An extended calculus of constructions by Zhaohui Luo (sections 7.1 & 7.2)

Teun van Brakel

Institute for Computing and Information Sciences – Software Science
Radboud University Nijmegen

3 December 2025

Table of contents

- Goal
- ω-sets
- Morphism of ω-sets
- Type universe
- Interpretation of context, types and terms
- First projection property
- Conclusion

Goal

The goal is to use sets to model the calculus of extended calculus of constructions

(recap / intuition)

- $(|A|, \Vdash_A)$
- Elements of $|A|$ are the *objects* of the set.
- The relation $n \Vdash_A x$ means:

  "the code n gives a (computational) representation of x."

ω-sets: definition

ω-set

An *ω-set* is a pair

$$A = (|A|, \Vdash_A)$$

where $|A|$ is a set (the carrier) and $\Vdash_A \subseteq \omega \times |A|$ is a *realizability relation*.

ω-sets: definition

ω-set

An *ω-set* is a pair

$$A = (|A|, \Vdash_A)$$

where $|A|$ is a set (the carrier) and $\Vdash_A \subseteq \omega \times |A|$ is a *realizability relation*.

Surjectivity condition

We require:

$$\forall a \in |A|.\ \exists n \in \omega.\ n \Vdash_A a.$$

(Every element has at least one realizer.)

ω-sets: Boolean example

Example: Booleans

Let $|A| = \{\text{true}, \text{false}\}$ and suppose we choose two codes in ω as realizers:

$$\omega \supseteq \{r_{\text{true}, r_{\text{false}}\}}.$$

Set

$$r_{\text{true}} \Vdash_A \text{true}, \qquad r_{\text{false}} \Vdash_A \text{false}.$$

Then $\Vdash_A$ consists of these pairs (plus any additional pairs allowed).

Morphisms (Recap)

example

suppose we have two sets
$(\mathbb{N}, \Vdash_{\mathbb{N}})$
and $(\mathbb{B}, \Vdash_{\mathbb{B}})$
If we want to go $\mathbb{N} \to \mathbb{B}$ we could use the function:
$f : even : \mathbb{N} \to \mathbb{B}$
e: is the implementation of the function such that
if $d \in \mathbb{N}$ & $d \Vdash_{\mathbb{N}} n$ then $ed \Vdash_{\mathbb{B}} even(n)$

Morphisms of ω-sets (definition)

A morphism $f : A \to B$ consists of:

1. A function on carriers $f : |A| \to |B|$;
2. *Computable realizer*: there exists $e \in \omega$ (a code) such that

$$\forall a \in |A| \, \forall m \in \omega. \quad m \Vdash_A a \Longrightarrow e \cdot m \Vdash_B f(a).$$

(Here $e \cdot m$ denotes Kleene application / partial recursive application.)

Type Universe Structure

- $\mathrm{Prop} \in \mathrm{Type}_0 \in \mathrm{Type}_1 \in \mathrm{Type}_2 \ldots$

Type Universe Structure

- $\text{Prop} \in \text{Type}_0 \in \text{Type}_1 \in \text{Type}_2 \dots$
- $\text{Prop} \subseteq \text{Type}_0 \subseteq \text{Type}_1 \subseteq \dots$

Type Universe Structure

- Prop $\in$ Type$_0$ $\in$ Type$_1$ $\in$ Type$_2 \ldots$
- Prop $\subseteq$ Type$_0$ $\subseteq$ Type$_1$ $\subseteq \ldots$
- Each Type$_j$ closed under Π and Σ

Type Universe Structure

- Prop $\in$ Type$_0$ $\in$ Type$_1$ $\in$ Type$_2 \ldots$
- Prop $\subseteq$ Type$_0$ $\subseteq$ Type$_1$ $\subseteq \ldots$
- Each Type$_j$ closed under Π and Σ
- Prop closed under Π

Empty context

The empty context is interpreted as the trivial ω-set:

$$[\![\langle\rangle]\!](1,\ \omega \times 1)$$

- Carrier $|1| = \{*\}$.
- Every $n \in \omega$ realizes $*$: $n \Vdash_{[\![\langle\rangle]\!]} *$.

Types as Γ-indexed families

A type judgment $\Gamma \vdash A : \mathrm{Type}_j$ is interpreted as:
$[\![A]\!] : |[\![\Gamma]\!]| \longrightarrow \omega\text{-set}.$

- For each environment $\gamma \in |[\![\Gamma]\!]|$, $[\![A]\!](\gamma)$ is an ω-set.
- So a type is a family of carriers with realizability relations depending on γ.

Context extention

Sementically

$[\![\Gamma, x : A]\!] = \sigma([\![\Gamma]\!], [\![A]\!])$

- $[\![A]\!]$ is a family $|[\![\Gamma]\!]| \to \omega - set$.
- σ bundles an environment with a value for the new variable.

The σ-construction (context extension)

Let Γ be an ω-set and $A : |\Gamma| \to$ be a Γ-indexed family.
Define:

$$|\sigma(\Gamma, A)|\{(\gamma, a) \mid \gamma \in |\Gamma|,\ a \in |A(\gamma)|\}.$$

Realizers:

$$\langle m, n\rangle \Vdash_{\sigma(\Gamma,A)} (\gamma, a) \quad \Longleftrightarrow \quad m \Vdash_{\Gamma} \gamma \ \wedge \ n \Vdash_{A(\gamma)} a.$$

So realizers for a pair are pairs of realizers.

Extention example

Example

Let Γ = (x:Bool)
Let A(x) = Nat
Then:
$|[\![\Gamma, x : A]\!]| = \{(true, n), (false, n) | n \in Nat\}$

Dependent extension example

Example (dependent)

Let $\Gamma = (n : Nat)$
Let $A(n) = \mathrm{Vec}(\mathrm{Bool}, n)$
Then:

$$|[\![\Gamma, x : A]\!]| = \{(n, v) \mid n \in Nat,\ v \in Bool^n\}.$$

Realizers:

- a realizer for n
- a realizer computing a Boolean vector of length n

Terms as morphisms

A term judgment $\Gamma \vdash M : A$ is interpreted as a morphism:

$$[\![M]\!] : |[\![\Gamma]\!]| \longrightarrow |[\![A]\!]|$$

together with a realizer-index e such that for each γ and each $m \Vdash_\Gamma \gamma$ we have:

$$e \cdot m \Vdash_{[\![A]\!](\gamma)} [\![M]\!](\gamma).$$

First Projection Property (FPP)

Definition

A map $f : \Gamma \to \sigma(\Gamma, A)$ satisfies FPP if:

$$\pi_1 \circ f = \mathrm{id}_\Gamma$$

and the realizer for $f(\gamma)$ depends only on a realizer for γ.

First Projection Property (FPP)

Definition

A map $f : \Gamma \to \sigma(\Gamma, A)$ satisfies FPP if:

$$\pi_1 \circ f = \mathrm{id}_\Gamma$$

and the realizer for $f(\gamma)$ depends only on a realizer for γ.

- FPP guarantees that a term producing an element of the extended context does not change the underlying environment.

First Projection Property (FPP)

Definition

A map $f : \Gamma \to \sigma(\Gamma, A)$ satisfies FPP if:

$$\pi_1 \circ f = \mathrm{id}_\Gamma$$

and the realizer for $f(\gamma)$ depends only on a realizer for γ.

- FPP guarantees that a term producing an element of the extended context does not change the underlying environment.
- It is crucial when interpreting dependent terms as sections of σ.

General interpretation

$\Gamma \in \omega\text{-Set}$
$K : |\Gamma| \to \omega\text{-Set}$
$K(\gamma) = (X, \omega \times X) \;\; \forall \gamma \in |\Gamma|$

$n \Vdash_{K(\gamma)} X$

This means that terms like (Γ, K) are similar to morphisms of $|\Gamma| \to X$

Summary

- ω-sets give a computational semantics for ECC.
- Contexts interpreted as ω-sets via the σ-construction.
- Types become families of ω-sets over contexts.
- Terms become computable morphisms.
- The First Projection Property ensures coherence of context extension.

Questions

Are there any questions?