CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Normalization for Type Theory with an Impredicative Universe

Master's thesis in Computer science and engineering

ZONGPU XIE

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2024

Master's thesis 2024

Normalization for Type Theory with an Impredicative Universe

ZONGPU XIE

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2024

Normalization for Type Theory with an Impredicative Universe
ZONGPU XIE

Supervisor: Thierry Coquand, Department of Computer Science and Engineering
Examiner: Andreas Abel, Department of Computer Science and Engineering

Master's Thesis 2024
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2024

Normalization for Type Theory with an Impredicative Universe

ZONGPU XIE
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

This thesis presents a novel proof of canonicity and normalization for a type theory a proof relevant impredicative universe and a hierarchy of predicative universes. The proof uses Artin gluing, and is structured in a modular way that makes it easier to extend to new type formers.

Keywords: type theory, normalization, impredicativity.

Acknowledgements

I would like to thank my supervisor Thierry Coquand, and I would like to thank Wen Kokke and Ambrus Kaposi for the help with finishing this thesis.

Zongpu Xie, Budapest, 18 March 2024

Contents

List of Figures

1

Introduction

Type theory is a formal system which can be used both as a foundation of mathematics and as a basis for proof assistants to check the proof of mathematical theorems and computer programs [12]. One key idea is to use the correspondence between propositions and types to reduce proof checking to type checking, for which it is crucial that this the type checking is decidable. The main result of this work is to provide a proof of decidability of type checking for a type theory with a hierarchy of universes and a proof relevant impredicative universe as defined by Coquand [2]. This theory is difficult to formalize due to the proof relevant impredicative universe.

Decidability of type checking can be reduced to proving the normalization property, which is a refinement of the canonicity property. Canonicity states that every term of the type of Booleans is convertible to either true or false in the empty context. Normalization is a property which states that terms in arbitrary contexts are convertible to normal forms with decidable equality.

This proof of canonicity and normalization is done using the technique of Artin gluing presented by Kaposi et al. [10]. This makes the proof modular and easy to extend to more type formers.

2

Background

We use intuitionistic set theory with an infinite hierarchy of universes $\mathscr{U}_n$ as the metatheory. We do not assume impredicativity in the metatheory.

2.1 Syntax

The grammar of terms, types, and contexts is given in fig. 2.1. The typing rules for the object type theory are listed in fig. 2.2 and fig. 2.3. The type theory has an infinite cumulative hierarchy of universes U_n, and an impredicate universe Prop with an operation T to lift an element of Prop to U_0. We write Prop for the impredicative universe and $\forall$ for the function space in Prop to clearly separate them from U and Π.

This type system also has the type Bool in Prop with large elimination, as such, the Prop universe is not proof irrelevant. It is possible to introduct other data types such as natural numbers as well (But some data types cannot have large

$$
\begin{aligned}
A, B, a, b, t ::= {} & x \mid \mathsf{U}_n \mid \mathsf{Prop} \mid \mathsf{T}(A) \\
& \mid \Pi_{x:A} B \mid t\ a \mid \lambda x.\, b \mid \forall_{x:A} B \mid \mathsf{un}(t) \mid \mathsf{mk}(t) \\
& \mid \mathsf{Bool} \mid \mathsf{false} \mid \mathsf{true} \mid \mathsf{elim}_{\mathsf{Bool}}(x.\, A, a_0, a_1, t) \\
\Gamma ::= {} & () \mid \Gamma,\ x : A
\end{aligned}
$$

Figure 2.1: BNF grammar of the syntax

$$\frac{}{() \ \mathsf{ctx}} \qquad \frac{\Gamma \vdash A \ \mathsf{type}_n}{\Gamma, \ x : A \ \mathsf{ctx}}$$

$$\frac{}{\Gamma \vdash \mathsf{U}_n \ \mathsf{type}_{n+1}} \qquad \frac{\Gamma \vdash A : \mathsf{U}_n}{\Gamma \vdash A \ \mathsf{type}_n}$$

$$\frac{}{\mathsf{Prop} \ \mathsf{type}_0} \qquad \frac{\Gamma \vdash A : \mathsf{Prop}}{\Gamma \vdash \mathsf{T}(A) \ \mathsf{type}_0} \qquad \frac{\Gamma \vdash A \ \mathsf{type}_n \qquad \Gamma, \ x : A \vdash B \ \mathsf{type}_n}{\Gamma \vdash \Pi_{x:A} B \ \mathsf{type}_n}$$

$$\frac{\Gamma \vdash a : A_0 \qquad \Gamma \vdash A_0 = A_1 \ \mathsf{type}_n}{\Gamma \vdash a : A_1} \qquad \frac{\Gamma \ \mathsf{ctx}}{\Gamma \vdash x : A} \ (x : A \text{ in } \Gamma)$$

$$\frac{\Gamma \vdash A \ \mathsf{type}_n}{\Gamma \vdash A : \mathsf{U}_n} \qquad \frac{\Gamma \vdash A : \mathsf{U}_n}{\Gamma \vdash A : \mathsf{U}_{n+1}}$$

$$\frac{\Gamma \vdash t : \Pi_{x:A} B \qquad \Gamma \vdash a : A}{\Gamma \vdash t \, a : B[a/x]} \qquad \frac{\Gamma, \ x : A \vdash b : B}{\Gamma \vdash \lambda x. \, b : \Pi_{x:A} B}$$

$$\frac{\Gamma \vdash A \ \mathsf{type}_n \qquad \Gamma, \ x : A \vdash B : \mathsf{Prop}}{\Gamma \vdash \forall_{x:A} B : \mathsf{Prop}}$$

$$\frac{\Gamma \vdash t : \mathsf{T}(\forall_{x:A} B)}{\Gamma \vdash \mathsf{un}(t) : \Pi_{x:A} \mathsf{T}(B)} \qquad \frac{\Gamma \vdash t : \Pi_{x:A} \mathsf{T}(B)}{\Gamma \vdash \mathsf{mk}(t) : \mathsf{T}(\forall_{x:A} B)}$$

$$\frac{}{\Gamma \vdash \mathsf{Bool} : \mathsf{Prop}} \qquad \frac{}{\Gamma \vdash \mathsf{false} : \mathsf{T}(\mathsf{Bool})} \qquad \frac{}{\Gamma \vdash \mathsf{true} : \mathsf{T}(\mathsf{Bool})}$$

$$\frac{\begin{array}{c}\Gamma, \ x : \mathsf{T}(\mathsf{Bool}) \vdash A \ \mathsf{type}_n \\ \Gamma \vdash a_0 : A[\mathsf{false}/x] \qquad \Gamma \vdash a_1 : A[\mathsf{true}/x] \qquad \Gamma \vdash t : \mathsf{Bool}\end{array}}{\Gamma \vdash \mathsf{elim}_{\mathsf{Bool}}(x. \, A, a_0, a_1, t) : A[t/x]}$$

Figure 2.2: Typing judgements for contexts, types, and terms

$$\frac{\Gamma,\ x\ :\ A \vdash b\ :\ B \qquad \Gamma \vdash a\ :\ A}{\Gamma \vdash (\lambda x.\, b)\ a = b[a/x]\ :\ B[a/x]} \qquad \frac{\Gamma \vdash t\ :\ \Pi_{x:A} B}{\Gamma \vdash (\lambda x.\, t\ x) = t\ :\ \Pi_{x:A} B}$$

$$\frac{\Gamma \vdash t\ :\ \Pi_{x:A} \mathsf{T}(B)}{\Gamma \vdash \mathsf{un}(\mathsf{mk}(t)) = t\ :\ \Pi_{x:A} \mathsf{T}(B)} \qquad \frac{\Gamma \vdash t\ :\ \mathsf{T}(\forall_{x:A} B)}{\Gamma \vdash \mathsf{mk}(\mathsf{un}(t)) = t\ :\ \mathsf{T}(\forall_{x:A} B)}$$

$$\frac{\Gamma,\ x\ :\ \mathsf{T}(\mathsf{Bool}) \vdash A\ \ \mathsf{type}_n \qquad \Gamma \vdash a_0\ :\ A[\mathsf{false}/x] \qquad \Gamma \vdash a_1\ :\ A[\mathsf{true}/x]}{\Gamma \vdash \mathsf{elim}_{\mathsf{Bool}}(x.\, A, a_0, a_1, \mathsf{false}) = a_0\ :\ A[\mathsf{false}/x]}$$

$$\frac{\Gamma,\ x\ :\ \mathsf{T}(\mathsf{Bool}) \vdash A\ \ \mathsf{type}_n \qquad \Gamma \vdash a_0\ :\ A[\mathsf{false}/x] \qquad \Gamma \vdash a_1\ :\ A[\mathsf{true}/x]}{\Gamma \vdash \mathsf{elim}_{\mathsf{Bool}}(x.\, A, a_0, a_1, \mathsf{true}) = a_1\ :\ A[\mathsf{true}/x]}$$

Figure 2.3: Equality judgements for terms, with equivalence and congruence rules omitted

eliminations without inconsistency [5]). Suprisingly we can prove the negation of the law of excluded middle [3].

The syntax has separate judgements for types and terms so that it closely matches the notion of a model of this theory.

The conversion rules are given as equality judgements, which support more models than if it were defined as a separate function as in Geuvers and Werner [8]. It is not clear a priori that the type constructors are injective or that subject reduction holds.

We only have a strict isomorphism with un and mk to convert between between $\mathsf{T}(\forall_{x:A} B)$ and $\Pi_{x:A} \mathsf{T}(B)$, however with the technique by Shulman [4] we can construct a model with the equality:

$$\frac{\Gamma,\ x\ :\ A \vdash B\ :\ \mathsf{Prop}}{\Gamma \vdash \mathsf{T}(\forall_{x:A} B) = \Pi_{x:A} \mathsf{T}(B)\ \ \mathsf{type}_0}$$

2.2 Category with families

Categories with families (CwFs) [1], [7] give a generalized algebraic theory presentation of type theory. The initial model of such theory can be seen as the term model of the type theory with De Bruijn indices and explicit parallel substitutions. We list the components below.

CwFs have four sorts: a collection of contexts, a set of substitutions $\Delta \to \Gamma$ if Δ and Γ are contexts, a set $\mathsf{Type}(\Gamma)$ for types in context Γ, and a set $\mathsf{Elem}(\Gamma, A)$ for the elements of type A in context Γ.

Contexts and substitutions form a category: we have an identity substitution $\mathsf{id} \in \Gamma \to \Gamma$ for any Γ, and if we have substitutions $\gamma \in \Delta \to \Gamma$ and $\delta \in \Theta \to \Delta$, then they can be composed as $\gamma\delta \in \Theta \to \Gamma$. The composition is associative with id as the identity:

$$\gamma(\delta\theta) = (\gamma\delta)\theta \qquad\qquad \mathsf{id}\ \gamma = \gamma = \gamma\ \mathsf{id}$$

There is a terminal context denoted as $()$ with a unique substitution $() \in \Gamma \to ()$ for any Γ.

Substitutions can be applied to types and elements: if $A \in \mathsf{Type}(\Gamma)$ and $\gamma \in \Delta \to \Gamma$, we have $A\gamma \in \mathsf{Type}(\Delta)$, if $a \in \mathsf{Elem}(\Gamma, A)$ and $\gamma \in \Delta \to \Gamma$ we have $a\gamma \in \mathsf{Elem}(\Delta, A\gamma)$. The following equations hold:

$$A\ \mathsf{id} = A \qquad A(\gamma\delta) = (A\gamma)\delta \qquad a\ \mathsf{id} = a \qquad a(\gamma\delta) = (a\gamma)\delta$$

We use concatenation for both composition and application of substitutions, it can be inferred from the left element.

There is a context extension operation where $\Gamma.A$ is a context if $A \in \mathsf{Type}(\Gamma)$. We have the projections $\mathsf{p} \in \Gamma.A \to \Gamma$ and $\mathsf{q} \in \mathsf{Elem}(\Gamma.A, A\mathsf{p})$, and if $\gamma \in \Delta \to \Gamma$ and $a \in \mathsf{Elem}(\Delta, A\gamma)$ we have $(\gamma, a) \in \Delta \to \Gamma.A$. The equations are:

$$\mathsf{p}(\gamma, a) = \gamma \qquad \mathsf{q}(\gamma, a) = a \qquad (\gamma, a)\delta = (\gamma\delta, a\delta) \qquad (\mathsf{p}, \mathsf{q}) = \mathsf{id}$$

If $\gamma \in \Delta \to \Gamma$, we can define $\gamma^+ \in \Delta.A\gamma \to \Gamma.A$ as $(\gamma\mathsf{p}, q)$.

For our type theory, we index the types such that $\mathsf{Type}_n \subseteq \mathsf{Type}_{n+1}$, then we have $\mathsf{U}_n \in \mathsf{Type}_{n+1}(\Gamma)$ and $\mathsf{Elem}(\Gamma, \mathsf{U}_n) = \mathsf{Type}_n(\Gamma)$.

It is straightforward to add the type formers:

- $\mathsf{Prop} \in \mathsf{Type}_0(\Gamma)$
- $\mathsf{T}(A) \in \mathsf{Type}_0(\Gamma)$ where $A \in \mathsf{Elem}(\Gamma, \mathsf{Prop})$
- $\Pi_A B \in \mathsf{Type}_n(\Gamma)$ where $A \in \mathsf{Type}_n(\Gamma)$ and $B \in \mathsf{Type}_n(\Gamma.A)$
- $\mathsf{app}(t, a) \in \mathsf{Elem}(\Gamma, B(\mathsf{id}, a))$ where $t \in \mathsf{Elem}(\Gamma, \Pi_A B)$ and $a \in \mathsf{Elem}(\Gamma, A)$
- $\lambda b \in \mathsf{Elem}(\Gamma, \Pi_A B)$ where $b \in \mathsf{Elem}(\Gamma.A, B)$
- $\forall_A B \in \mathsf{Elem}(\Gamma, \mathsf{Prop})$ where $A \in \mathsf{Type}_n(\Gamma)$ and $B \in \mathsf{Elem}(\Gamma.A, \mathsf{Prop})$
- $\mathsf{un}(t) \in \mathsf{Elem}(\Gamma, \Pi_{x:A}\mathsf{T}(B))$ where $t \in \mathsf{Elem}(\Gamma, \mathsf{T}(\forall_{x:A} B))$
- $\mathsf{mk}(t) \in \mathsf{Elem}(\Gamma, \mathsf{T}(\forall_{x:A} B))$ where $t \in \mathsf{Elem}(\Gamma, \Pi_{x:A}\mathsf{T}(B))$
- $\mathsf{Bool} \in \mathsf{Elem}(\Gamma, \mathsf{Prop})$
- $\mathsf{false}, \mathsf{true} \in \mathsf{Elem}(\Gamma, \mathsf{T}(\mathsf{Bool}))$
- $\mathsf{elim}_{\mathsf{Bool}}(A, a_0, a_1, t) \in \mathsf{Elem}(\Gamma, A(\mathsf{id}, t))$ where $A \in \mathsf{Type}_n(\Gamma.\mathsf{T}(\mathsf{Bool}))$, $a_0 \in \mathsf{Elem}(\Gamma, A(\mathsf{id}, \mathsf{false}))$, $a_1 \in \mathsf{Elem}(\Gamma, A(\mathsf{id}, \mathsf{true}))$, and $t \in \mathsf{Elem}(\Gamma, \mathsf{T}(\mathsf{Bool}))$

We have the following equations:

$$\mathsf{app}(\lambda b, a) = b(\mathsf{id}, a) \qquad \lambda(\mathsf{app}(t\mathsf{p}, \mathsf{q})) = t \qquad \mathsf{un}(\mathsf{mk}(t)) = t \qquad \mathsf{mk}(\mathsf{un}(t)) = t$$

$$\mathsf{elim}_{\mathsf{Bool}}(A, a_0, a_1, \mathsf{false}) = a_0 \qquad \mathsf{elim}_{\mathsf{Bool}}(A, a_0, a_1, \mathsf{true}) = a_1$$

All the type and term formers also have naturality laws:

$$\mathsf{Prop}\ \gamma = \mathsf{Prop} \qquad \mathsf{T}(A)\gamma = \mathsf{T}(A\gamma)$$

$$(\Pi_A B)\gamma = \Pi_{A\gamma} B\gamma^+ \qquad (\lambda b)\gamma = \lambda(b\gamma^+) \qquad \mathsf{app}(t, a)\gamma = \mathsf{app}(t\gamma, a\gamma)$$

$$(\forall_A B)\gamma = \forall_{A\gamma} B\gamma^+ \qquad \mathsf{un}(t)\gamma = \mathsf{un}(t\gamma) \qquad \mathsf{mk}(t)\gamma = \mathsf{mk}(t\gamma)$$

$$\mathsf{Bool}\ \gamma = \mathsf{Bool} \qquad \mathsf{false}\ \gamma = \mathsf{false} \qquad \mathsf{true}\ \gamma = \mathsf{true}$$

$$\mathsf{elim}_{\mathsf{Bool}}(A, a_0, a_1, t)\gamma = \mathsf{elim}_{\mathsf{Bool}}(A\gamma^+, a_0\gamma, a_1\gamma, t\gamma)$$

We write $A \to B$ for $\Pi_A B\mathrm{p}$ and $A \Rightarrow B$ for $\forall_A B\mathrm{p}$.

2.3 Artin gluing

We prove canonicity and normalization using Artin gluing, which is a technique for building a new model of type theory given two models and a pseudomorphism between them [10]. A pseudomorphism is a morphism between the CwFs but with the mapping of contexts only preserved up to isomorphism.

For canonicity, we use a pseudomorphism between the term model and the D-set model defined in chapter 3. For normalization, we use a pseudomophism between the term model and a suitable presheaf model defined in chapter 5 following the technique by Coquand [4].

3

D-set model

We first construct a D-set model [13] similarly to the ω-set model defined by Hofmann [9], which will be used as the target model for Artin gluing.

3.1 Combinatory algebra

We assume a total combinatory algebra, which can be thought of as the syntax of the untyped lambda calculus. It consists of a set D with a left associative application operator $\cdot \in D \times D \to D$ and two elements $\mathsf{k}, \mathsf{s} \in D$ such that:

$$\mathsf{k} \cdot x \cdot y = x \qquad\qquad \mathsf{s} \cdot x \cdot y \cdot z = x \cdot z \cdot (y \cdot z)$$

We also assume that it is a non-trivial model, so that $\mathsf{k} \neq \mathsf{s}$. Other elements and operations on D can be defined in terms of k and s (e.g. by translation from lambda calculus [6]):

$$\begin{aligned}
\mathsf{i} \cdot x &= x \\
\mathsf{b} \cdot x \cdot y \cdot z &= x \cdot (y \cdot z) \\
\langle x, y \rangle \cdot z &= z \cdot x \cdot y \\
\pi_1 \cdot \langle x, y \rangle &= x \\
\pi_2 \cdot \langle x, y \rangle &= y \\
\langle\!\langle x, y \rangle\!\rangle \cdot z &= \langle x \cdot z, y \cdot z \rangle \\
\lambda x \cdot y \cdot z &= x \cdot \langle y, z \rangle
\end{aligned}$$

Note that D does not have to have decidable equality.

3.2 Category with families

A D-set Γ is a pair consisting of a set $|\Gamma|$ and a realization relation $\Vdash_\Gamma \subseteq D \times |\Gamma|$, such that any $\gamma \in |\Gamma|$ has an $x \in D$ where $x \Vdash_\Gamma \gamma$ holds. We say that x realizes γ. Let $D\text{-set}_n$ be a subset of D-set where the first component is in $\mathcal{U}_n$.

If A is a set in $\mathcal{U}_n$, we have a $\mathbf{\Delta}(A)$ in $D\text{-set}_n$ where $|\mathbf{\Delta}(A)| = A$ and $x \Vdash_{\mathbf{\Delta}(A)} a$ holds for all x and a.

The components of the model are defined as follows:

- A context is a D-set.
- A substitution $\gamma \in \Delta \to \Gamma$ is a function $\gamma \in |\Delta| \to |\Gamma|$, such that there is a realizer $x \in D$ where $y \Vdash_\Delta \delta$ implies $x \cdot y \Vdash_\Gamma \gamma(\delta)$.
- $A \in \mathsf{Type}_n(\Gamma)$ is a family A_γ in $D\text{-set}_n$ over $\gamma \in |\Gamma|$.
- $a \in \mathsf{Elem}(\Gamma, A)$ is $a \in \prod_{\gamma \in |\Gamma|} |A_\gamma|$, such that there is a realizer $x \in D$ where $y \Vdash_\Gamma \gamma$ implies $x \cdot y \Vdash_{A_\gamma} a(\gamma)$.
- id is defined by $\mathsf{id}(\gamma) = \gamma$ with the realizer i.
- Composition $\gamma\delta$ is defined by $(\gamma\delta)(\theta) = \theta$ with the realizer $\mathsf{b} \cdot x \cdot y$ where x realizes γ and y realizes δ.
- The terminal context is defined as $\Delta\{\star\}$.
- The substitution $()$ is defined by $()(\gamma) = \star$.
- $A\gamma$ is defined by $(A\gamma)_\delta = A_{\gamma(\delta)}$.
- $a\gamma$ is defined by $(a\gamma)(\delta) = a(\gamma(\delta))$ with the realizer $\mathsf{b} \cdot x \cdot y$ where x realizes a and y realizes γ.
- $\Gamma.A$ is defined by $|\Gamma.A| = \sum_{\gamma \in |\Gamma|} |A_\gamma|$, and $x \Vdash_{\Gamma.A} (\gamma, a)$ iff $\pi_1 \cdot x \Vdash_\Gamma \gamma$ and $\pi_2 \cdot x \Vdash_{A_\gamma} a$, any $(\gamma, a) \in |\Gamma.A|$ has the realizer $\langle x, y \rangle$ where $x \Vdash_\Gamma \gamma$ and $y \Vdash_{A_\gamma} a$.
- p is defined by $\mathsf{p}(\gamma, a) = \gamma$ with the realizer π_1.
- q is defined by $\mathsf{q}(\gamma, a) = a$ with the realizer π_2.
- (γ, a) is defined by $(\gamma, a)(\delta) = (\gamma(\delta), a(\delta))$ with the realizer $\langle\!\langle x, y \rangle\!\rangle$ where x realizes γ and y realizes a.

3.3 Type formers

We can define type formers in the model:

- U_n is defined as $\mathbf{\Delta}(D\text{-set}_n)$.
- $\Pi_A B$ is defined by $|(\Pi_A B)_\gamma| = \mathsf{Elem}(A_\gamma, a.\, B_{(\gamma,a)})$, and $x \Vdash_{(\Pi_A B)_\gamma} b$ iff $y \Vdash_{A_\gamma} a$ implies $x \cdot y \Vdash_{B_{(\gamma,a)}} b(a)$.
- $\mathsf{app}(t, a)$ is defined by $\mathsf{app}(t, a)(\gamma) = t(\gamma)(a(\gamma))$ with the realizer $\mathsf{s} \cdot x \cdot y$ where x realizes t and y realizes a
- λb is defined by $(\lambda b)(\gamma)(a) = b(\gamma, a)$ where $(\lambda b)(\gamma)$ has the realizer $\lambda x \cdot y$ where x realizes b and $y \Vdash_\Gamma \gamma$, and λb has the realizer λx where x realizes b.
- $\Sigma_A B \in \mathsf{Type}_n(\Gamma)$ where $A \in \mathsf{Type}_n(\Gamma)$ and $B \in \mathsf{Type}_n(\Gamma.A)$, it is defined by $(\Sigma_A B)_\gamma = A_\gamma.C$ where $C_a = B_{(\gamma,a)}$.
- $\mathsf{fst}(t) \in \mathsf{Elem}(\Gamma, A)$ where $t \in \mathsf{Elem}(\Gamma, \Sigma_A B)$, is defined by $\mathsf{fst}(t)(\gamma) = t(\gamma).1$ with the realizer $\mathsf{b} \cdot \pi_1 \cdot x$ where x realizes t.
- $\mathsf{snd}(t) \in \mathsf{Elem}(\Gamma, B(\mathsf{id}, \mathsf{fst}(t)))$ where $t \in \mathsf{Elem}(\Gamma, \Sigma_A B)$, is defined by $\mathsf{snd}(t)(\gamma) = t(\gamma).2$ with the realizer $\mathsf{b} \cdot \pi_2 \cdot x$ where x realizes t.
- $(a, b) \in \mathsf{Elem}(\Gamma, \Sigma_A B)$ where $a \in \mathsf{Elem}(\Gamma, A)$ and $b \in \mathsf{Elem}(\Gamma, B(\mathsf{id}, a))$, is defined by $(a, b)(\gamma) = (a(\gamma), b(\gamma))$ with the realizer $\langle\!\langle x, y \rangle\!\rangle$ where x realizes a and y realizes b.
- $\top \in \mathsf{Type}_0(\Gamma)$ is defined by $\top_\gamma = ()$.
- $\mathsf{tt} \in \mathsf{Elem}(\Gamma, \top)$ is defined by $\mathsf{tt}(\gamma) = \star$.

We can also define Booleans in Type_0:

- $\mathsf{Bool}_0 \in \mathsf{Type}_0(\Gamma)$ is defined by $|\mathsf{Bool}_{0_\gamma}| = \{0, 1\}$, and $x \Vdash_{\mathsf{Bool}_{0_\gamma}} 0$ iff $x = \bar{0}$ and $x \Vdash_{\mathsf{Bool}_{0_\gamma}} 1$ iff $x = \bar{1}$, where $\bar{0} = \mathsf{k}$ and $\bar{1} = \mathsf{k} \cdot \mathsf{i}$ as Church encoded Booleans.
- $\mathsf{false}_0 \in \mathsf{Elem}(\Gamma, \mathsf{Bool}_0)$ is defined by $\mathsf{false}_0(\gamma) = 0$ with the realizer $\mathsf{k} \cdot \bar{0}$.
- $\mathsf{true}_0 \in \mathsf{Elem}(\Gamma, \mathsf{Bool}_0)$ is defined by $\mathsf{true}_0(\gamma) = 1$ with the realizer $\mathsf{k} \cdot \bar{1}$.

- $\mathsf{elim}_{\mathsf{Bool}_0}(A, a_0, a_1, t) \in \mathsf{Elem}(\Gamma, A(\mathsf{id}, t))$, where $A \in \mathsf{Type}_n(\Gamma.\mathsf{Bool}_0)$, $a_0 \in \mathsf{Elem}(\Gamma, A(\mathsf{id}, \mathsf{false}_0))$, $a_1 \in \mathsf{Elem}(\Gamma, A(\mathsf{id}, \mathsf{true}_0))$, and $t \in \mathsf{Elem}(\Gamma, \mathsf{Bool}_0)$, is defined by $\mathsf{elim}_{\mathsf{Bool}_0}(A, a_0, a_1, t)(\gamma) = \begin{cases} a_0 & \text{if } t(\gamma) = 0 \\ a_1 & \text{if } t(\gamma) = 1 \end{cases}$ with the realizer $\mathsf{s} \cdot (\mathsf{s} \cdot x \cdot y) \cdot z$ where x realizes t, y realizes a_0, and z realizes a_1.

3.4 Impredicative universe

We call a D-set Γ modest iff $x \Vdash_\Gamma \gamma_1$ and $x \Vdash_\Gamma \gamma_2$ implies $\gamma_1 = \gamma_2$. An $A \in \mathsf{Type}_n(\Gamma)$ is modest iff A_γ is modest for any $\gamma \in \Gamma$. We have that $\Pi_A B$ is modest if B is modest, and Bool_0 is modest because the combinatory algebra is non-trivial.

Let $\mathsf{PER}(A)$ be the set of symmetric and transitive relations on the set A.

We can then define the impredicative universe Prop by $\mathsf{Prop}_\gamma = \mathbf{\Delta}(\mathsf{PER}(D))$.

$\mathsf{T}(A)$ is defined by $|\mathsf{T}(A)_\gamma| = \{x \in D : A(\gamma)(x, x)\}/A(\gamma)$ and $x \Vdash_{\mathsf{T}(A)_\gamma} S$ iff $[x]_{A(\gamma)} = S$. We have that $\mathsf{T}(A)$ is modest for any A.

We can define an operation $[A] \in \mathsf{Elem}(\Gamma, \mathsf{Prop})$ where $A \in \mathsf{Type}_n(\Gamma)$ and is modest, it is defined by the PER: $[A](\gamma)(x, y)$ iff there is an $a \in |A_\gamma|$ such that $x \Vdash_{A_\gamma} a$ and $y \Vdash_{A_\gamma} a$.

We have an isomorphism between $\mathsf{Elem}(\Gamma, \mathsf{T}[A])$ and $\mathsf{Elem}(\Gamma, A)$ for any modest A:

- $\mathsf{un}(t) \in \mathsf{Elem}(\Gamma, A)$ where $t \in \mathsf{Elem}(\Gamma, \mathsf{T}[A])$, is defined by $\mathsf{un}(t)(\gamma) = a$ where $a \in |A_\gamma|$ is the unique element such that for any $x \in D$ where $[x]_{[A](\gamma)} = t(\gamma)$ we have $x \Vdash_{A_\gamma} a$. The realizer of $\mathsf{un}(t)$ is x where x realizes t.
- $\mathsf{mk}(a) \in \mathsf{Elem}(\Gamma, \mathsf{T}[A])$ where $a \in \mathsf{Elem}(\Gamma, A)$, is defined by $\mathsf{mk}(a)(\gamma) = [x]_{[A](\gamma)}$ where x realizes $a(\gamma)$, with the realizer x.

Then $\forall_A B$ can be defined as $[\Pi_A \mathsf{T}(B)]$, and Bool can be defined as $[\mathsf{Bool}_0]$.

For the proof of canonicity, we use subsets. $\{a \in A : P(a)\} \in \mathsf{Type}_n(\Gamma)$ where $A \in \mathsf{Type}_n(\Gamma)$ and P is a predicate over $|A_\gamma|$ for any $\gamma \in |\Gamma|$, is defined by $|\{a \in A : P(a)\}_\gamma| = \{a \in |A_\gamma| : P(a)\}$ and $x \Vdash_{\{a \in A : P(a)\}_\gamma} a$ iff $x \Vdash_{A_\gamma} a$. And $\{a \in A : P(a)\}$ is modest if A is modest.

We define a special syntax for function space if the domain is a set. $\prod_{a \in A} B(a) \in \mathsf{Type}_n(\Gamma)$ where A is a set and $B(a) \in \mathsf{Type}_n(\Gamma)$, is defined as $\Pi_{\mathbf{\Delta}(A)} C$ where $C_{(\gamma,a)} = B(a)_\gamma$. Likewise in Prop, $\prod_{a \in A} B(a) \in \mathsf{Elem}(\Gamma, \mathsf{Prop})$ where A is a set and $B(a) \in \mathsf{Elem}(\Gamma, \mathsf{Prop})$, is defined as $\forall_{\mathbf{\Delta}(A)} C$ where $C_{(\gamma,a)} = B(a)_\gamma$.

4

Canonicity

Assuming a model $\mathscr{M}$, we can construct a canonicity model $\mathscr{M}^*$ by gluing over a pseudomorphism from $\mathscr{M}$ to the D-set model, the pseudomorphism is defined by:

$$|\Gamma| = \mathbf{\Delta}(() \to \Gamma) \qquad |\gamma|(\delta) = \gamma\delta \qquad |A|_\gamma = \mathbf{\Delta}(\mathsf{Elem}((), A\gamma)) \qquad |a|(\gamma) = a\gamma$$

We have isomorphisms between $|()|$ and $()$, and between $|\Gamma.A|$ and $|\Gamma|.|A|$.

4.1 Category with families

Every sort in the resulting model is defined to be dependent pairs, and the first components of the operations are defined to be the corresponding syntactic operations. If we only consider the second component, it can be seen as a displayed model. We list the components as follows:

- A context Γ in $\mathscr{M}^*$ is a pair consisting of a context Γ in $\mathscr{M}$ and a family Γ'_γ of D-sets over $\gamma \in () \to \Gamma$.
- A substitution $\gamma \in \Delta \to^* \Gamma$ is a pair consisting of $\gamma \in \Delta \to \Gamma$ and $\gamma'_\delta \in \Delta'_\delta \to \Gamma'_{\gamma\delta}$ where $\delta \in () \to \Delta$.
- $A \in \mathsf{Type}^*_n(\Gamma)$ is a pair consisting of $A \in \mathsf{Type}_n(\Gamma)$ and $A'_{\gamma,a} \in \mathsf{Type}_n(\Gamma'_\gamma)$ where $\gamma \in () \to \Gamma$ and $a \in \mathsf{Elem}((), A\gamma)$.
- $a \in \mathsf{Elem}^*(\Gamma, A)$ is a pair consisting of $a \in \mathsf{Elem}(\Gamma, A)$ and $a'_\gamma \in \mathsf{Elem}(\Gamma'_\gamma, A'_{\gamma,a\gamma})$ where $\gamma \in () \to \Gamma$.

In the following definitions of operations in this model, the first components are omitted.

- id is defined by $\mathsf{id}'_\gamma = \mathsf{id}$.
- Composition $\gamma\delta$ is defined by $(\gamma\delta)'_\theta = \gamma'_{\delta\theta}\delta'_\theta$.
- The terminal context is defined by $()'_\gamma = ()$.
- The substitution $()$ is defined by $()'_\gamma = ()$.
- $A\gamma$ is defined by $(A\gamma)'_{\delta,a} = A'_{\gamma\delta,a}\gamma'_\delta$.
- $a\gamma$ is defined by $(a\gamma)'_\delta = a'_{\gamma\delta}\gamma'_\delta$.
- $\Gamma.A$ is defined by $(\Gamma.A)'_\gamma = \Gamma'_{\mathsf{p}\gamma}.A'_{\mathsf{p}\gamma,\mathsf{q}\gamma}$.
- p is defined by $\mathsf{p}'_\gamma = \mathsf{p}$.
- q is defined by $\mathsf{q}'_\gamma = \mathsf{q}$.
- (γ, a) is defined by $(\gamma, a)'_\delta = (\gamma'_\delta, a'_\delta)$.

4.2 Type formers

For the type formers we have:

- U_n is defined by $\mathsf{U}_{n\gamma,A} = \mathbf{\Delta}(\mathsf{Elem}((), A)) \to \mathsf{U}_n$.
- $\Pi_A B$ is defined by $(\Pi_A B)'_{\gamma,t} = \prod_{a\in\mathsf{Elem}((),A\gamma)} \Pi_{A'_{\gamma,a}} B'_{(\gamma,a),\mathsf{app}(t,a)}$.
- $\mathsf{app}(t, a)$ is defined by $\mathsf{app}(t, a)'_\gamma = \mathsf{app}(t'_\gamma(a\gamma), a'_\gamma)$.
- λb is defined by $(\lambda b)'_\gamma(a) = \lambda b'_{(\gamma,a)}$.
- Prop is defined by $\mathsf{Prop}'_{\gamma,A} = \mathbf{\Delta}(\mathsf{Elem}((), \mathsf{T}(A))) \to \mathsf{Prop}$
- $\mathsf{T}(A)$ is defined by $\mathsf{T}(A)'_{\gamma,a} = \mathsf{T}(A'_\gamma(a))$.
- $\forall_A B$ is defined by $(\forall_A B)'_\gamma(t) = \prod_{a\in\mathsf{Elem}((),A\gamma)} \forall_{A'_{\gamma,a}} B'_{(\gamma,a)}(\mathsf{app}(t, a))$.
- Bool is defined by $\mathsf{Bool}'_\gamma(t) = [\{u \in \mathsf{Bool}_0 \,:\, (t = \mathsf{false} \wedge u = 0) \vee (t = \mathsf{true} \wedge u = 1)\}]$.

- false is defined by $\mathsf{false}'_\gamma = \mathsf{mk}(\mathsf{false}_0)$.
- true is defined by $\mathsf{true}'_\gamma = \mathsf{mk}(\mathsf{true}_0)$.
- $\mathsf{elim}_{\mathsf{Bool}}(A, a_0, a_1, t)$ is defined by $\mathsf{elim}_{\mathsf{Bool}}(A, a_0, a_1, t)'_\gamma = \mathsf{elim}_{\mathsf{Bool}_0}(A'_{(\gamma,t\gamma),\mathsf{elim}_{\mathsf{Bool}}(A\gamma,a_0\gamma,a_1\gamma,t\gamma)}(\mathsf{p}, \mathsf{mk}(\mathsf{q})), a_0{}'_\gamma, a_1{}'_\gamma, \mathsf{un}(t'_\gamma))$

We can prove that there is an isomorphism between $\mathsf{Elem}(\Gamma, \mathsf{T}(\forall_A B))$ and $\mathsf{Elem}(\Gamma, \Pi_A \mathsf{T}(B))$.

4.3 Result

Instantiating $\mathscr{M}$ to be the initial model $\mathscr{M}_0$, we have a homomorphism from $\mathscr{M}_0$ to $\mathscr{M}_0^*$. For any $t \in \mathsf{Elem}((), \mathsf{T}(\mathsf{Bool}))$ we have $t^* \in \mathsf{Elem}^*((), \mathsf{T}(\mathsf{Bool}))$, where the first component of t^* is defined to be equal to t, and we have ${t^*}'_() \in \mathsf{Elem}((), \mathsf{T}([\{u \in \mathsf{Bool}_0 : (t = \mathsf{false} \wedge u = 0) \vee (t = \mathsf{true} \wedge u = 1)\}]))$ and $\mathsf{un}({t^*}'_())(\star) \in \{u \in \{0, 1\} : (t = \mathsf{false} \wedge u = 0) \vee (t = \mathsf{true} \wedge u = 1)\}$, which implies $t = \mathsf{false}$ or $t = \mathsf{true}$, and $\mathsf{un}(\mathsf{false}'_())(\star) = 0 \neq 1 = \mathsf{un}(\mathsf{true}'_())(\star)$ therefore $\mathsf{false} \neq \mathsf{true}$.

5

D-presheaf model

To prove normalization, we construct a presheaf model valued in D-sets over some category $\mathscr{C}$. We call such a presheaf D-presheaf.

5.1 Category with families

The components of the model are defined as follows:

- A context Γ is a pair consisting of a family Γ_X of D-sets over object X in $\mathscr{C}$, and a restriction operation $\Gamma_f \in \Gamma_X \to \Gamma_Y$ where $f \in Y \to X$, such that $\Gamma_{\mathsf{id}} = \mathsf{id}$ and $\Gamma_{fg} = \Gamma_g\, \Gamma_f$.
- A substitution $\gamma \in \Delta \to \Gamma$ is a D-set substitution $\gamma_X \in \Delta_X \to \Gamma_X$ for any X, such that $\gamma_Y\, \Delta_f = \Gamma_f\, \gamma_X$.
- $A \in \mathsf{Type}_n(\Gamma)$ is a pair consisting of $A_X \in \mathsf{Type}_n(\Gamma_X)$ where X is an object in $\mathscr{C}$, that is, a family $A_{X,\gamma}$ of D-set$_n$ over object X in $\mathscr{C}$ and $\gamma \in |\Gamma_X|$, and a restriction operation $A_f \in \mathsf{Elem}(\Gamma_X.A_X, A_Y\, \Gamma_f\, \mathsf{p})$ where $f \in Y \to X$, such that $A_{\mathsf{id}} = \mathsf{q}$ and $A_{fg} = A_g\, (\Gamma_f\, \mathsf{p}, A_f)$.
- $a \in \mathsf{Elem}(\Gamma, A)$ is $a_X \in \mathsf{Elem}(\Gamma_X, A_X)$ for any X, such that $a_Y\, \Gamma_f = A_f\, (\mathsf{id}, a_X)$.
- id is defined by $\mathsf{id}_X = \mathsf{id}$.
- Composition $\gamma\delta$ is defined by $(\gamma\delta)_X = \gamma_X\, \delta_X$.
- The terminal context is defined by $()_X = ()$ and $()_f = ()$.
- The substitution $()$ is defined by $()_X = ()$.
- $A\gamma$ is defined by $(A\gamma)_X = A_X\, \gamma_X$ and $(A\gamma)_f = A_f\, \gamma_X^+$.

- $a\gamma$ is defined by $(a\gamma)_X = a_X\,\gamma_X$.
- $\Gamma.A$ is defined by $(\Gamma.A)_X = \Gamma_X.A_X$ and $(\Gamma.A)_f = (\Gamma_f\,\mathsf{p}, A_f)$.
- p is defined by $\mathsf{p}_X = \mathsf{p}$.
- q is defined by $\mathsf{q}_X = \mathsf{q}$.
- (γ, a) is defined by $(\gamma, a)_X = (\gamma_X, a_X)$.

5.2 Type formers

For the type formers, we first define some operations related to Yoneda embedding:

- $\mathsf{y}X$ is a context if X is an object in $\mathscr{C}$, it is defined by $(\mathsf{y}X)_Y = \mathbf{\Delta}(Y \to X)$ and $(\mathsf{y}X)_g(f) = fg$.
- $\mathsf{y}f \in \mathsf{y}Y \to \mathsf{y}X$ where $f \in Y \to X$, is defined by $(\mathsf{y}f)_X(g) = fg$.
- $\hat{\gamma} \in \mathsf{y}X \to \Gamma$ where $\gamma \in |\Gamma_X|$, is defined by $\hat{\gamma}_Y(f) = \Gamma_f(\gamma)$.

We can then define the type formers in the model:

- U_n is defined by $(\mathsf{U}_n)_{X,\gamma} = \mathbf{\Delta}(\mathsf{Type}_n(\mathsf{y}X))$.
- $\Pi_A B$ is defined by $|(\Pi_A B)_{X,\gamma}| = \mathsf{Elem}(\mathsf{y}X.A\hat{\gamma}, B\hat{\gamma}^+)$, $x \Vdash_{(\Pi_A B)_{X,\gamma}} b$ iff $y \Vdash_{A_{Y,\Gamma_f(\gamma)}} a$ implies $x \cdot y \Vdash_{B_{Y,(\Gamma_f(\gamma),a)}} b_Y(f,a)$, and $(\Pi_A B)_f(\gamma, b) = b(\mathsf{y}f)^+$ with the realizer π_2.
- $\mathsf{app}(t,a)$ is defined by $\mathsf{app}(t,a)_X(\gamma) = t_X(\gamma)_X(\mathsf{id}, a_X(\gamma))$ with the realizer $\mathsf{s}\cdot x \cdot y$ where x realizes f_X and y realizes a_X.
- λb is defined by $(\lambda b)_X(\gamma) = b\hat{\gamma}^+$ with the realizer λx where x realizes b.
- $\Sigma_A B \in \mathsf{Type}_n(\Gamma)$ where $A \in \mathsf{Type}_n(\Gamma)$ and $B \in \mathsf{Type}_n(\Gamma.A)$, it is defined by $(\Sigma_A B)_X = \Sigma_{A_X} B_X$ and $(\Sigma_A B)_f = (A_f(\mathsf{p}, \mathsf{fst}(\mathsf{q})), B_f(\mathsf{p}, \mathsf{snd}(\mathsf{q})))$.
- $\mathsf{fst}(t) \in \mathsf{Elem}(\Gamma, A)$ where $t \in \mathsf{Elem}(\Gamma, \Sigma_A B)$, is defined by $\mathsf{fst}(t)_X = \mathsf{fst}(t_X)$.
- $\mathsf{snd}(t) \in \mathsf{Elem}(\Gamma, B(\mathsf{id}, \mathsf{fst}(t)))$ where $t \in \mathsf{Elem}(\Gamma, \Sigma_A B)$, is defined by $\mathsf{snd}(t)_X = \mathsf{snd}(t_X)$.

- $(a, b) \in \mathsf{Elem}(\Gamma, \Sigma_A B)$ where $a \in \mathsf{Elem}(\Gamma, A)$ and $b \in \mathsf{Elem}(\Gamma, B(\mathsf{id}, a))$, is defined by $(a, b)_X = (a_X, b_X)$.
- $\top \in \mathsf{Type}_0(\Gamma)$ is defined by $\top_X = \top$ and $\top_f = \mathsf{tt}$.
- $\mathsf{tt} \in \mathsf{Elem}(\Gamma, \top)$ is defined by $\mathsf{tt}_X = \mathsf{tt}$.

We can define Booleans in Type_0:

- $\mathsf{Bool}_0 \in \mathsf{Type}_0(\Gamma)$ is defined by $\mathsf{Bool}_X = \mathsf{Bool}_0$ and $\mathsf{Bool}_f = \mathsf{q}$.
- $\mathsf{false}_0 \in \mathsf{Elem}(\Gamma, \mathsf{Bool}_0)$ is defined by $\mathsf{false}_X = \mathsf{false}_0$.
- $\mathsf{true}_0 \in \mathsf{Elem}(\Gamma, \mathsf{Bool}_0)$ is defined by $\mathsf{true}_X = \mathsf{true}_0$.
- $\mathsf{elim}_{\mathsf{Bool}_0}(A, a_0, a_1, t) \in \mathsf{Elem}(\Gamma, A(\mathsf{id}, t))$, where $A \in \mathsf{Type}_n(\Gamma.\mathsf{Bool}_0)$, $a_0 \in \mathsf{Elem}(\Gamma, A(\mathsf{id}, \mathsf{false}_0))$, $a_1 \in \mathsf{Elem}(\Gamma, A(\mathsf{id}, \mathsf{true}_0))$, and $t \in \mathsf{Elem}(\Gamma, \mathsf{Bool}_0)$, is defined by $\mathsf{elim}_{\mathsf{Bool}_0}(A, a_0, a_1, t)_X = \mathsf{elim}_{\mathsf{Bool}_0}(A_X, (a_0)_X, (a_1)_X, t_X)$.

5.3 Impredicative universe

We call an $A \in \mathsf{Type}_n(\Gamma)$ modest iff A_X is modest for any X. We have that $\Pi_A B$ is modest if B is modest, and Bool_0 is modest.

We define $R \in \mathsf{PER}(A, \Gamma)$, where A is a set and Γ is a D-presheaf, to be a family $R_{X,\gamma} \in \mathsf{PER}(A)$ where $\gamma \in |\Gamma_X|$, such that $R_{X,\gamma}(a_1, a_2)$ implies $R_{Y,\Gamma_f(\gamma)}(a_1, a_2)$ for any $f \in Y \to X$. We also have $R\gamma \in \mathsf{PER}(A, \Delta)$, where $R \in \mathsf{PER}(A, \Gamma)$ and $\gamma \in \Delta \to \Gamma$, defined by $(R\gamma)_{X,\delta} = R_{X,\gamma_X(\delta)}$.

We then define $\mathsf{Prop} \in \mathsf{Type}_0(\Gamma)$ by $\mathsf{Prop}_{X,\gamma} = \mathbf{\Delta}(\mathsf{PER}(D, \mathsf{y}X))$ and $\mathsf{Prop}_f(\gamma, A) = A(\mathsf{y}f)$.

$\mathsf{T}(A)$ is defined by $|\mathsf{T}(A)_{X,\gamma}| = \{x \in D : A_X(\gamma)_{X,1}(x, x)\}/A_X(\gamma)_{X,1}$, $x \Vdash_{\mathsf{T}(A)_{X,\gamma}} S$ iff $[x]_{A_X(\gamma)_{X,1}} = S$, and $\mathsf{T}(A)_f(\gamma, S) = [x]_{A_X(\gamma)_{Y,f}}$ with $x \in S$. We have that $\mathsf{T}(A)$ is modest for any A.

We define $[A] \in \mathsf{Elem}(\Gamma, \mathsf{Prop})$ where $A \in \mathsf{Type}_n(\Gamma)$ and is modest, it is defined by $[A]_X(\gamma)_{Y,f} = [A_Y](\Gamma_f(\gamma))$.

We have an isomorphism between $\mathsf{Elem}(\Gamma, \mathsf{T}[A])$ and $\mathsf{Elem}(\Gamma, A)$ for any modest A:

- $\mathsf{un}(t) \in \mathsf{Elem}(\Gamma, A)$ where $t \in \mathsf{Elem}(\Gamma, \mathsf{T}[A])$, is defined by $\mathsf{un}(t)_X(\gamma) = a$ where $a \in |A_{X,\gamma}|$ is the unique element such that for any $x \in D$ where $[x]_{[A_X](\gamma)} = t_X(\gamma)$ we have $x \Vdash_{A_{X,\gamma}} a$. The realizer of $\mathsf{un}(t)_X$ is x where x realizes t.
- $\mathsf{mk}(a) \in \mathsf{Elem}(\Gamma, \mathsf{T}[A])$ where $a \in \mathsf{Elem}(\Gamma, A)$, is defined by $\mathsf{mk}(a)_X(\gamma) = [x]_{[A_X](\gamma)}$ where x realizes $a_X(\gamma)$ with the realizer x.

Then $\forall_A B$ can be defined as $[\Pi_A \mathsf{T}(B)]$, and Bool can be defined as $[\mathsf{Bool}_0]$.

6

Normalization

To prove normalization, we first define the category of telescopes, assuming a model $\mathscr{M}$.

6.1 Category of telescopes

We define telescopes inductively with:

- $()$ is a telescope
- $X.A$ is a telescope where X is a telescope and $A \in \mathsf{Type}\langle X \rangle$

together with the operation $\langle X \rangle$ which is a context in $\mathscr{M}$ for a telescope X, defined by:

$$\langle () \rangle = () \qquad\qquad \langle X.A \rangle = \langle X \rangle.A$$

We inductively define weakenings as the morphisms $Y \to X$ where X and Y are telescopes, with:

- $() \in () \to ()$
- $f\mathsf{p} \in Y.A \to X$ where $f \in Y \to X$
- $f^+ \in Y.A\langle f \rangle \to X.A$ where $f \in Y \to X$

together with the operation $\langle f \rangle \in \langle Y \rangle \to \langle X \rangle$ where $f \in Y \to X$, defined by:

$$\langle () \rangle = () \qquad \langle f\mathsf{p} \rangle = \langle f \rangle \mathsf{p} \qquad \langle f^+ \rangle = \langle f \rangle^+$$

$\mathsf{id} \in X \to X$ is defined by:

$$\mathsf{id}_() = () \qquad \mathsf{id}_{X.A} = \mathsf{id}_X^+$$

Composition $fg \in Z \to X$ where $f \in Y \to X$ and $g \in Z \to Y$ is defined by:

$$f() = f \qquad f(g\mathsf{p}) = (fg)\mathsf{p} \qquad (f\mathsf{p})g^+ = (fg)\mathsf{p} \qquad f^+g^+ = (fg)^+$$

6.2 Neutral and normal terms

We define the set of syntactical expressions with

$$\begin{aligned} A, B, a, b, t ::= {} & \mathsf{v}_n \mid \mathsf{U}_n \mid \Pi\ A\ B \mid \mathsf{app}\ A\ B\ t\ a \mid \lambda\ A\ B\ b \\ & \mid \mathsf{Prop} \mid \mathsf{T}\ A \mid \forall\ A\ B \mid \mathsf{un}\ A\ B\ t \mid \mathsf{mk}\ A\ B\ t \\ & \mid \mathsf{Bool} \mid \mathsf{false} \mid \mathsf{true} \mid \mathsf{elim}_{\mathsf{Bool}}\ A\ a_0\ a_1\ t \end{aligned}$$

where $n \in \mathbb{N}$.

We then inductively define $\mathsf{Term}(X, A)$ which is a well-typed subset of syntactical expressions where $A \in \mathsf{Type}_n\langle X \rangle$. Together we have an operation $\langle a \rangle \in \mathsf{Elem}(\langle X \rangle, A)$ where $a \in \mathsf{Term}(X, A)$.

We define mutually inductive sets $\mathsf{Norm}(X, A)$ and $\mathsf{Neut}(X, A)$ to be subsets of $\mathsf{Term}(X, A)$, with:

- $a^N \in \mathsf{Norm}(X, \langle A^N \rangle)$ where $A^N \in \mathsf{Neut}(X, \mathsf{U}_n)$ and $a^N \in \mathsf{Neut}(X, \langle A^N \rangle)$
- $\mathsf{v}_0 \in \mathsf{Neut}(X.A, A\mathsf{p})$
- $\mathsf{v}_{n+1} \in \mathsf{Neut}(X.B, A\mathsf{p})$ where $\mathsf{v}_n \in \mathsf{Neut}(X, A)$
- $\mathsf{U}_n \in \mathsf{Norm}(X, \mathsf{U}_{n+1})$
- $A^N \in \mathsf{Norm}(X, \mathsf{U}_n)$ where $A^N \in \mathsf{Neut}(X, \mathsf{U}_n)$

- $\Pi\ A^N\ B^N \in \mathsf{Norm}(X, \mathsf{U}_n)$ where $A^N \in \mathsf{Norm}(X, \mathsf{U}_n)$ and $B^N \in \mathsf{Norm}(X.\langle A^N\rangle, \mathsf{U}_n)$
- $\mathsf{app}\ A^N\ B^N\ t^N\ a^N \in \mathsf{Neut}(X, \langle B^N\rangle(\mathsf{id}, \langle a^N\rangle))$ where $A^N \in \mathsf{Norm}(X, \mathsf{U}_n)$, $B^N \in \mathsf{Norm}(X.\langle A^N\rangle, \mathsf{U}_n)$, $t^N \in \mathsf{Neut}(X, \Pi_{\langle A^N\rangle}\langle B^N\rangle)$, and $a^N \in \mathsf{Norm}(X, \langle A^N\rangle)$
- $\lambda\ A^N\ B^N\ b^N \in \mathsf{Norm}(X, \Pi_{\langle A^N\rangle}\langle B^N\rangle)$ where $A^N \in \mathsf{Norm}(X, \mathsf{U}_n)$, $B^N \in \mathsf{Norm}(X.\langle A^N\rangle, \mathsf{U}_n)$, and $b^N \in \mathsf{Norm}(X.\langle A^N\rangle, \langle B^N\rangle)$
- $\mathsf{Prop} \in \mathsf{Norm}(X, \mathsf{U}_0)$
- $A^N \in \mathsf{Norm}(X, \mathsf{Prop})$ where $A^N \in \mathsf{Neut}(X, \mathsf{Prop})$
- $\mathsf{T}\ A^N \in \mathsf{Norm}(X, \mathsf{U}_0)$ where $A^N \in \mathsf{Norm}(X, \mathsf{Prop})$
- $\forall\ A^N\ B^N \in \mathsf{Norm}(X, \mathsf{Prop})$ where $A^N \in \mathsf{Norm}(X, \mathsf{U}_n)$ and $B^N \in \mathsf{Norm}(X.\langle A^N\rangle, \mathsf{Prop})$
- $\mathsf{un}\ A^N\ B^N\ t^N \in \mathsf{Neut}(X, \Pi_{\langle A^N\rangle}\mathsf{T}(\langle B^N\rangle))$ where $A^N \in \mathsf{Norm}(X, \mathsf{U}_n)$, $B^N \in \mathsf{Norm}(X.\langle A^N\rangle, \mathsf{Prop})$, and $t^N \in \mathsf{Neut}(X, \mathsf{T}(\forall_{\langle A^N\rangle}\langle B^N\rangle))$
- $\mathsf{mk}\ A^N\ B^N\ t^N \in \mathsf{Norm}(X, \mathsf{T}(\forall_{\langle A^N\rangle}\langle B^N\rangle)))$ where $A^N \in \mathsf{Norm}(X, \mathsf{U}_n)$, $B^N \in \mathsf{Norm}(X.\langle A^N\rangle, \mathsf{Prop})$, and $t^N \in \mathsf{Norm}(X, \Pi_{\langle A^N\rangle}\mathsf{T}(\langle B^N\rangle))$
- $\mathsf{Bool} \in \mathsf{Norm}(X, \mathsf{Prop})$
- $t^N \in \mathsf{Norm}(X, \mathsf{T}(\mathsf{Bool}))$ where $t^N \in \mathsf{Neut}(X, \mathsf{T}(\mathsf{Bool}))$
- $\mathsf{false}, \mathsf{true} \in \mathsf{Norm}(X, \mathsf{T}(\mathsf{Bool}))$
- $\mathsf{elim}_{\mathsf{Bool}}\ A^N\ a_0^N\ a_1^N\ t^N \in \mathsf{Neut}(X, \langle A^N\rangle(\mathsf{id}, \langle t^N\rangle))$ where $A^N \in \mathsf{Norm}(X.\mathsf{T}(\mathsf{Bool}), \mathsf{U}_n)$, $a_0^N \in \mathsf{Norm}(X, \langle A^N\rangle(\mathsf{id}, \mathsf{false}))$, $a_1^N \in \mathsf{Norm}(X, \langle A^N\rangle(\mathsf{id}, \mathsf{true}))$, and $t^N \in \mathsf{Neut}(X, \mathsf{T}(\mathsf{Bool}))$

We can define operations $a^N f \in \mathsf{Neut}(Y, A\langle f\rangle)$ for $a^N \in \mathsf{Neut}(X, A)$ and $f \in Y \to X$, and $a^N f \in \mathsf{Norm}(Y, A\langle f\rangle)$ for $a^N \in \mathsf{Norm}(X, A)$ and $f \in Y \to X$.

We define $\mathsf{Term}(X, A)$ as a modest D-set, where $|\mathsf{Term}(X, A)| = \mathsf{Term}(X, A)$, and the realizability relation relates Church encoded terms with the syntactical expressions. $\mathsf{Neut}(X, A)$ and $\mathsf{Norm}(X, A)$ as modest D-sets can then be defined as $\{t \in \mathsf{Term}(X, A) : t \in \mathsf{Neut}(X, A)\}$ and $\{t \in \mathsf{Term}(X, A) : t \in \mathsf{Norm}(X, A)\}$.

6.3 Constants

Before constructing the model, we define some constants to be internal to the D-presheaf model, where we instantiate the category $\mathscr{C}$ to be the category of telescopes:

- Type_n is a type defined by $\mathsf{Type}_{n\,X} = \mathbf{\Delta}(\mathsf{Type}_n\langle X\rangle)$ and $\mathsf{Type}_{n\,f}(A) = A\langle f\rangle$.
- $\mathsf{Elem}(A)$ is a type for $A : \mathsf{Type}_n$, defined by $\mathsf{Elem}_X(A) = \mathbf{\Delta}(\mathsf{Elem}(\langle X\rangle, A))$ and $\mathsf{Elem}_f(A, a) = a\langle f\rangle$.
- $|\Gamma|$ is a type for any context Γ in $\mathscr{M}$, is defined by $|\Gamma|_X = \mathbf{\Delta}(\langle X\rangle \to \Gamma)$ and $|\Gamma|_f(\gamma) = \gamma\langle f\rangle$.
- $|\gamma| : |\Delta| \to |\Gamma|$ for $\gamma \in \Delta \to \Gamma$, is defined by $|\gamma|_X(\delta) = \gamma\delta$.
- $|A| : |\Gamma| \to \mathsf{Type}_n$ for $A \in \mathsf{Type}_n(\Gamma)$, is defined by $|A|_X(\gamma) = A\gamma$.
- $|a| : \Pi_{\gamma : |\Gamma|}\mathsf{Elem}(|A|\ \gamma)$ for $a \in \mathsf{Elem}(\Gamma, A)$, is defined by $|a|_X(\gamma) = a\gamma$.
- $\mathsf{fst} : |\Gamma.A| \to |\Gamma|$ is defined by $\mathsf{fst}_X(\gamma) = \mathsf{p}\gamma$.
- $\mathsf{snd} : \Pi_{\gamma : |\Gamma.A|} \to |\Gamma|(\mathsf{fst}\ \gamma)$ is defined by $\mathsf{snd}_X(\gamma) = \mathsf{q}\gamma$.
- $\mathsf{pair} : \Pi_{\gamma : |\Gamma|} \to |A|\ \gamma \to |\Gamma.A|$ is defined by $\mathsf{pair}_X(\gamma, a) = (\gamma, a)$.
- $\mathsf{U}_n : \mathsf{Type}_{n+1}$ is defined by $\mathsf{U}_{n\,X} = \mathsf{U}_n$.
- $\mathsf{app} : \Pi_{\gamma : |\Gamma|}\mathsf{Elem}(|\Pi_A B|\ \gamma) \to \Pi_{a : \mathsf{Elem}(|A|\ \gamma)}\mathsf{Elem}(|B|\ (\mathsf{pair}\ \gamma\ a))$ where $A \in \mathsf{Type}_n(\Gamma)$ and $B \in \mathsf{Type}_n(\Gamma.A)$, is defined by $\mathsf{app}_X(\gamma, t, a) = \mathsf{app}(t, a)$.
- $\mathsf{Prop} : \mathsf{Type}_0$ is defined by $\mathsf{Prop}_X = \mathsf{Prop}$.
- $\mathsf{T} : \mathsf{Elem}(\mathsf{Prop}) \to \mathsf{Type}_0$ where $\mathsf{T}_X(A) = \mathsf{T}(A)$.
- $\mathsf{un} : \Pi_{\gamma : |\Gamma|}\mathsf{Elem}(\mathsf{T}\ (|\forall_A B|\ \gamma)) \to \mathsf{Elem}(|\Pi_A \mathsf{T}(B)|\ \gamma)$ is defined by $\mathsf{un}_X(\gamma, t) = \mathsf{un}(t)$.
- $\mathsf{Bool} : \mathsf{Elem}(\mathsf{Prop})$ is defined by $\mathsf{Bool}_X = \mathsf{Bool}$.
- $\mathsf{false} : \mathsf{Elem}(\mathsf{T}\ \mathsf{Bool})$ is defined by $\mathsf{false}_X = \mathsf{false}$.
- $\mathsf{true} : \mathsf{Elem}(\mathsf{T}\ \mathsf{Bool})$ is defined by $\mathsf{true}_X = \mathsf{true}$.

- $\mathsf{elim}_{\mathsf{Bool}} : \Pi_{\gamma : |\Gamma|}\mathsf{Elem}(|A|\ (\mathsf{pair}\ \gamma\ \mathsf{false})) \to \mathsf{Elem}(|A|\ (\mathsf{pair}\ \gamma\ \mathsf{true})) \to \Pi_{t : \mathsf{Elem}(\mathsf{T}\ \mathsf{Bool})}\mathsf{Elem}(|A|\ (\mathsf{pair}\ \gamma\ t))$ where $A \in \mathsf{Type}_n(\Gamma.\mathsf{T}(\mathsf{Bool}))$, is defined by $\mathsf{elim}_{\mathsf{Bool}\,X}(a_0, a_1, t) = \mathsf{elim}_{\mathsf{Bool}}(A\gamma^+, a_0, a_1, t)$.
- $\mathsf{Neut}(A)$ is a type for $A : \mathsf{Type}_n$, defined by $\mathsf{Neut}_X(A) = \mathsf{Neut}(X, A)$ and $\mathsf{Neut}_f(A, a^N) = a^N f$.
- We write $\mathsf{Norm}(A)|a$ for $A : \mathsf{Type}$ and $a : \mathsf{Elem}(A)$, it is a type defined by $\mathsf{Norm}_X(A)|a = \{a^N \in \mathsf{Norm}(X, A) : \langle a^N \rangle = a\}$ and $(\mathsf{Norm}_X(A)|a)(a^N) = a^N f$
- We write $\langle a^N \rangle : \mathsf{Elem}(A)$ for $a^N : \mathsf{Neut}(A)$ defined by $\langle a^N \rangle_X = \langle a^N \rangle$.

We then define constants for neutral and normal term constructors. For $A \in \mathsf{Type}_n(\Gamma)$ and $B \in \mathsf{Type}_n(\Gamma.A)$, given $\gamma : |\Gamma|$, $A^N : \mathsf{Norm}(\mathsf{U}_n)|(|A|\ \gamma)$, and $B^N : \Pi_{a : \mathsf{Neut}(|A|\ \gamma)}\mathsf{Norm}(\mathsf{U}_n)|(|B|\ (\mathsf{pair}\ \gamma\ \langle a \rangle))$, we define:

- $\Pi^S\ \gamma\ A^N\ B^N : \mathsf{Norm}(\mathsf{U}_n)|(|\Pi_A B|\ \gamma)$ is defined by $\Pi^S_X(\gamma, A^N, B^N) = \Pi\ A^N\ (B^N_{X.\langle A^N \rangle}(\mathsf{p}, \mathsf{v}_0))$.
- $\mathsf{app}^S\ \gamma\ A^N\ B^N : \mathsf{Neut}(|\Pi_A B|\ \gamma) \to \Pi_{a : \mathsf{Elem}(|A|\ \gamma)}\mathsf{Norm}(|A|\ \gamma)|a \to \mathsf{Neut}(|B|\ (\mathsf{pair}\ \gamma\ a))$ is defined by $\mathsf{app}^S_X(\gamma, A^N, B^N, t, a, a^N) = \mathsf{app}\ A^N\ (B^N_{X.\langle A^N \rangle}(\mathsf{p}, \mathsf{v}_0))\ t\ a^N$.
- $\lambda^S\ \gamma\ A^N\ B^N : \Pi_{t : \mathsf{Elem}(|\Pi_A B|\ \gamma)}$ $(\Pi_{a : \mathsf{Neut}(|A|\ \gamma)}\mathsf{Norm}(|B|\ (\mathsf{pair}\ \gamma\ \langle a \rangle))|(\mathsf{app}\ \gamma\ t\ \langle a \rangle)) \to \mathsf{Norm}(|\Pi_A B|\ \gamma)|t$ is defined by $\lambda^S_X(\gamma, A^N, B^N, t, t^N) = \lambda\ A^N\ (B^N_{X.\langle A^N \rangle}(\mathsf{p}, \mathsf{v}_0))\ (t^N_{X.\langle A^N \rangle}(\mathsf{p}, \mathsf{v}_0))$.

The constants for $\forall$ can be defined similarly. For $A \in \mathsf{Type}_n(\Gamma)$ and $B \in \mathsf{Elem}(\Gamma.A, \mathsf{Prop})$, given $\gamma : |\Gamma|$, $A^N : \mathsf{Norm}(\mathsf{U}_n)|(|A|\ \gamma)$, and $B^N : \Pi_{a : \mathsf{Neut}(|A|\ \gamma)}\mathsf{Norm}(\mathsf{Prop})|(|B|\ (\mathsf{pair}\ \gamma\ \langle a \rangle))$, we have:

- $\forall^S\ \gamma\ A^N\ B^N : \mathsf{Norm}(\mathsf{Prop})|(|\forall_A B|\ \gamma)$
- $\mathsf{un}^S\ \gamma\ A^N\ B^N : \mathsf{Neut}(\mathsf{T}\ (|\forall_A B|\ \gamma)) \to \mathsf{Neut}(|\Pi_A \mathsf{T}(B)|\ \gamma)$
- $\mathsf{mk}^S\ \gamma\ A^N\ B^N : \Pi_{t : \mathsf{Elem}(\mathsf{T}\ (|\forall_A B|\ \gamma))}\mathsf{Norm}(|\Pi_A \mathsf{T}(B)|\ \gamma)|(\mathsf{un}\ t) \to \mathsf{Norm}(\mathsf{T}\ (|\forall_A B|\ \gamma))|t$

The other constants are straightforward:

- $\mathsf{ne}^S : \Pi_{A^N : \mathsf{Neut}(\mathsf{U}_n)}\Pi_{a^N : \mathsf{Neut}\langle A^N \rangle}\mathsf{Norm}\langle A^N \rangle|\langle a^N \rangle$

- U_n^S : $\mathsf{Norm}(\mathsf{U}_{n+1})|\mathsf{U}_n$
- $\mathsf{ne}_{\mathsf{U}}^S$: $\Pi_{A^N:\mathsf{Neut}(\mathsf{U}_n)}\mathsf{Norm}(\mathsf{U}_n)|\langle A^N\rangle$
- Prop^S : $\mathsf{Norm}(\mathsf{U}_0)|\mathsf{Prop}$
- $\mathsf{ne}_{\mathsf{Prop}}^S$: $\Pi_{A^N:\mathsf{Neut}(\mathsf{Prop})}\mathsf{Norm}(\mathsf{Prop})|\langle A^N\rangle$
- T^S : $\Pi_{\gamma:|\Gamma|}\mathsf{Norm}(\mathsf{Prop})|(|A|\ \gamma) \to \mathsf{Norm}(\mathsf{U}_0)|(|\mathsf{T}(A)|\ \gamma)$
- Bool^S : $\mathsf{Norm}(\mathsf{Prop})|\mathsf{Bool}$
- $\mathsf{ne}_{\mathsf{Bool}}^S$: $\Pi_{t^N:\mathsf{Neut}(\mathsf{T}\ \mathsf{Bool})}\mathsf{Norm}(\mathsf{T}\ \mathsf{Bool})|\langle t^N\rangle$
- false^S : $\mathsf{Norm}(\mathsf{T}\ \mathsf{Bool})|\mathsf{false}$
- true^S : $\mathsf{Norm}(\mathsf{T}\ \mathsf{Bool})|\mathsf{true}$
- $\mathsf{elim}_{\mathsf{Bool}}^S$: $\Pi_{\gamma:|\Gamma|}(\Pi_{t^N:\mathsf{Neut}(\mathsf{T}\ \mathsf{Bool})} \to \mathsf{Norm}(\mathsf{U}_n)|(|A|\ (\mathsf{pair}\ \gamma\ \langle t^N\rangle))) \to$
  $\Pi_{a_0:\mathsf{Elem}(|A|\ (\mathsf{pair}\ \gamma\ \mathsf{false}))}\mathsf{Norm}(|A|\ (\mathsf{pair}\ \gamma\ \mathsf{false}))|a_0 \to$
  $\Pi_{a_1:\mathsf{Elem}(|A|\ (\mathsf{pair}\ \gamma\ \mathsf{true}))}\mathsf{Norm}(|A|\ (\mathsf{pair}\ \gamma\ \mathsf{true}))|a_1 \to \Pi_{t^N:\mathsf{Neut}(\mathsf{T}\ \mathsf{Bool})} \to$
  $\mathsf{Neut}(|A|\ (\mathsf{pair}\ \gamma\ \langle t^N\rangle))$

Finally, we can define a constant for the eliminator of Norm(T Bool) with the signature by induction externally:

- $\mathsf{elim}_{\mathsf{Norm}(\mathsf{T}\ \mathsf{Bool})}$:
  $\Pi_{A:\Pi_{t:\mathsf{Elem}(\mathsf{T}\ \mathsf{Bool})}\mathsf{Norm}(\mathsf{T}\ \mathsf{Bool})|t\to\mathsf{U}_n}(\Pi_{t^N:\mathsf{Neut}(\mathsf{T}\ \mathsf{Bool})}A\ \langle t^N\rangle\ (\mathsf{ne}_{\mathsf{Bool}}^S\ t)) \to$
  $A\ \mathsf{false}\ \mathsf{false}^S \to A\ \mathsf{true}\ \mathsf{true}^S \to \Pi_{t:\mathsf{Elem}(\mathsf{T}\ \mathsf{Bool})}\Pi_{t^N:\mathsf{Norm}(\mathsf{T}\ \mathsf{Bool})|t} \to A\ t\ t^N$.

6.4 Category with families

We first define a type $\mathsf{Type}'_n(A)$ for A : Type_n internally. For A' : $\mathsf{Type}'_n(A)$ we have a tuple with four fields, where:

- A' : $\mathsf{Elem}(A) \to \mathsf{U}_n$
- $\mathsf{n}_{A'}$: $\mathsf{Norm}(\mathsf{U}_n)|A$
- $\mathsf{q}_{A'}$: $\Pi_{a:\mathsf{Elem}(A)}A'\ a \to \mathsf{Norm}(A)|a$

- $\mathsf{r}_{A'} : \Pi_{a : \mathsf{Neut}(A)} A'\ \langle a \rangle$

We can then define the components of the model:

- A context Γ in $\mathscr{M}^*$ is a pair consisting of a context Γ in $\mathscr{M}$ and a type $\Gamma'\ \gamma$ in the D-presheaf model for $\gamma : |\Gamma|$.
- A substitution $\gamma \in \Delta \to^* \Gamma$ is a pair consisting of $\gamma \in \Delta \to \Gamma$ and $\gamma' : \Pi_{\delta : |\Delta|} \Delta'\ \delta \to \Gamma'\ (|\gamma|\ \delta)$.
- $A \in \mathsf{Type}^*_n(\Gamma)$ is a pair consisting of $A \in \mathsf{Type}_n(\Gamma)$ and $A' : \Pi_{\gamma : |\Gamma|} \Gamma'\ \gamma \to \mathsf{Type}'_n(|A|\ \gamma)$.
- $a \in \mathsf{Elem}^*(\Gamma, A)$ is a pair consisting of $a \in \mathsf{Elem}(\Gamma, A)$ and $a' : \Pi_{\gamma : |\Gamma|} \Pi_{\gamma' : \Gamma'\ \gamma} A'\ \gamma\ \gamma'\ (|a|\ \gamma)$.
- id is defined by $\mathsf{id}'\ \gamma\ \gamma' = \gamma'$.
- Composition $\gamma\delta$ is defined by $(\gamma\delta)'\ \theta\ \theta' = \gamma'\ (|\delta|\ \theta)\ (\delta'\ \theta\ \theta')$
- The terminal context is defined by $()' = \top$.
- The substitution $()$ is defined by $()'\ \gamma\ \gamma' = \mathsf{tt}$.
- $A\gamma$ is defined by $(A\gamma)'\ \delta\ \delta' = A'\ (|\gamma|\ \delta)\ (\gamma'\ \delta\ \delta')$.
- $a\gamma$ is defined by $(a\gamma)'\ \delta\ \delta' = a'\ (|\gamma|\ \delta)\ (\gamma'\ \delta\ \delta')$.
- $\Gamma.A$ is defined by $(\Gamma.A)'\ \gamma = \Sigma_{\gamma' : \Gamma'\ (\mathsf{fst}\ \gamma)} A'\ \gamma\ \gamma'\ (\mathsf{snd}\ \gamma)$.
- p is defined by $\mathsf{p}'\ \gamma\ (\gamma', a') = \gamma'$.
- q is defined by $\mathsf{q}'\ \gamma\ (\gamma', a') = a'$.
- (γ, a) is defined by $(\gamma, a)'\ \delta\ \delta' = (\gamma'\ \delta\ \delta', a'\ \delta\ \delta')$.

6.5 Type formers

For the type formers we have:

- U_n is defined by the components:
  - $\mathsf{U}'_n\ \gamma\ \gamma'\ A = \mathsf{Type}'_n(A)$
  - $\mathsf{n}_{\mathsf{U}'_n\ \gamma\ \gamma'} = \mathsf{U}^S_n$
  - $\mathsf{q}_{\mathsf{U}'_n\ \gamma\ \gamma'}\ A\ A' = \mathsf{n}_{A'}$
  - $\mathsf{r}_{\mathsf{U}'_n\ \gamma\ \gamma'}\ A^n$ is a tuple with the components:
    - $\mathsf{r}_{\mathsf{U}'_n\ \gamma\ \gamma'}\ A^n\ a = \mathsf{Norm}\langle A^n\rangle|a$
    - $\mathsf{n}_{\mathsf{r}_{\mathsf{U}'_n\ \gamma\ \gamma'}\ A^n} = \mathsf{ne}^S_{\mathsf{U}}\ A^n$
    - $\mathsf{q}_{\mathsf{r}_{\mathsf{U}'_n\ \gamma\ \gamma'}\ A^n}\ a\ a' = a'$
    - $\mathsf{r}_{\mathsf{r}_{\mathsf{U}'_n\ \gamma\ \gamma'}\ A^n}\ a^N = \mathsf{ne}^S\ A^n\ a^N$
- $\Pi_A B$ is defined by the components:
  - $(\Pi_A B)'\ \gamma\ \gamma'\ t = \Pi_{a:\mathsf{Elem}(|A|\ \gamma)}\Pi_{a':C\ a} D\ a\ a'\ (\mathsf{app}\ \gamma\ t\ a)$
  - $\mathsf{n}_{(\Pi_A B)'\ \gamma\ \gamma'} = \Pi^S\ \gamma\ \mathsf{n}_C\ N$
  - $\mathsf{q}_{(\Pi_A B)'\ \gamma\ \gamma'}\ t\ t' = \lambda^S\ \gamma\ \mathsf{n}_C\ N\ t\ t'$
  - $\mathsf{r}_{(\Pi_A B)'\ \gamma\ \gamma'}\ t^N\ a\ a' = \mathsf{r}_{D\ a\ a'}\ (\mathsf{app}^S\ \gamma\ \mathsf{n}_C\ N\ t^N\ a\ (\mathsf{q}_C\ a\ a'))$

  where we write $C = A'\ \gamma\ \gamma'$ and $D\ a\ a' = B'\ (\mathsf{pair}\ \gamma\ a)\ (\gamma', a')$, and define $N\ a^N = \mathsf{n}_{D\ \langle a^N\rangle\ (\mathsf{r}_C\ a^N)}$ and $g\ a^N = \mathsf{q}_{D\ \langle a^N\rangle\ (\mathsf{r}_C\ a^N)}\ (\mathsf{app}\ \gamma\ t\ \langle a^N\rangle)\ (t'\ \langle a^N\rangle\ (\mathsf{r}_C\ a^N))$
- $\mathsf{app}(t, a)$ is defined by $\mathsf{app}(t, a)'\ \gamma\ \gamma' = t'\ \gamma\ \gamma'\ a\ (a'\ \gamma\ \gamma')$.
- λb is defined by $(\lambda b)'\ \gamma\ \gamma'\ a\ a' = b'\ (\mathsf{pair}\ \gamma\ a)\ (\gamma', a')$.
- Prop is defined by the components:
  - A' : $\mathsf{Prop}'\ \gamma\ \gamma'\ A$ is a tuple with four fields, where:
    - A' : $\mathsf{Elem}(\mathsf{T}\ A) \to \mathsf{Prop}$
    - $\mathsf{n}_{A'}$: $\mathsf{Norm}(\mathsf{Prop})|A$

    - $\mathsf{q}_{A'} : \Pi_{a : \mathsf{Elem}(\mathsf{T}\ A)} \mathsf{T}\ (A'\ a) \to \mathsf{Norm}(\mathsf{T}\ A)|a$
    - $\mathsf{r}_{A'} : \Pi_{a : \mathsf{Neut}(\mathsf{T}\ A)} \mathsf{T}\ (A'\ \langle a \rangle)$
  - $\mathsf{n}_{\mathsf{Prop}'\ \gamma\ \gamma'} = \mathsf{Prop}^S$
  - $\mathsf{q}_{\mathsf{Prop}'\ \gamma\ \gamma'} = \mathsf{n}_{A'}$
  - $\mathsf{r}_{\mathsf{Prop}'\ \gamma\ \gamma'}\ A^N$ is a tuple with the components:
    - $\mathsf{r}_{\mathsf{Prop}'\ \gamma\ \gamma'}\ A^N\ a = \mathsf{Norm}(\mathsf{T}\ \langle A^N \rangle)|a$
    - $\mathsf{n}_{\mathsf{r}_{\mathsf{Prop}'\ \gamma\ \gamma'}\ A^N} = \mathsf{ne}^S_{\mathsf{Prop}}\ A^N$
    - $\mathsf{q}_{\mathsf{r}_{\mathsf{Prop}'\ \gamma\ \gamma'}\ A^N}\ a\ a' = a'$
    - $\mathsf{r}_{\mathsf{r}_{\mathsf{Prop}'\ \gamma\ \gamma'}\ A^N}\ a^N = \mathsf{ne}^S(\mathsf{T}^S\ \gamma\ A^N, a^N)$
- $\mathsf{T}(A)$ is defined by the components:
  - $\mathsf{T}(A)'\ \gamma\ \gamma'\ a = \mathsf{T}(A'\ \gamma\ \gamma\ a)$
  - $\mathsf{n}_{\mathsf{T}(A)'\gamma\ \gamma'} = \mathsf{T}^S\ (\mathsf{n}_{A'\ \gamma\ \gamma'})$
  - $\mathsf{q}_{\mathsf{T}(A)'\ \gamma\ \gamma'}\ a\ a' = \mathsf{q}_{A'\ \gamma\ \gamma'}\ a\ a'$
  - $\mathsf{r}_{\mathsf{T}(A)'\ \gamma\ \gamma'}\ a^N = \mathsf{r}_{A'\ \gamma\ \gamma'}\ a^N$
- $\forall_A B$ is defined by the components:
  - $(\forall_A B)'\ \gamma\ \gamma'\ t = \forall_{a : \mathsf{Elem}(|A|\ \gamma)} \forall_{a' : C\ a} D\ a\ a'\ (\mathsf{app}\ \gamma\ (\mathsf{un}\ \gamma\ t)\ a)$
  - $\mathsf{n}_{(\forall_A B)'\ \gamma\ \gamma'} = \forall^S\ \gamma\ \mathsf{n}_C\ N$
  - $\mathsf{q}_{(\forall_A B)'\ \gamma\ \gamma'}\ t\ t' = \mathsf{mk}^S\ \gamma\ \mathsf{n}_C\ N\ t\ (\mathsf{q}_E\ (\mathsf{un}\ \gamma\ t)\ (\mathsf{un}_2\ t))$
  - $\mathsf{r}_{(\forall_A B)'\ \gamma\ \gamma'}\ t^N = \mathsf{mk}_2\ (\mathsf{r}_E\ (\mathsf{un}^S\ \gamma\ \mathsf{n}_C\ N\ t^N))$

  where we write $C = A'\ \gamma\ \gamma'$, $D\ a\ a' = B'\ (\mathsf{pair}\ \gamma\ a)\ (\gamma', a')$, and $E = (\Pi_A \mathsf{T}(B))'\ \gamma\ \gamma'$, and define $N\ a^N = \mathsf{n}_{D\ \langle a \rangle\ (\mathsf{r}_C\ a^N)}$, $\mathsf{un}_2\ t\ a\ a' = \mathsf{un}\ (\mathsf{un}\ t\ a)\ a'$, and $\mathsf{mk}_2\ t = \mathsf{mk}\ (\lambda a.\ \mathsf{mk}\ (\lambda a'.\ t\ a\ a'))$.
- Bool is defined by the components:
  - $\mathsf{Bool}'\ \gamma\ \gamma'\ t = [\mathsf{Norm}(\mathsf{T}\ \mathsf{Bool})|t]$

  - $\mathsf{n}_{\mathsf{Bool}'\ \gamma\ \gamma'} = \mathsf{Bool}^S$
  - $\mathsf{q}_{\mathsf{Bool}'\ \gamma\ \gamma'}\ t\ t' = \mathsf{un}\ t'$
  - $\mathsf{r}_{\mathsf{Bool}'\ \gamma\ \gamma'}\ t^N = \mathsf{mk}\ (\mathsf{ne}^S_{\mathsf{Bool}}\ t^N)$
- false is defined by $\mathsf{false}'\ \gamma\ \gamma' = \mathsf{mk}\ \mathsf{false}^S$
- true is defined by $\mathsf{true}'\ \gamma\ \gamma' = \mathsf{mk}\ \mathsf{true}^S$
- $\mathsf{elim}_{\mathsf{Bool}}$ is defined by $\mathsf{elim}_{\mathsf{Bool}}(A, a_0, a_1, t)'\ \gamma\ \gamma' =$ $\mathsf{elim}_{\mathsf{Norm(Bool)}}(P, k, a_0'\ \gamma\ \gamma', a_1\ \gamma\ \gamma', |t|\ \gamma, \mathsf{un}\ (t'\ \gamma\ \gamma'))$ where we define:
  - $B\ t\ t' = A'\ (\mathsf{pair}\ \gamma\ t)\ (\gamma', t')$
  - $P\ t\ t' = B\ t\ t'\ (\mathsf{elim}_{\mathsf{Bool}}\ \gamma\ (|a_0|\ \gamma)\ (|a_1|\ \gamma)\ t)$
  - $N\ t^N = \mathsf{n}_{B\ \langle t^N\rangle\ (\mathsf{ne}^S_{\mathsf{Bool}}\ t^N)}$
  - $b\ t^N = \mathsf{q}_{B\ \langle t^N\rangle\ (\mathsf{ne}^S_{\mathsf{Bool}}\ t^N)}\ (|a_0|\ \gamma)\ (a_0'\ \gamma\ \gamma')$
  - $c\ t^N = \mathsf{q}_{B\ \langle t^N\rangle\ (\mathsf{ne}^S_{\mathsf{Bool}}\ t^N)}\ (|a_1|\ \gamma)\ (a_1'\ \gamma\ \gamma')$
  - $k\ t^N = \mathsf{r}_{B\ \langle t^N\rangle\ (\mathsf{ne}^S_{\mathsf{Bool}}\ t^N)}\ (\mathsf{elim}^S_{\mathsf{Bool}}\ \gamma\ N\ (|a_0|\ \gamma)\ (b\ t^N)\ (|a_1|\ \gamma)\ (c\ t^N))\ t^N)$

6.6 Result

Finally, we instantiate $\mathscr{M}$ to be the initial model $\mathscr{M}_0$.

We define the operation $\mathsf{r}_{\mathsf{id}} \in |\Gamma'_\Gamma(\mathsf{id})|$ by induction on Γ, which is a context in $\mathscr{M}_0$, with:

$$\mathsf{r}_{\mathsf{id}_{()}} = \star \qquad\qquad \mathsf{r}_{\mathsf{id}_{\Gamma.A}} = (\Gamma'_{\mathsf{id}\ \mathsf{p}}(\mathsf{r}_{\mathsf{id}_\Gamma}), \mathsf{r}_{A'_{\Gamma.A}(\mathsf{p},\gamma')}(\mathsf{v}_0))$$

We then get the normalization operation $\mathsf{norm}(a) \in \{a^N \in \mathsf{Norm}(\Gamma, A) : \langle a^N\rangle = a\}$ for $a \in \mathsf{Elem}(\Gamma, A)$, defined by $\mathsf{norm}(a) = \mathsf{q}_{A'_\Gamma(\mathsf{id},\mathsf{r}_{\mathsf{id}})}(a, a'(\mathsf{id}, \mathsf{r}_{\mathsf{id}}))$.

From this, we get decidability of equality of elements, since $a = a'$ iff $\mathsf{norm}(a) = \mathsf{norm}(a')$.

7

Conclusion

As stated in the introduction, this proof of normalization has the modularity property. The proof can be easily extended with more type formers such as the dependent pair type, unit type, etc. as opposed to the argument by Geuvers and Werner [8]. Adding type formers just amounts to adding components to each model we have defined.

7.1 Formalization attempt

Parts of chapters 3 and 4 were formalized in the proof assistant Agda. Most of the D-set model could be defined strictly by using strict propositions, and the canonicity model could then be mostly strict by shallowly embedding the syntax [11]. Formalizing normalization is difficult, as defining a presheaf model requires a large amount of coercion operations over equalities, which prevents the model from being strict and makes it require even more coercions.

Bibliography

[1] S. Castellan, P. Clairambault, and P. Dybjer, "Categories with families: Unityped, simply typed, and dependently typed," in *Joachim Lambek: The Interplay of Mathematics, Logic, and Linguistics*, C. Casadio and P. J. Scott, Eds. Cham: Springer, 2021, pp. 135–180, ISBN: 978-3-030-66545-6. DOI: 10.1007/978-3-030-66545-6_5.

[2] T. Coquand, "An analysis of Girard's paradox," in *Proceedings of the Symposium on Logic in Computer Science*, IEEE Computer Society, 1986, pp. 227–236.

[3] T. Coquand, "Metamathematical investigations of a calculus of constructions," INRIA, Tech. Rep. RR-1088, 1989. [Online]. Available: https://inria.hal.science/inria-00075471.

[4] T. Coquand, "Reduction free normalisation for a proof irrelevant type of propositions," *Log. Methods Comput. Sci.*, vol. 19, 3 2023. DOI: 10.46298/lmcs-19(3:5)2023.

[5] T. Coquand, "The paradox of trees in type theory," *BIT*, vol. 32, no. 1, pp. 10–14, 1992. DOI: 10.1007/BF01995104.

[6] H. B. Curry and R. Feys, *Combinatory Logic*. Amsterdam: North Holland, 1958, vol. 1.

[7] P. Dybjer, "Internal type theory," in *Types for Proofs and Programs*, S. Berardi and M. Coppo, Eds., ser. Lecture Notes in Computer Science, vol. 1158, Berlin, Heidelberg: Springer, 1996, pp. 120–134, ISBN: 978-3-540-70722-6. DOI: 10.1007/3-540-61780-9_66.

[8] H. Geuvers and B. Werner, "On the Church–Rosser property for expressive type systems and its consequences for their metatheoretic study," in *Proceedings of the Ninth Annual Symposium on Logic in Computer Science*, IEEE Computer Society, 1994, pp. 320–329. DOI: 10.1109/LICS.1994.316057.

[9] M. Hofmann, "Syntax and semantics of dependent types," in *Extensional Constructs in Intensional Type Theory*. London: Springer, 1997, pp. 13–54, ISBN: 978-1-4471-0963-1. DOI: 10.1007/978-1-4471-0963-1_2.

[10] A. Kaposi, S. Huber, and C. Sattler, "Gluing for type theory," in *4th International Conference on Formal Structures for Computation and Deduction*, H. Geuvers, Ed., ser. LIPIcs, vol. 131, Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019, 25:1–25:19, ISBN: 978-3-95977-107-8. DOI: 10.4230/LIPIcs.FSCD.2019.25.

[11] A. Kaposi, A. Kovács, and N. Kraus, "Shallow embedding of type theory is morally correct," in *Mathematics of Program Construction*, G. Hutton, Ed., ser. Lecture Notes in Computer Science, vol. 11825, Cham: Springer, 2019, pp. 329–365, ISBN: 978-3-030-33636-3. DOI: https://doi.org/10.1007/978-3-030-33636-3_12.

[12] P. Martin-Löf, "Constructive mathematics and computer programming," in *Logic, Methodology and Philosophy of Science VI*, ser. Studies in Logic and the Foundations of Mathematics, L. J. Cohen, J. Łoś, H. Pfeiffer, and K.-P. Podewski, Eds., vol. 104, Elsevier, 1982, pp. 153–175. DOI: https://doi.org/10.1016/S0049-237X(09)70189-2.

[13] T. Streicher, *Semantics of Type theory – Correctness, Completeness and Independence Results* (Progress in Theoretical Computer Science). Birkhäuser, 1991, ISBN: 978-0-8176-3594-7.