lambda(\gamma, (x, y)) \in \Gamma.\mathsf{Sig}(\sigma, \tau).s(\gamma, x, y) \end{aligned}$$

4 Peano's 4th axiom

A famous example in an introduction to type theory course is to show that 0 is not 1, *i.e.*, the type $\text{Id}(0, \text{suc}(0)) \to \mathbb{0}$ is inhabited. The proof (in Agda) usually goes as follows. First, we define the following function that essentially characterizes equality on the natural numbers:

```
eq : nat -> nat -> Set
eq z z = unit
eq z (suc n) = empty
eq (suc m) z = empty
eq (suc m) (suc n) = eq m n
```

where Set is a universe closed under $\mathbb{0}$ and $\mathbb{1}$.

Then we show that eq is complete for $\mathrm{Id}_{\mathbb{N}}(x, y)$, *i.e.*, if $\mathrm{Id}_{\mathbb{N}}(x, y)$, then eq x y is inhabited.

```
complete : (x y : nat) -> Id x y -> eq x y
complete z .z (refl .z) = *
complete (suc x) .(suc x) (refl .(suc x)) = complete x x (refl x)
```

And finally, we can now easily find a term of type $\mathrm{Id}(0, \mathrm{suc}(0)) \longrightarrow \mathbb{0}$.

```
pa4 : Id 0 (suc 0) -> empty
pa4 p = complete 0 (suc 0) p
```

Remark 54. Doing a pattern match on p is cheating.

In hindsight, it is not clear why we need to bother with the function eq. The following proposition shows that it is necessary.

Proposition 55. *The type* $\mathrm{Id}(0, \mathrm{suc}(0)) \longrightarrow \mathbb{0}$ *is not inhabited in a Martin-Löf type theory* $\mathcal{T}$ *without universes.*

Proof. If the type is inhabited, then $\mathrm{Tm}(\mathbf{1}_C, \mathsf{Id}_{\mathsf{nat}}\{\mathsf{zero}, \mathsf{suc}(\mathsf{zero})\} \longrightarrow \mathsf{empty})$ is not empty for all CwF C that supports $\mathcal{T}$.

Since $\mathcal{T}$ has no universes, it can be interpreted into the truth value model; but $\mathrm{Tm}(\mathbf{1}_{\mathbb{T}}, \mathsf{Id}_{\mathsf{nat}}\{\mathsf{zero}, \mathsf{suc}(\mathsf{zero})\} \rightarrow \mathsf{empty})$ is empty because the latter type is interpreted as the exponential $\mathrm{tt} \longrightarrow \mathsf{ff}$, *i.e.*, ff. □

References

[Hof97] Martin Hofmann. *Syntax and Semantics of Dependent Types*, page 79–130. Publications of the Newton Institute. Cambridge University Press, 1997.