The Category of Iterative Sets in Cubical Agda

Master Thesis, Stockholm University

Fabian Lukas Grubmüller

Friday, 6 February 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

Tarski-style Universe

Tarski-style Universe

- A universe $\mathcal{U}$ of **codes**

Tarski-style Universe

- A universe $\mathcal{U}$ of **codes**
- A **decoding function** $\mathtt{El} : \mathcal{U} \to \mathrm{Type}$

Tarski-style Universe

- A universe $\mathcal{U}$ of **codes**
- A **decoding function** $\mathtt{El} : \mathcal{U} \to \mathrm{Type}$

Goal: A Universe of Sets

Tarski-style Universe

- A universe $\mathcal{U}$ of **codes**
- A **decoding function** $\mathtt{El} : \mathcal{U} \to \mathrm{Type}$

Goal: A Universe of Sets

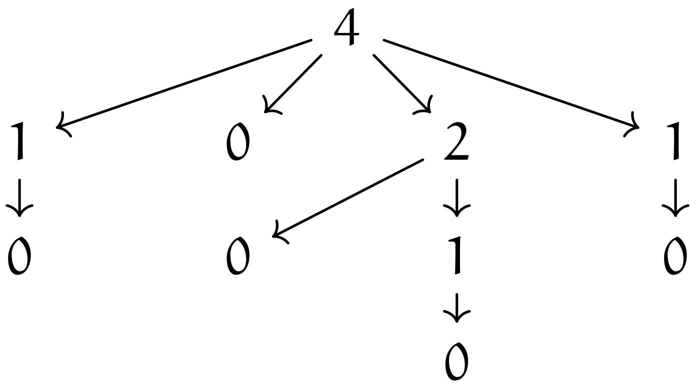
Idea: Viewing **sets** as **trees**:

- A universe $\mathcal{U}$ of **codes**
- A **decoding function** $\mathtt{El} : \mathcal{U} \to \mathrm{Type}$

Goal: A Universe of Sets

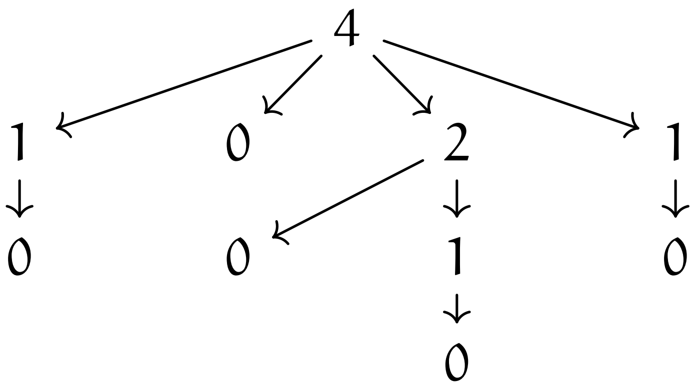
Idea: Viewing **sets** as **trees**:

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

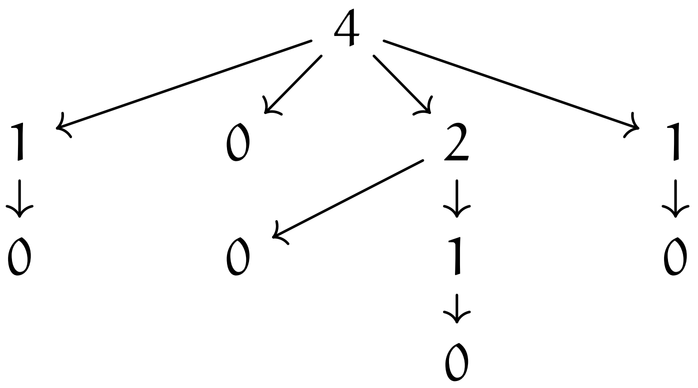


Iterative Multisets

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

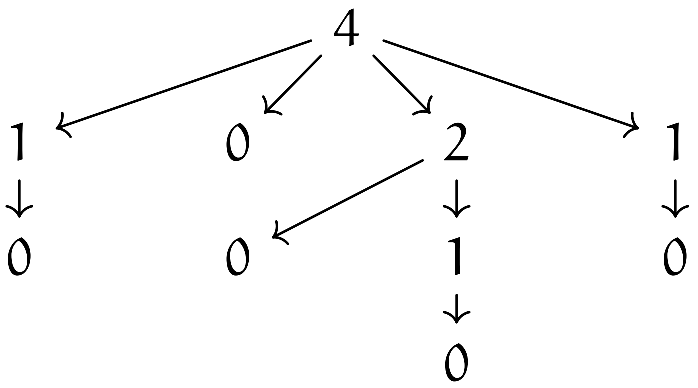


$$\{\{\emptyset\}, \emptyset, \{\emptyset, \{\emptyset\}\}, \{\emptyset\}\}$$



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

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



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

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 $A : \mathrm{Type}$ and $B : A \to \mathrm{Type}$ has elements

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

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

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

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

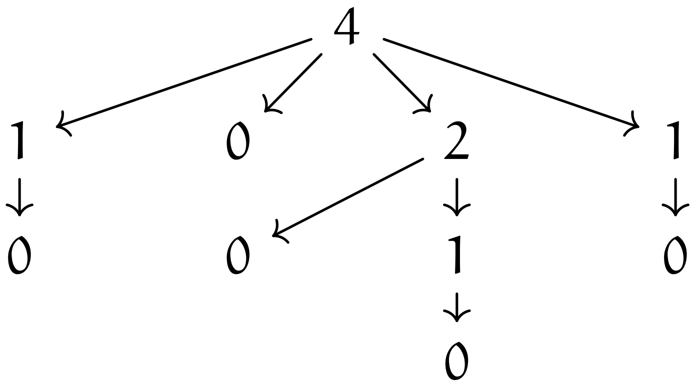
$$\mathrm{sup}(A, f) \quad \textbf{with} \quad A : \mathrm{Type}, \quad f : A \to V^\infty$$

Iterative Multisets

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

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

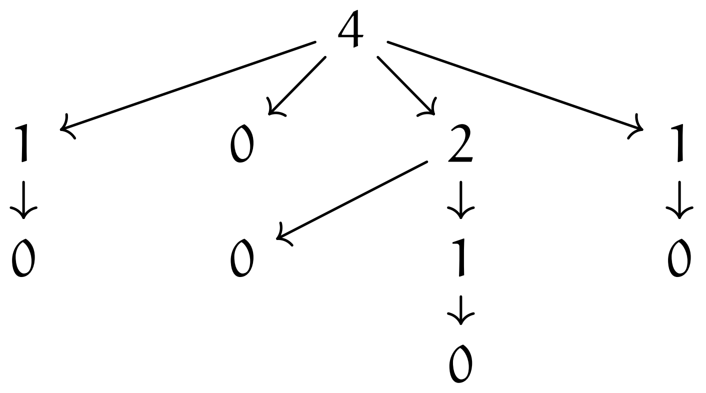
$$\{\{\emptyset\}, \emptyset, \{\emptyset, \{\emptyset\}\}, \{\emptyset\}\}$$



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

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

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



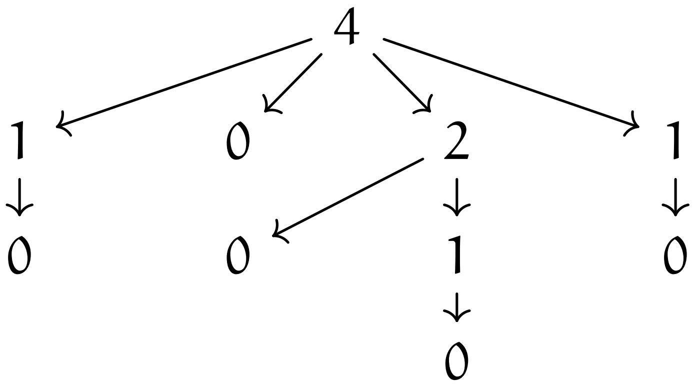
- **multisets** with **multiplicities** $(x \in^\infty (\mathrm{sup}(B, g)) := \mathrm{fiber}_g(x) := \sum_{b:B} g(b) \equiv_{V^\infty} x$

Iterative Multisets

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

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

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



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

Iterative Sets

Iterative Sets

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

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

Definition (Hereditarily) iterative sets:

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

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

- **decodings** similar: $\mathrm{El}(\sup(A, f), P) :\equiv A$,

Idea: for *sets*, fibers 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: $V^0 :\equiv \sum_{x:V^\infty} \mathsf{isIterativeSet}(x)$

- **decodings** similar: $\mathsf{El}(\mathsf{sup}(A, f), P) :\equiv A$, **multiplicities** $x \in^0 y$ **propositional**

Properties

Properties

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

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

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**

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**
- Universe **closed** under many standard **type formers**

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**
- Universe **closed** under many standard **type formers**
  - empty set, singleton sets, (un-) ordered pairs, natural numbers

Properties

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

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

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

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

- V^0 is a **set** $\rightsquigarrow$ all **decodings** are **sets**
- 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
- V^0 is a **category** closed under many standard **constructions**

3. Iterative Sets as a model of Dependent Type Theory

Categories with Families

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

Categories with Families

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** $\mathcal{T}\mathsf{y} : \mathcal{C}^{\mathsf{op}} \longrightarrow \mathbf{hSets}$

Categories with Families

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 $\Delta : \mathcal{C}$:

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

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

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

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

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

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

Total: ~4700 LOC

Naturality of Σ-structure

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

Challenges

Challenges

- using **“more cubical”** techniques

Challenges

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

Challenges

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

Challenges

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

Technical Limitations

Challenges

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

Technical Limitations

- documentation/discoverability

Challenges

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

Technical Limitations

- documentation/discoverability
- confusing goals

Challenges

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

Technical Limitations

- documentation/discoverability
- confusing goals
- poor error messages

Challenges

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

Technical Limitations

- documentation/discoverability
- confusing goals
- poor error messages

Outlook

Challenges

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

Technical Limitations

- documentation/discoverability
- confusing goals
- poor error messages

Outlook

- finish formalization

Challenges

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

Technical Limitations

- documentation/discoverability
- confusing goals
- poor error messages

Outlook

- finish formalization
- different model structure instead of CwF

Challenges

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

Technical Limitations

- documentation/discoverability
- confusing goals
- poor error messages

Outlook

- finish formalization
- different model structure instead of CwF

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.

5. Appendix

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

$\mathcal{V}^0$ 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:** $\mathrm{Tm}(\Gamma, A) \coloneqq \prod_{y:\,\mathrm{El}(\Gamma)} \mathrm{El}(A(y))$

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:** $S_\Gamma(A, B) \coloneqq \lambda x : \mathrm{El}(\Gamma). \quad \Sigma^0(A(x), \lambda a.B(x, a))$