The Category of Iterative Sets in Cubical Agda

Stockholm–Gothenburg Type Theory Meeting

Fabian Lukas Grubmüller

Wednesday, 22 April 2026

Contents

1. Motivation

Motivation

Motivation

Previous article by Gratzer, Gylterud, Mörtberg, Stenholm [1]

Motivation

Previous article by Gratzer, Gylterud, Mörtberg, Stenholm [1]

- developing iterative sets

Motivation

Previous article by Gratzer, Gylterud, Mörtberg, Stenholm [1]

- developing iterative sets
- formalization in Agda Unimath

Motivation

Previous article by Gratzer, Gylterud, Mörtberg, Stenholm [1]

- developing iterative sets
- formalization in Agda Unimath
- problems with defining additional structures on CwF

Motivation

Previous article by Gratzer, Gylterud, Mörtberg, Stenholm [1]

- developing iterative sets
- formalization in Agda Unimath
- problems with defining additional structures on CwF

> [...] even though the naturality requires complex path algebra to state properly, in the specific CwF on V all these paths are given by reflexivity.

Motivation

Previous article by Gratzer, Gylterud, Mörtberg, Stenholm [1]

- developing iterative sets
- formalization in Agda Unimath
- problems with defining additional structures on CwF

> [...] even though the naturality requires complex path algebra to state properly, in the specific CwF on V all these paths are given by reflexivity.

Idea: Easier to formalize in cubical setting

2. Iterative Sets

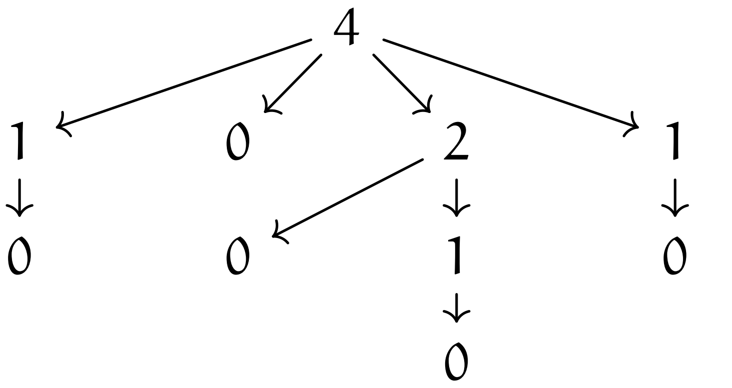
Iterative Multisets

Iterative Multisets

Idea: Viewing **sets** as **trees**:

Idea: Viewing **sets** as **trees**:

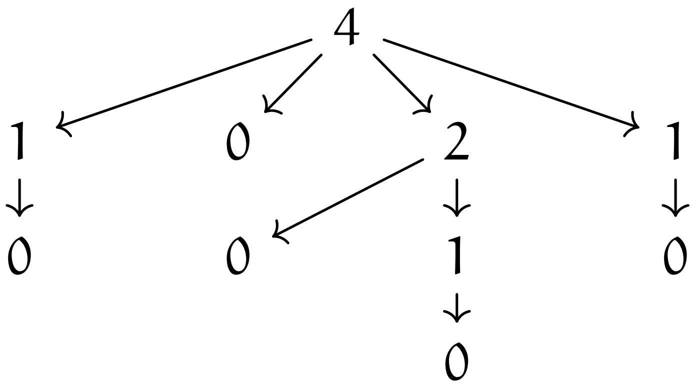
$$\{\{\emptyset\}, \emptyset, \{\emptyset, \{\emptyset\}\}, \{\emptyset\}\}$$



Iterative Multisets

Idea: Viewing **sets** as **trees**:

$$\{\{\emptyset\}, \emptyset, \{\emptyset, \{\emptyset\}\}, \{\emptyset\}\}$$

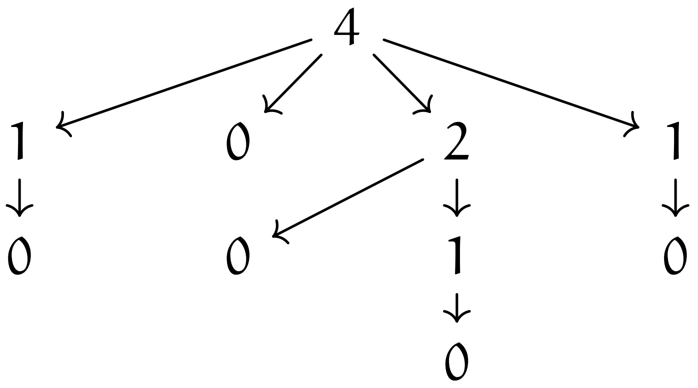


Data: some **type** A and some **function** $f : A \to$ ...

Iterative Multisets

Idea: Viewing **sets** as **trees**:

$\{\{\emptyset\}, \emptyset, \{\emptyset, \{\emptyset\}\}, \{\emptyset\}\}$



Data: some **type** A and some **function** $f : A \to \ldots$

Definition **W-types:** $W_{x:A} B(x)$ with $A : \mathrm{Type}$ and $B : A \to \mathrm{Type}$ has elements

$$\mathrm{sup}(a, f) \quad \textbf{with} \quad a : A, \quad f : B(a) \to W_{x:A} B(x)$$

Iterative Multisets

Definition **W-types:** $W_{x:A}B(x)$ with $A : \mathsf{Type}$ and $B : A \to \mathsf{Type}$ has elements

$$\mathsf{sup}(a, f) \quad \textbf{with} \quad a : A, \quad f : B(a) \to W_{x:A}B(x)$$

Iterative Multisets

Definition **W-types:** $W_{x:A} B(x)$ with $A : \mathsf{Type}$ and $B : A \to \mathsf{Type}$ has elements

$$\mathsf{sup}(a, f) \quad \textbf{with} \quad a : A, \quad f : B(a) \to W_{x:A} B(x)$$

Definition $V^\infty := W_{A:\,\mathsf{Type}} A$ has elements

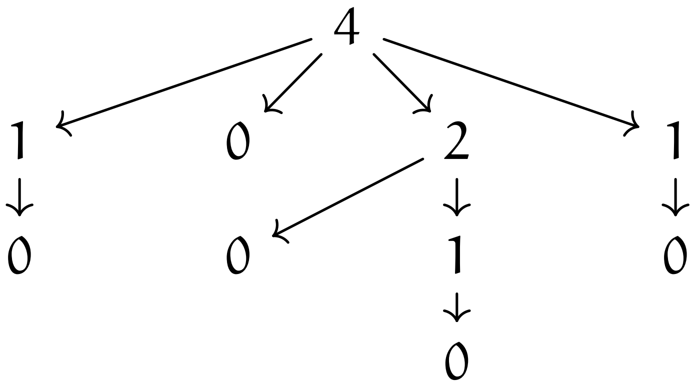
$$\mathsf{sup}(A, f) \quad \textbf{with} \quad A : \mathsf{Type}, \quad f : A \to V^\infty$$

Iterative Multisets

Definition $V^\infty := W_{A:\,\mathrm{Type}} A$ has elements

$$\mathrm{sup}(A, f) \quad \textbf{with} \quad A : \mathrm{Type}, \quad f : A \to V^\infty$$

$$\{\{\emptyset\}, \emptyset, \{\emptyset, \{\emptyset\}\}, \{\emptyset\}\}$$

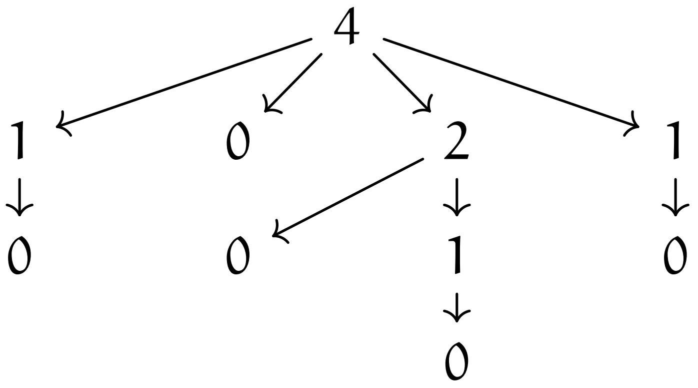


Iterative Multisets

Definition $V^\infty :\equiv W_{A:\,\mathsf{Type}} A$ has elements

$$\mathsf{sup}(A, f) \quad \textbf{with} \quad A : \mathsf{Type}, \quad f : A \to V^\infty$$

$\{\{\emptyset\}, \emptyset, \{\emptyset, \{\emptyset\}\}, \{\emptyset\}\}$



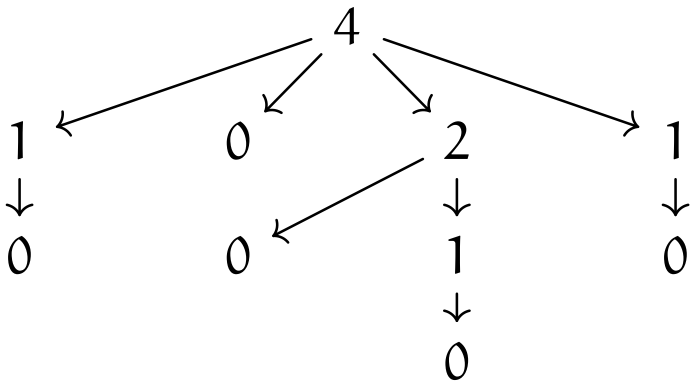
- **multisets** with **multiplicities** $(x \in^\infty (\mathsf{sup}(B, g)) :\equiv \mathsf{fiber}_g(x) :\equiv \sum_{b:B} g(b) =_{V^\infty} x$

Iterative Multisets

Definition $V^\infty := W_{A:\,\mathrm{Type}} A$ has elements

$$\sup(A, f) \quad \textbf{with} \quad A : \mathrm{Type}, \quad f : A \to V^\infty$$

$\{\{\emptyset\}, \emptyset, \{\emptyset, \{\emptyset\}\}, \{\emptyset\}\}$



- **multisets** with **multiplicities** $(x \in^\infty (\sup(B, g)) := \mathrm{fiber}_g(x) := \sum_{b:B} g(b) \equiv_{V^\infty} x$
- **decoding:** $\mathrm{El}(\sup(A, f)) := A$

Iterative Sets

Iterative Sets

Idea: for *sets*, multiplicities should be propositional ($\rightsquigarrow$ embedding)

Idea: for *sets*, multiplicities should be propositional ($\rightsquigarrow$ embedding)

Definition (Hereditarily) iterative sets:

$$\mathrm{isIterativeSet}(\mathrm{sup}(A, f)) :\equiv \mathrm{isEmbedding}(f) \times \prod_{x:A} \mathrm{isIterativeSet}(f(x))$$

Idea: for *sets*, multiplicities should be propositional ($\rightsquigarrow$ embedding)

Definition (Hereditarily) iterative sets:

$$\mathsf{isIterativeSet}(\mathsf{sup}(A, f)) :\equiv \mathsf{isEmbedding}(f) \times \prod_{x:A} \mathsf{isIterativeSet}(f(x))$$

Definition Type of iterative sets: $V^0 :\equiv \sum_{x:V^\infty} \mathsf{isIterativeSet}(x)$

Idea: for *sets*, multiplicities should be propositional ($\rightsquigarrow$ embedding)

Definition (Hereditarily) iterative sets:

$$\mathsf{isIterativeSet}(\mathsf{sup}(A, f)) :\equiv \mathsf{isEmbedding}(f) \times \prod_{x:A} \mathsf{isIterativeSet}(f(x))$$

Definition Type of iterative sets: $V^0 :\equiv \sum_{x:V^\infty} \mathsf{isIterativeSet}(x)$

- NB: hierarchy of truncated multisets V^n [2]

Idea: for *sets*, multiplicities should be propositional ($\rightsquigarrow$ embedding)

Definition (Hereditarily) iterative sets:

$$\mathsf{isIterativeSet}(\mathsf{sup}(A, f)) :\equiv \mathsf{isEmbedding}(f) \times \prod_{x:A} \mathsf{isIterativeSet}(f(x))$$

Definition Type of iterative sets: $V^0 :\equiv \sum_{x:V^\infty} \mathsf{isIterativeSet}(x)$

- NB: hierarchy of truncated multisets V^n [2]
- **decodings** similar: $\mathsf{El}(\mathsf{sup}(A, f), P) :\equiv A$,

Iterative Sets

Idea: for *sets*, multiplicities should be propositional ($\rightsquigarrow$ embedding)

Definition (Hereditarily) iterative sets:

$$\mathrm{isIterativeSet}(\mathrm{sup}(A, f)) :\equiv \mathrm{isEmbedding}(f) \times \prod_{x:A} \mathrm{isIterativeSet}(f(x))$$

Definition Type of iterative sets: $V^0 :\equiv \sum_{x:V^\infty} \mathrm{isIterativeSet}(x)$

- NB: hierarchy of truncated multisets V^n [2]
- **decodings** similar: $\mathrm{El}(\mathrm{sup}(A, f), P) :\equiv A$, **multiplicities** $x \in^0 y$ **propositional**

Properties

Properties

- **Fibration Identity Principle** (~ Extensionality)

$$(x \equiv_{V^0} y) \simeq \prod_{z:V^0} \left(z \in^0 x\right) \simeq \left(z \in^0 y\right)$$

Properties

- **Fibration Identity Principle** (~ Extensionality)

$$(x \equiv_{V^0} y) \simeq \prod_{z:V^0} \left(z \in^0 x\right) \simeq \left(z \in^0 y\right)$$

- V^0 is a **set** $\rightsquigarrow$ all **decodings** are **sets**

Properties

- **Fibration Identity Principle** (~ Extensionality)

$$(x \equiv_{V^0} y) \simeq \prod_{z:V^0} \big(z \in^0 x\big) \simeq \big(z \in^0 y\big)$$

- V^0 is a **set** $\rightsquigarrow$ all **decodings** are **sets** (**not univalent**)
- Universe **closed** under many standard **type formers**

Properties

- **Fibration Identity Principle** (~ Extensionality)

$$(x \equiv_{V^0} y) \simeq \prod_{z:V^0} \big(z \in^0 x\big) \simeq \big(z \in^0 y\big)$$

- V^0 is a **set** $\rightsquigarrow$ all **decodings** are **sets** (**not univalent**)
- Universe **closed** under many standard **type formers**
  - empty set, singleton sets, (un-) ordered pairs, natural numbers

Properties

- **Fibration Identity Principle** (~ Extensionality)

$$(x \equiv_{V^0} y) \simeq \prod_{z:V^0} (z \in^0 x) \simeq (z \in^0 y)$$

- V^0 is a **set** $\rightsquigarrow$ all **decodings** are **sets** (**not univalent**)
- Universe **closed** under many standard **type formers**
  - empty set, singleton sets, (un-) ordered pairs, natural numbers
  - identity types, Π-types, Σ-types, quotients, universe codes

Properties

- **Fibration Identity Principle** (~ Extensionality)

$$(x \equiv_{V^0} y) \simeq \prod_{z:V^0} \big(z \in^0 x\big) \simeq \big(z \in^0 y\big)$$

- V^0 is a **set** $\rightsquigarrow$ all **decodings** are **sets** (**not univalent**)
- Universe **closed** under many standard **type formers**
  - empty set, singleton sets, (un-) ordered pairs, natural numbers
  - identity types, Π-types, Σ-types, quotients, universe codes

3. Iterative Sets as a model of Dependent Type Theory

Categories with Families

Definition A category with families (**CwF**) consists of:

Definition A category with families (**CwF**) consists of:

- a **category** $\mathcal{C}$ with a **terminal object**

Categories with Families

Definition A category with families (**CwF**) consists of:

- a **category** $\mathcal{C}$ with a **terminal object**
- a **presheaf** $\mathcal{T}\mathsf{y} : \mathcal{C}^{\mathrm{op}} \longrightarrow \mathbf{hSets}$

Definition A category with families (**CwF**) consists of:

- a **category** $\mathcal{C}$ with a **terminal object**
- a **presheaf** $\mathrm{Ty} : \mathcal{C}^{\mathrm{op}} \to \mathbf{hSets}$
- a **presheaf** $\mathrm{Tm} : \left(\int \mathrm{Ty}\right)^{\mathrm{op}} \to \mathbf{hSets}$

Definition A category with families (**CwF**) consists of:

- a **category** $\mathcal{C}$ with a **terminal object**
- a **presheaf** $\mathrm{Ty} : \mathcal{C}^{\mathrm{op}} \to \mathbf{hSets}$
- a **presheaf** $\mathrm{Tm} : \left(\int \mathrm{Ty}\right)^{\mathrm{op}} \to \mathbf{hSets}$
- a **functor** $- . - : \int \mathrm{Ty} \to \mathcal{C}$

Definition A category with families (**CwF**) consists of:

- a **category** $\mathcal{C}$ with a **terminal object**
- a **presheaf** $\mathrm{Ty} : \mathcal{C}^{\mathrm{op}} \to \mathbf{hSets}$
- a **presheaf** $\mathrm{Tm} : \left(\int \mathrm{Ty}\right)^{\mathrm{op}} \to \mathbf{hSets}$
- a **functor** $- \,.\, - : \int \mathrm{Ty} \to \mathcal{C}$
- for each $(\Gamma, A) : \int \mathrm{Ty}$ a **natural equivalence** in $\Delta : \mathcal{C}$:

$$\mathcal{C}(\Delta, \Gamma \,.\, A) \simeq \sum_{\gamma : \mathcal{C}(\Delta, \Gamma)} \mathrm{Tm}(\Delta, \gamma^* A)$$

Σ-structures

Definition A **Σ-Structure** on a CwF $\mathcal{C}$ consists of

Σ-structures

Definition A **Σ-Structure** on a CwF $\mathcal{C}$ consists of

- An operation $\mathcal{S} : (\Gamma : \mathcal{C})(A : \mathrm{Ty}\,\Gamma) \to \mathrm{Ty}(\Gamma \,.\, A) \to \mathrm{Ty}\,\Gamma$

Definition A **Σ-Structure** on a CwF $\mathcal{C}$ consists of

- An operation $\mathcal{S} : (\Gamma : \mathcal{C})(A : \mathrm{Ty}\,\Gamma) \to \mathrm{Ty}(\Gamma\,.\,A) \to \mathrm{Ty}\,\Gamma$
- For $\Gamma : \mathcal{C}$, $A : \mathrm{Ty}\,\Gamma$, $B : \mathrm{Ty}(\Gamma\,.\,A)$ a natural (in Γ) iso

Σ-structures

Definition A **Σ-Structure** on a CwF $\mathcal{C}$ consists of

- An operation $S : (\Gamma : \mathcal{C})(A : \mathrm{Ty}\,\Gamma) \to \mathrm{Ty}(\Gamma \,.\, A) \to \mathrm{Ty}\,\Gamma$
- For $\Gamma : \mathcal{C}$, $A : \mathrm{Ty}\,\Gamma$, $B : \mathrm{Ty}(\Gamma \,.\, A)$ a natural (in Γ) iso

$$\mathrm{Tm}(\Gamma, S_\Gamma(A, B)) \simeq \sum_{a : \mathrm{Tm}(\Gamma, A)} \mathrm{Tm}\big(\Gamma, \langle 1_\Gamma, a\rangle^* B\big)$$

$\mathcal{V}^0$ as a CwF $\mathcal{V}$

V^0 as a CwF $\mathcal{V}$

- **objects:** V^0

$\mathcal{V}^0$ as a CwF $\mathcal{V}$

- **objects:** $\mathcal{V}^0$
- **arrows:** $\mathcal{V}(\Gamma, \Delta) :\equiv \mathrm{El}(\Gamma) \rightarrow \mathrm{El}(\Delta)$

V^0 as a CwF $\mathcal{V}$

- **objects:** V^0
- **arrows:** $\mathcal{V}(\Gamma, \Delta) :\equiv \mathsf{El}(\Gamma) \to \mathsf{El}(\Delta)$
- **types:** $\mathsf{Ty}(\Gamma) :\equiv \mathsf{El}(\Gamma) \to V^0$

V^0 as a CwF $\mathcal{V}$

- **objects:** V^0
- **arrows:** $\mathcal{V}(\Gamma, \Delta) :\equiv \mathrm{El}(\Gamma) \to \mathrm{El}(\Delta)$
- **types:** $\mathrm{Ty}(\Gamma) :\equiv \mathrm{El}(\Gamma) \to V^0$
- **terms:** $\mathrm{Tm}(\Gamma, A) :\equiv \prod_{y : \mathrm{El}(\Gamma)} \mathrm{El}(A(y))$

V^0 as a CwF $\mathcal{V}$

- **objects:** V^0
- **arrows:** $\mathcal{V}(\Gamma, \Delta) :\equiv \mathsf{El}(\Gamma) \to \mathsf{El}(\Delta)$
- **types:** $\mathsf{Ty}(\Gamma) :\equiv \mathsf{El}(\Gamma) \to V^0$
- **terms:** $\mathsf{Tm}(\Gamma, A) :\equiv \prod_{y : \mathsf{El}(\Gamma)} \mathsf{El}(A(y))$
- **context extension:** $\Gamma \,.\, A :\equiv \Sigma^0(\Gamma, A)$

V^0 as a CwF $\mathcal{V}$

- **objects:** V^0
- **arrows:** $\mathcal{V}(\Gamma, \Delta) \coloneqq \mathrm{El}(\Gamma) \to \mathrm{El}(\Delta)$
- **types:** $\mathrm{Ty}(\Gamma) \coloneqq \mathrm{El}(\Gamma) \to V^0$
- **terms:** $\mathrm{Tm}(\Gamma, A) \coloneqq \prod_{y : \mathrm{El}(\Gamma)} \mathrm{El}(A(y))$
- **context extension:** $\Gamma \,.\, A \coloneqq \Sigma^0(\Gamma, A)$
- **Σ-structure:** $S_\Gamma(A, B) \coloneqq \lambda x : \mathrm{El}(\Gamma). \quad \Sigma^0(A(x), \lambda a.B(x, a))$

4. Formalization

Formalization Steps

Formalization Steps

Part 1

- Iterative **multisets**
- Iterative **sets**
- ... as **universe**
- ... as **category**

Formalization Steps

Part 1

- Iterative **multisets**
- Iterative **sets**
- ... as **universe**
- ... as **category**

Part 2

- **CwF** and **Σ-structure** (general)

Formalization Steps

Part 1

- Iterative **multisets**
- Iterative **sets**
- ... as **universe**
- ... as **category**

Part 2

- **CwF** and **Σ-structure** (general)
- **Instantiation** for iterative sets

Formalization Steps

Part 1

- Iterative **multisets**
- Iterative **sets**
- ... as **universe**
- ... as **category**

Part 2

- **CwF** and **Σ-structure** (general)
- **Instantiation** for iterative sets (3 phases)
  1. Naive
  2. "Cubical"
  3. Ad-hoc functions

Formalization Steps

Part 1

- Iterative **multisets**
- Iterative **sets**
- ... as **universe**
- ... as **category**

Part 2

- **CwF** and **Σ-structure** (general)
- **Instantiation** for iterative sets (3 phases)
  1. Naive
  2. “Cubical”
  3. Ad-hoc functions
- **General statement** for all Tarski set-universes of sets

Towards Pairs in V^0

Towards Pairs in V^0

- **empty set**: $\emptyset^0 := (0, \lambda())$

Towards Pairs in V^0

- **empty set**: $\emptyset^0 := (0, \lambda())$
- **singleton**: $\{x\}^0 := (1, \mathrm{const}(x))$

Towards Pairs in V^0

- **empty set**: $\emptyset^0 := (0, \lambda())$
- **singleton**: $\{x\}^0 := (1, \mathrm{const}(x))$
- **unordered pairs**: $\{x, y\}^0 := (\mathrm{Bool}, f)$ where $f(\mathrm{false}) = x$, $f(\mathrm{true}) = y$

Towards Pairs in V^0

- **empty set**: $\emptyset^0 := (0, \lambda())$
- **singleton**: $\{x\}^0 := (1, \mathrm{const}(x))$
- **unordered pairs**: $\{x, y\}^0 := (\mathrm{Bool}, f)$ where $f(\mathrm{false}) = x$, $f(\mathrm{true}) = y$
- **ordered pairs**:

$$\langle x, y\rangle^0 := \left\{ \left\{ \{x\}^0, \emptyset^0 \right\}^0, \left\{ \{y\}^0 \right\}^0 \right\}^0$$

Towards Pairs in V^0

- **empty set**: $\emptyset^0 := (0, \lambda())$
- **singleton**: $\{x\}^0 := (1, \mathrm{const}(x))$
- **unordered pairs**: $\{x, y\}^0 := (\mathrm{Bool}, f)$ where $f(\mathrm{false}) = x$, $f(\mathrm{true}) = y$
- **ordered pairs**:

$$\langle x, y\rangle^0 := \left\{\left\{\{x\}^0, \emptyset^0\right\}^0, \left\{\{y\}^0\right\}^0\right\}^0$$

- $\rightsquigarrow$ **Π / Σ types**

$\mathcal{V}^0$ as a CwF $\mathcal{V}$

V^0 as a CwF $\mathcal{V}$

- **objects:** V^0
- **arrows:** $\mathcal{V}(\Gamma, \Delta) \coloneqq \mathrm{El}(\Gamma) \to \mathrm{El}(\Delta)$
- **types:** $\mathrm{Ty}(\Gamma) \coloneqq \mathrm{El}(\Gamma) \to V^0$
- **terms:** $\mathrm{Tm}(\Gamma, A) \coloneqq \prod_{y:\, \mathrm{El}(\Gamma)} \mathrm{El}(A(y))$
- **context extension:** $\Gamma \,.\, A \coloneqq \Sigma^0(\Gamma, A)$
- **Σ-structure:** $S_\Gamma(A, B) \coloneqq \lambda x : \mathrm{El}(\Gamma). \quad \Sigma^0(A(x), \lambda a.B(x, a))$

$(\mathsf{U}, \mathsf{El})$ as a CwF $\mathcal{U}$

$(\mathsf{U}, \mathsf{El})$ as a CwF $\mathcal{U}$

- **objects:** U
- **arrows:** $\mathcal{U}(\Gamma, \Delta) :\equiv \mathsf{El}(\Gamma) \to \mathsf{El}(\Delta)$
- **types:** $\mathsf{Ty}(\Gamma) :\equiv \mathsf{El}(\Gamma) \to \mathcal{U}$
- **terms:** $\mathsf{Tm}(\Gamma, A) :\equiv \prod_{y : \mathsf{El}(\Gamma)} \mathsf{El}(A(y))$
- **context extension:** $\Gamma \,.\, A :\equiv \Sigma(\Gamma, A)$
- **Σ-structure:** $S_\Gamma(A, B) :\equiv \lambda x : \mathsf{El}(\Gamma). \quad \Sigma(A(x), \lambda a.B(x, a))$

Naturality of Context Extension

$$
\begin{array}{ccccc}
\Delta & & \mathcal{C}(\Delta, \Gamma \,.\, A) & \xrightarrow{\;\cong\;} & \sum_{\tau : \mathcal{C}(\Delta,\Gamma)} \mathrm{Tm}(\Delta, (\tau^* A)) \\
 & & & & \Big\downarrow {\scriptstyle (\mathrm{id}, -[\sigma])} \\
 & & & & \sum_{\tau : \mathcal{C}(\Delta,\Gamma)} \mathrm{Tm}(\Delta', (\sigma^*(\tau^* A))) \\
\Big\uparrow {\scriptstyle \sigma} & & \Big\downarrow {\scriptstyle \sigma^*} & & \Big\| {\scriptstyle (\mathrm{id}, \mathrm{transport}_p(-))} \\
 & & & & \sum_{\tau : \mathcal{C}(\Delta,\Gamma)} \mathrm{Tm}\big(\Delta', \big((\tau \circ \sigma)^* A\big)\big) \\
 & & & & \Big\downarrow {\scriptstyle (\sigma^*, \mathrm{id})} \\
\Delta' & & \mathcal{C}(\Delta', \Gamma \,.\, A) & \xrightarrow{\;\cong\;} & \sum_{\tau : \mathcal{C}(\Delta',\Gamma)} \mathrm{Tm}(\Delta', (\tau^* A))
\end{array}
$$

Naturality of Σ-structure

$$
\begin{array}{ccccc}
\Gamma & & \mathrm{Tm}(\Gamma, S_\Gamma(A, B)) & \xrightarrow{\ \cong\ } & \sum_{a' : \mathrm{Tm}(\Gamma, A)} \mathrm{Tm}\big(\Gamma, \langle 1_\Gamma, a' \rangle^* B\big) \\
\uparrow {\scriptstyle f} & & \downarrow {\scriptstyle -[f]} & & \downarrow {\scriptstyle (\mathrm{id},\, -[f])} \\
 & & & & \sum_{a' : \mathrm{Tm}(\Gamma, A)} \mathrm{Tm}\big(\Delta, f^* \langle 1_\Gamma, a' \rangle^* B\big) \\
 & & & & \downarrow {\scriptstyle \left(\mathrm{id},\, \mathrm{subst}_q^{\mathrm{Tm}(\Delta, -)}(-)\right)} \\
\Delta & & \mathrm{Tm}(\Delta, f^* S_\Gamma(A, B)) & & \sum_{a' : \mathrm{Tm}(\Gamma, A)} \mathrm{Tm}\big(\Delta, \langle 1_\Delta, a'[f] \rangle^* \langle f, A \rangle^* B\big) \\
\| & & \downarrow {\scriptstyle \mathrm{subst}_p^{\mathrm{Tm}(\Delta, -)}(-)} & & \downarrow {\scriptstyle (-[f],\, \mathrm{id})} \\
\Delta & & \mathrm{Tm}\big(\Delta, S_\Delta\big(f^* A, \langle f, A \rangle^* B\big)\big) & \xrightarrow{\ \cong\ } & \sum_{a : \mathrm{Tm}(\Delta, f^* A)} \mathrm{Tm}\big(\Delta, \langle 1_\Delta, a \rangle^* \langle f, A \rangle^* B\big)
\end{array}
$$

Challenges

Challenges

Mathematical

Challenges

Mathematical

- using **"more cubical"** techniques

Challenges

Mathematical

- using **"more cubical"** techniques
- **non-definitional** behaviour of **transport** for refl

Challenges

Mathematical

- using **“more cubical”** techniques
- **non-definitional** behaviour of **transport** for refl
- lack of implicit J-rule

Challenges

Mathematical

- using **"more cubical"** techniques
- **non-definitional** behaviour of **transport** for refl
- lack of implicit J-rule

Meta-Level

Challenges

Mathematical

- using **"more cubical"** techniques
- **non-definitional** behaviour of **transport** for refl
- lack of implicit J-rule

Meta-Level

- documentation/discoverability

Challenges

Mathematical

- using **"more cubical"** techniques
- **non-definitional** behaviour of **transport** for refl
- lack of implicit J-rule

Meta-Level

- documentation/discoverability
- confusing goals/normalization

Challenges

Mathematical

- using **"more cubical"** techniques
- **non-definitional** behaviour of **transport** for refl
- lack of implicit J-rule

Meta-Level

- documentation/discoverability
- confusing goals/normalization
- confusing error messages

Challenges

Mathematical

- using **“more cubical”** techniques
- **non-definitional** behaviour of **transport** for $refl$
- lack of implicit J-rule

Meta-Level

- documentation/discoverability
- confusing goals/normalization
- confusing error messages
- termination?

Outlook

Challenges

Mathematical

- using **“more cubical”** techniques
- **non-definitional** behaviour of **transport** for $refl$
- lack of implicit J-rule

Meta-Level

- documentation/discoverability
- confusing goals/normalization
- confusing error messages
- termination?

Outlook

- complete formalization

Challenges

Mathematical

- using **“more cubical”** techniques
- **non-definitional** behaviour of **transport** for $refl$
- lack of implicit J-rule

Meta-Level

- documentation/discoverability
- confusing goals/normalization
- confusing error messages
- termination?

Outlook

- complete formalization

Bibliography

[1] Daniel Gratzer, Håkon Robbestad Gylterud, Anders Mörtberg, and Elisabeth Stenholm, "The category of iterative sets in homotopy type theory and univalent foundations," *Mathematical Structures in Computer Science*, 2024, doi: 10.1017/S0960129524000288.

[2] Håkon Robbestad Gylterud and Elisabeth Stenholm, "Univalent material set theory," 2024. doi: 10.48550/arXiv.2312.13024.

[3] F. L. Grubmüller and Various Upstream Contributors, *My agda-cubical Fork*. [Online]. Available: https://github.com/flgrubm/cubical