Cubical Indexed Inductive Types

Evan Cavallo

Carnegie Mellon University

jww Robert Harper

Higher inductive types

Higher inductive types

- roll quotients and inductive types into one

Higher inductive types

- roll quotients and inductive types into one

$$\begin{aligned}
&\mathbf{data}\ \mathrm{int}\ \mathbf{where}\\
&\mid \mathrm{neg}(n : \mathrm{nat}) : \mathrm{int}\\
&\mid \mathrm{pos}(n : \mathrm{nat}) : \mathrm{int}\\
&\mid \mathrm{seg} : \mathrm{Id}_{\mathrm{int}}(\mathrm{neg}(0), \mathrm{pos}(0))
\end{aligned}$$

Higher inductive types

- roll quotients and inductive types into one

$$\begin{aligned}&\textbf{data}\ \mathrm{int}\ \textbf{where}\\&|\ \mathrm{neg}(n:\mathrm{nat}):\mathrm{int}\\&|\ \mathrm{pos}(n:\mathrm{nat}):\mathrm{int}\\&|\ \mathrm{seg}:\mathrm{Id}_{\mathrm{int}}(\mathrm{neg}(0),\mathrm{pos}(0))\end{aligned}$$

- Useful generality beyond ordinary quotients

Higher inductive types

- roll quotients and inductive types into one

$$\begin{aligned}&\textbf{data}\ \mathbf{int}\ \textbf{where}\\ &\mid \mathbf{neg}(n : \mathbf{nat}) : \mathbf{int}\\ &\mid \mathbf{pos}(n : \mathbf{nat}) : \mathbf{int}\\ &\mid \mathbf{seg} : \mathbf{Id}_{\mathbf{int}}(\mathbf{neg}(0), \mathbf{pos}(0))\end{aligned}$$

- Useful generality beyond ordinary quotients
  - → In ordinary TT: Cauchy reals, QIITs, . . .

Higher inductive types

- roll quotients and inductive types into one

$$\begin{aligned}&\textbf{data int where}\\&\mid \mathbf{neg}(n : \mathbf{nat}) : \mathbf{int}\\&\mid \mathbf{pos}(n : \mathbf{nat}) : \mathbf{int}\\&\mid \mathbf{seg} : \mathbf{Id}_{\mathbf{int}}(\mathbf{neg}(0), \mathbf{pos}(0))\end{aligned}$$

- Useful generality beyond ordinary quotients
  - → In ordinary TT: Cauchy reals, QIITs, . . .
  - → In higher-d TT: truncations, . . .

Higher inductive types: what are they?

Higher inductive types: what are they?

- HoTT: idea, many examples
  - → circles, pushouts, . . . ("ordinary" quotients)
  - → truncations, localizations (higher inductive types)
  - → Cauchy reals (higher inductive-inductive types)

Higher inductive types: what are they?

- HoTT: idea, many examples
  - → circles, pushouts, . . . ("ordinary" quotients)
  - → truncations, localizations (higher inductive types)
  - → Cauchy reals (higher inductive-inductive types)
- **no precise syntactic schema**
  - → what forms can constructors take?
  - → deriving eliminators

Higher inductive types: what are they?

- HoTT: idea, many examples
  - → circles, pushouts, . . . ("ordinary" quotients)
  - → truncations, localizations (higher inductive types)
  - → Cauchy reals (higher inductive-inductive types)
- **no precise syntactic schema**
  - → what forms can constructors take?
  - → deriving eliminators
- **missing semantics for many**
  - → how do programs with HITs compute?
  - → which HITs exist in denotational models? (pushouts?)

Higher inductive types: what are they?

- advances in syntax
  - → Sojakova 2014: W-suspensions
  - → Basold, Geuvers, & van der Weide 2017: 1-d HITs
  - → Dybjer & Moeneclaey 2017: 2-d HITs
  - → Kaposi & Kovács 2018: n-d HIITs

Higher inductive types: what are they?

- advances in semantics
  - → Lumsdaine & Shulman 2017: simplicial model cats
    - → syntax? HITs with parameters?
  - → Dybjer & Moeneclaey 2017: groupoid model
  - → Altenkirch, Kaposi, & Kovács 2019: w/ UIP

Higher inductive types: what are they?

- advances in semantics
  - → Lumsdaine & Shulman 2017: simplicial model cats
    - → syntax? HITs with parameters?
  - → Dybjer & Moeneclaey 2017: groupoid model
  - → Altenkirch, Kaposi, & Kovács 2019: w/ UIP

- the "edge of understanding": syntax + semantics
  - ⓐ DM, AKK (truncated)
  - ⓑ **cubical type theories**

Cubical type theories

Cubical type theories

- Motive: higher type theories w/ computational meaning

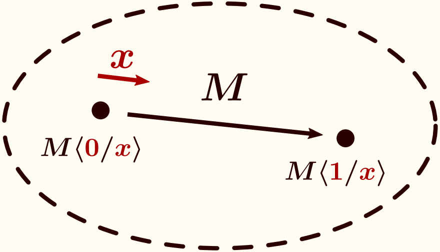
Cubical type theories

- Motive: higher type theories w/ computational meaning
- Several varieties:
  - → De Morgan cubes
    Cohen, Coquand, Huber, & Mörtberg 2015
  - → Cartesian cubes
    Angiuli, Favonia, & Harper 2018
    Angiuli, Brunerie, Coquand, Favonia, Licata, & Harper 2019
  - → Substructural cubes
    Bezem, Coquand, & Huber 2013&2017

Cubical type theories

- Motive: higher type theories w/ computational meaning

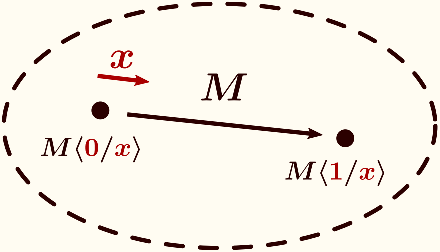
$$M \in A\ [\Psi, x]$$



Cubical type theories

- Motive: higher type theories w/ computational meaning

$$M \in A \; [\Psi, x]$$

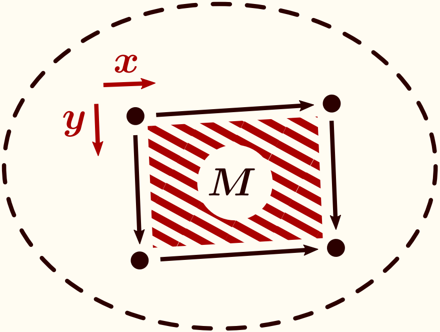


$$\lambda^{\mathbb{I}} x.M \in \mathrm{Path}_{x.A}(M\langle 0/x \rangle, M\langle 1/x \rangle) \; [\Psi]$$

Cubical type theories

- Motive: higher type theories w/ computational meaning

$$M \in A\ [\Psi, x, y]$$



Cubical type theories: why?

Cubical type theories: why?

- Developed to give computational meaning to HoTT

Cubical type theories: why?

- ↻ Developed to give computational meaning to HoTT
  - → also: constructive denotational models

Cubical type theories: why?

- Developed to give computational meaning to HoTT
  - → also: constructive denotational models
  - → also: *any* models for many HITs

Cubical type theories: why?

- **Developed to give computational meaning to HoTT**
  - → also: constructive denotational models
  - → also: *any* models for many HITs

- **Useful for practical formalization**
  - → early motivation: Licata & Brunerie 2015

Cubical type theories: why?

- Developed to give computational meaning to HoTT
  - → also: constructive denotational models
  - → also: *any* models for many HITs
- Useful for practical formalization
  - → early motivation: Licata & Brunerie 2015
- For HoTT purists: laboratory of higher ideas

Cubical inductive types: syntax

Cubical inductive types: syntax

- Higher constructors via dimension arguments

$$\mathbf{data}\ \mathrm{int}\ \mathbf{where}$$
$$\mid \mathrm{neg}(n : \mathrm{nat}) : \mathrm{int}$$
$$\mid \mathrm{pos}(n : \mathrm{nat}) : \mathrm{int}$$
$$\mid \mathrm{seg}(x : \mathbb{I}) : \mathrm{int}\ [x = 0 \hookrightarrow \mathrm{neg}(0) \mid x = 1 \hookrightarrow \mathrm{pos}(0)]$$

Cubical inductive types: syntax

- Higher constructors via dimension arguments

$$\mathbf{data}\ \mathrm{int}\ \mathbf{where}$$
$$\mid \mathrm{neg}(n : \mathrm{nat}) : \mathrm{int}$$
$$\mid \mathrm{pos}(n : \mathrm{nat}) : \mathrm{int}$$
$$\mid \mathrm{seg}(x : \mathbb{I}) : \mathrm{int}\ [x = 0 \hookrightarrow \mathrm{neg}(0) \mid x = 1 \hookrightarrow \mathrm{pos}(0)]$$

- Path constructors live in the inductive type

Cubical inductive types: syntax

- Higher constructors via dimension arguments

$$\mathbf{data}\ \mathrm{int}\ \mathbf{where}$$
$$\mid \mathrm{neg}(n : \mathrm{nat}) : \mathrm{int}$$
$$\mid \mathrm{pos}(n : \mathrm{nat}) : \mathrm{int}$$
$$\mid \mathrm{seg}(x : \mathbb{I}) : \mathrm{int}\ [x = 0 \hookrightarrow \mathrm{neg}(0) \mid x = 1 \hookrightarrow \mathrm{pos}(0)]$$

- Path constructors live in the inductive type
- Scales well to >1-d (spheres, torus, . . .)

Cubical inductive types: syntax

- **Elimination:** pattern matching + coherence reqs

Cubical inductive types: syntax

- **Elimination:** pattern matching + coherence reqs

$$\mathbf{elim}\ Z$$
$$\mid \mathrm{neg}(n) \to M_- \in P(\mathrm{neg}(n))$$
$$\mid \mathrm{pos}(n) \to M_+ \in P(\mathrm{pos}(n))$$
$$\mid \mathrm{seg}(x) \to M_0 \in P(\mathrm{seg}(x))$$

Cubical inductive types: syntax

- **Elimination:** pattern matching + coherence reqs

$$\begin{aligned}&\mathbf{elim}\ Z\\ &\mid \mathrm{neg}(n) \to M_- \in P(\mathrm{neg}(n))\\ &\mid \mathrm{pos}(n) \to M_+ \in P(\mathrm{pos}(n))\\ &\mid \mathrm{seg}(x) \to M_0 \in P(\mathrm{seg}(x))\end{aligned} \quad \begin{pmatrix} M_0\langle 0/x\rangle = M_-[0/n] \\ M_0\langle 1/x\rangle = M_+[0/n] \end{pmatrix}$$

Cubical inductive types: syntax

- **Elimination:** pattern matching + coherence reqs

$$\begin{aligned}&\mathbf{elim}\ Z\\ &\mid \mathrm{neg}(n) \to M_- \in P(\mathrm{neg}(n))\\ &\mid \mathrm{pos}(n) \to M_+ \in P(\mathrm{pos}(n))\\ &\mid \mathrm{seg}(x) \to M_0 \in P(\mathrm{seg}(x))\end{aligned} \qquad \begin{pmatrix} M_0\langle 0/x\rangle = M_-[0/n] \\ M_0\langle 1/x\rangle = M_+[0/n]\end{pmatrix}$$

- reduces when applied to a constructor

$$\mathbf{elim}\ (\mathrm{seg}(y))\ [\cdots] \ = \ M_0\langle y/x\rangle$$

Cubical inductive types: syntax

- Dealing with higher dimensions

Cubical inductive types: syntax

- Dealing with higher dimensions
  - → HITs with >1-d constructors

$$\textbf{data}\ \text{sphere}\ \textbf{where}$$
$$|\ \text{base} : \text{sphere}$$
$$|\ \text{surf}(x : \mathbb{I}, y : \mathbb{I}) : \text{sphere}$$
$$[x = 0 \mid x = 1 \mid y = 0 \mid y = 1 \hookrightarrow \text{base}]$$

Cubical inductive types: syntax

- Dealing with higher dimensions
  - → HITs with >1-d constructors

$$\textbf{data}\ \text{sphere}\ \textbf{where}$$
$$|\ \text{base} : \text{sphere}$$
$$|\ \text{surf}(x : \mathbb{I}, y : \mathbb{I}) : \text{sphere}$$
$$[x = 0 \mid x = 1 \mid y = 0 \mid y = 1 \hookrightarrow \text{base}]$$

  - → pattern-matching on two HITs at once

$$\textbf{elim}\ Z_1, Z_2$$
$$|\ \text{seg}(x), \text{seg}(y) \to M_{00}$$
$$\vdots$$

Cubical inductive types: syntax

- C & Harper: schema for indexed cubical HITs

$$
\begin{aligned}
&\textbf{data } \mathrm{X} \textbf{ where}\\
&\mid \mathrm{constr}_1\\
&\quad (a_1 : A_1) \cdots (a_k : A_k)\\
&\quad (b_1 : \mathrm{B}_1) \cdots (b_k : \mathrm{B}_k)\\
&\quad (x_1, \ldots, x_\ell : \mathrm{dim})\\
&\quad [r_1 = r_1' \hookrightarrow \mathrm{M}_1 \mid \cdots \mid r_j = r_j' \hookrightarrow \mathrm{M}_j]\\
&\mid \mathrm{constr}_n \cdots
\end{aligned}
$$

Cubical inductive types: syntax

- C & Harper: schema for indexed cubical HITs

$$
\begin{aligned}
&\textbf{data } \mathrm{X} \textbf{ where}\\
&\mid \mathrm{constr}_1\\
&\quad (a_1 : A_1) \cdots (a_k : A_k) \leftarrow \text{non-recursive}\\
&\quad (b_1 : \mathrm{B}_1) \cdots (b_k : \mathrm{B}_k) \leftarrow \text{recursive}\\
&\quad (x_1, \ldots, x_\ell : \mathrm{dim})\\
&\quad [r_1 = r_1' \hookrightarrow \mathrm{M}_1 \mid \cdots \mid r_j = r_j' \hookrightarrow \mathrm{M}_j]\\
&\mid \mathrm{constr}_n \cdots
\end{aligned}
$$

Cubical inductive types: syntax

- C & Harper: schema for indexed cubical HITs

$$
\begin{aligned}
&\textbf{data X where}\\
&\mid \text{constr}_1\\
&\quad (a_1 : A_1) \cdots (a_k : A_k) \leftarrow \text{non-recursive}\\
&\quad (b_1 : \mathrm{B}_1) \cdots (b_k : \mathrm{B}_k) \leftarrow \text{recursive}\\
&\quad (x_1, \ldots, x_\ell : \text{dim})\\
&\quad [r_1 = r'_1 \hookrightarrow \mathrm{M}_1 \mid \cdots \mid r_j = r'_j \hookrightarrow \mathrm{M}_j]\\
&\mid \text{constr}_n \cdots
\end{aligned}
$$

$$\mathrm{B} ::= \mathbf{X} \mid (a : A) \to \mathrm{B}$$

Cubical inductive types: syntax

- C & Harper: schema for indexed cubical HITs

$$\begin{aligned}
&\textbf{data X where}\\
&\mid \text{constr}_1\\
&\quad (a_1 : A_1)\cdots(a_k : A_k) \leftarrow \text{non-recursive}\\
&\quad (b_1 : \mathrm{B}_1)\cdots(b_k : \mathrm{B}_k) \leftarrow \text{recursive}\\
&\quad (x_1, \ldots, x_\ell : \text{dim}) \leftarrow \text{dimensions}\\
&\quad [r_1 = r_1' \hookrightarrow \mathrm{M}_1 \mid \cdots \mid r_j = r_j' \hookrightarrow \mathrm{M}_j] \leftarrow \text{boundary}\\
&\mid \text{constr}_n \cdots
\end{aligned}$$

$$\mathrm{B} ::= \mathbf{X} \mid (a : A) \to \mathrm{B}$$

Cubical inductive types: syntax

- C & Harper: schema for indexed cubical HITs

$$\begin{array}{l}
\textbf{data } \mathbf{X} \textbf{ where} \\
\mid \mathrm{constr}_1 \\
\quad (a_1 : A_1) \cdots (a_k : A_k) \leftarrow \text{non-recursive} \\
\quad (b_1 : \mathrm{B}_1) \cdots (b_k : \mathrm{B}_k) \leftarrow \text{recursive} \\
\quad (x_1, \ldots, x_\ell : \mathrm{dim}) \leftarrow \text{dimensions} \\
\quad [r_1 = r_1' \hookrightarrow \mathrm{M}_1 \mid \cdots \mid r_j = r_j' \hookrightarrow \mathrm{M}_j] \leftarrow \text{boundary} \\
\mid \mathrm{constr}_n \cdots
\end{array}$$

$$\mathrm{B} ::= \mathbf{X} \mid (a : A) \to \mathrm{B}$$

$$\mathrm{M} ::= b \mid \mathrm{constr}_i(\vec{M}, \vec{\mathrm{M}}, \vec{r}) \mid \mathrm{hcom}(\cdots) \mid \lambda a.\mathrm{M} \mid \mathrm{M} M$$

Cubical inductive types: semantics

Cubical inductive types: semantics

- Ingredients of a cubical type

Cubical inductive types: semantics

- Ingredients of a cubical type
  - → Values

Cubical inductive types: semantics

- Ingredients of a cubical type
  - → Values

$$\langle M, N \rangle \in A \times B$$
$$\lambda a.N \in A \to B$$
$$\lambda^{\mathbb{I}} x.N \in \mathrm{Path}_{x.A}(M_0, M_1)$$
$$?? \in \mathrm{int}$$

Cubical inductive types: semantics

- Ingredients of a cubical type
  - → Values

$$\begin{aligned} \langle M, N \rangle &\in A \times B \\ \lambda a.N &\in A \to B \\ \lambda^{\mathbb{I}} x.N &\in \mathrm{Path}_{x.A}(M_0, M_1) \\ ?? &\in \mathrm{int} \end{aligned}$$

  - → Coercion and composition operations

Interlude: semantics of cubical type theory

Interlude: semantics of cubical type theory

- **Coercion:** all constructions respect paths

Interlude: semantics of cubical type theory

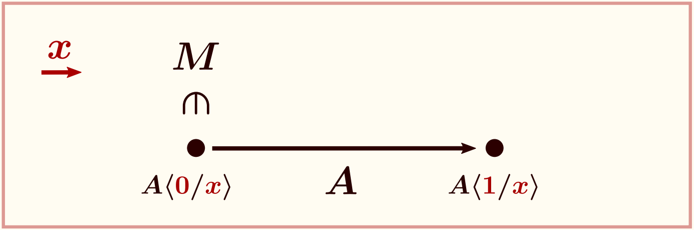
- Coercion: all constructions respect paths

$$\frac{A \ \mathsf{type}\ [\Psi, x] \qquad M \in A\langle r/x\rangle\ [\Psi]}{\mathsf{coe}^{r \rightsquigarrow s}_{x.A}(M) \in A\langle s/x\rangle\ [\Psi]}$$

Interlude: semantics of cubical type theory

- Coercion: all constructions respect paths

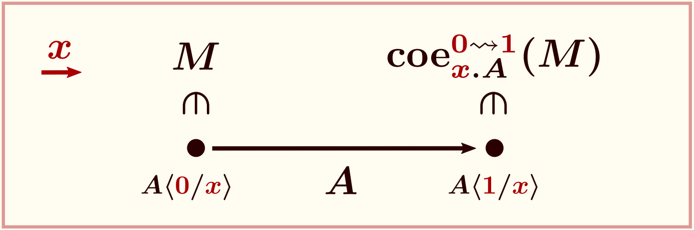
$$\frac{A \text{ type } [\Psi, x] \qquad M \in A\langle r/x \rangle\ [\Psi]}{\mathrm{coe}_{x.A}^{r \rightsquigarrow s}(M) \in A\langle s/x \rangle\ [\Psi]}$$



Interlude: semantics of cubical type theory

- **Coercion:** all constructions respect paths

$$\frac{A \ \mathsf{type}\ [\Psi, x] \qquad M \in A\langle r/x\rangle\ [\Psi]}{\mathsf{coe}^{r \rightsquigarrow s}_{x.A}(M) \in A\langle s/x\rangle\ [\Psi]}$$



Interlude: semantics of cubical type theory

- ⥁ Coercion: all constructions respect paths
  - → Evaluate by cases on the type line

Interlude: semantics of cubical type theory

- ⟳ Coercion: all constructions respect paths

  → Evaluate by cases on the type line

$$\mathbf{coe}^{r \rightsquigarrow s}_{x.A \times B}(\langle M, N \rangle)$$

$$\longmapsto$$

$$\langle \mathbf{coe}^{r \rightsquigarrow s}_{x.A}(M), \mathbf{coe}^{r \rightsquigarrow s}_{x.B}(N) \rangle$$

Interlude: semantics of cubical type theory

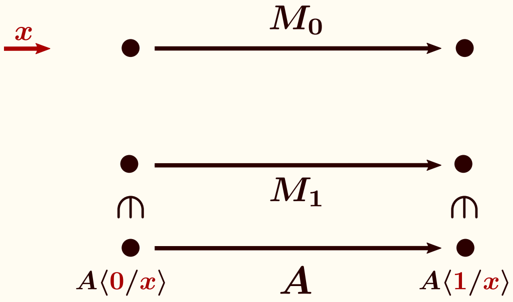
- ↻ Coercion: all constructions respect paths
  - → Evaluate by cases on the type line

$$\mathbf{coe}^{r \rightsquigarrow s}_{x.\mathrm{Path}_{y.A}(M_0,M_1)}(\lambda^{\mathbb{I}} y.N) \longmapsto ??$$

Interlude: semantics of cubical type theory

- **Coercion:** all constructions respect paths
  - → Evaluate by cases on the type line

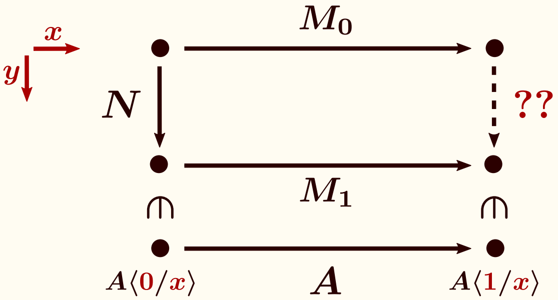
$$\mathbf{coe}^{r\rightsquigarrow s}_{x.\mathrm{Path}_{y.A}(M_0,M_1)}(\lambda^{\mathbb{I}} y.N) \longmapsto ??$$



Interlude: semantics of cubical type theory

- Coercion: all constructions respect paths
  - → Evaluate by cases on the type line

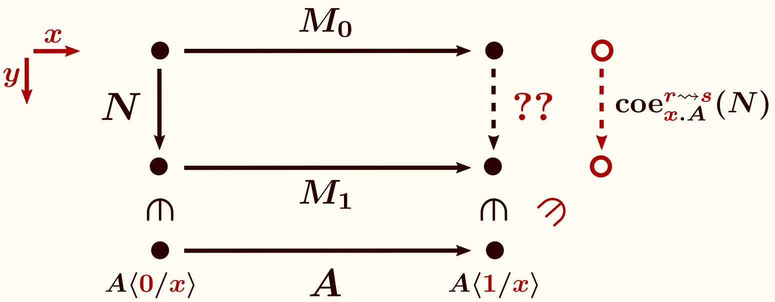
$$\mathbf{coe}^{r \rightsquigarrow s}_{x.\mathrm{Path}_{y.A}(M_0,M_1)}(\lambda^{\mathbb{I}} y.N) \longmapsto ??$$



Interlude: semantics of cubical type theory

- Coercion: all constructions respect paths
  - → Evaluate by cases on the type line

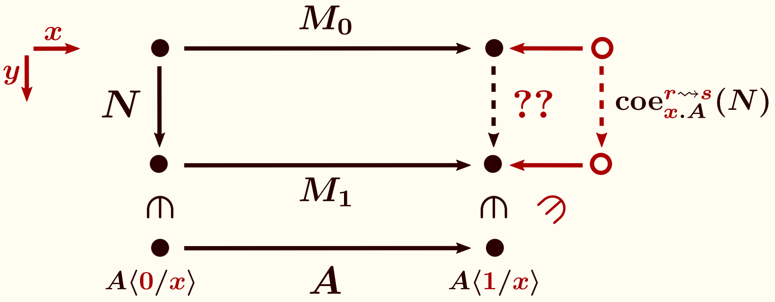
$$\mathbf{coe}^{r \rightsquigarrow s}_{x.\mathrm{Path}_{y.A}(M_0, M_1)}(\lambda^{\mathbb{I}} y. N) \longmapsto ??$$



Interlude: semantics of cubical type theory

- **Coercion:** all constructions respect paths
  - → Evaluate by cases on the type line

$$\mathbf{coe}^{r \rightsquigarrow s}_{x.\mathtt{Path}_{y.A}(M_0, M_1)}(\lambda^{\mathbb{I}} y.N) \longmapsto ??$$



Interlude: semantics of cubical type theory

- **Composition:** strengthening the induction hypothesis

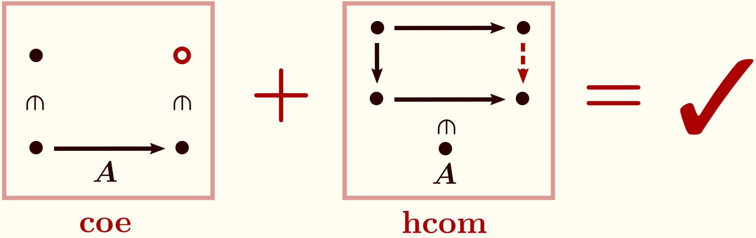
Interlude: semantics of cubical type theory

- **Composition:** strengthening the induction hypothesis
  - ⓐ add homogeneous composition

Interlude: semantics of cubical type theory

⟳ **Composition:** strengthening the induction hypothesis

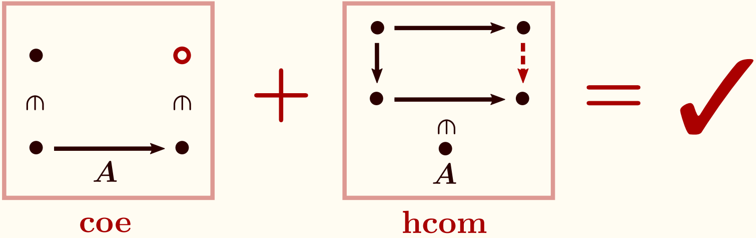
ⓐ add homogeneous composition



Interlude: semantics of cubical type theory

↻ Composition: strengthening the induction hypothesis

ⓐ add homogeneous composition



(ⓑ generalize **coe** to heterogeneous composition)

Cubical inductive types: semantics

Cubical inductive types: semantics

- What are the values of a higher inductive type?

Cubical inductive types: semantics

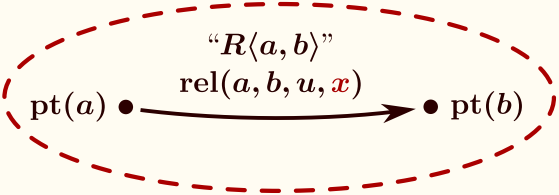
- What are the values of a higher inductive type?

$$A : \mathcal{U}, R : A \times A \to \mathcal{U} \vdash \textbf{data}\ \text{quo}\ \textbf{where}$$
$$\mid \text{pt}(a : A) : \text{quo}$$
$$\mid \text{rel}(a : A, b : A, u : R\langle a, b\rangle, x : \mathbb{I}) : \text{quo}$$
$$[x = 0 \hookrightarrow \text{pt}(a) \mid x = 1 \hookrightarrow \text{pt}(b)]$$

Cubical inductive types: semantics

- What are the values of a higher inductive type?

$$A : \mathcal{U}, R : A \times A \to \mathcal{U} \vdash \textbf{data}\ \mathrm{quo}\ \textbf{where}$$
$$\mid \mathrm{pt}(a : A) : \mathrm{quo}$$
$$\mid \mathrm{rel}(a : A, b : A, u : R\langle a, b\rangle, x : \mathbb{I}) : \mathrm{quo}$$
$$[x = 0 \hookrightarrow \mathrm{pt}(a) \mid x = 1 \hookrightarrow \mathrm{pt}(b)]$$



Cubical inductive types: semantics

- What are the values of a higher inductive type?

$$A : \mathcal{U}, R : A \times A \to \mathcal{U} \vdash \textbf{data}\ \text{quo}\ \textbf{where}$$
$$\mid \text{pt}(a : A) : \text{quo}$$
$$\mid \text{rel}(a : A, b : A, u : R\langle a, b\rangle, x : \mathbb{I}) : \text{quo}$$
$$[x = 0 \hookrightarrow \text{pt}(a) \mid x = 1 \hookrightarrow \text{pt}(b)]$$

$$\text{pt}(M)\ \textbf{val} \qquad \text{rel}(M, N, P, x)\ \textbf{val}$$
$$\left(\begin{array}{l} \text{rel}(M, N, P, 0) \longmapsto \text{pt}(M) \\ \text{rel}(M, N, P, 1) \longmapsto \text{pt}(N) \end{array}\right)$$

Cubical inductive types: semantics

- What are the values of a higher inductive type?

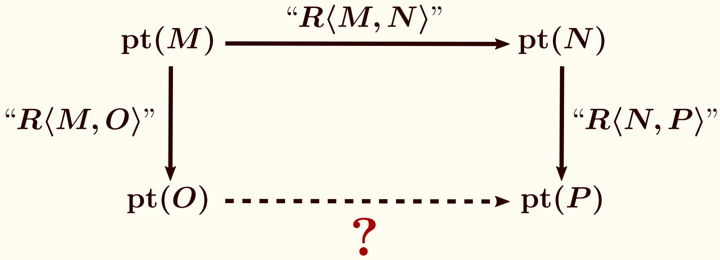
$$A : \mathcal{U}, R : A \times A \to \mathcal{U} \vdash \textbf{data}\ \mathrm{quo}\ \textbf{where}$$
$$\mid \mathrm{pt}(a : A) : \mathrm{quo}$$
$$\mid \mathrm{rel}(a : A, b : A, u : R\langle a, b\rangle, x : \mathbb{I}) : \mathrm{quo}$$
$$[x = 0 \hookrightarrow \mathrm{pt}(a) \mid x = 1 \hookrightarrow \mathrm{pt}(b)]$$

$$\mathrm{pt}(M)\ \textbf{val} \qquad \frac{\mathrm{rel}(M, N, P, x)\ \textbf{val}}{\begin{pmatrix} \mathrm{rel}(M, N, P, 0) \longmapsto \mathrm{pt}(M) \\ \mathrm{rel}(M, N, P, 1) \longmapsto \mathrm{pt}(N) \end{pmatrix}}$$

- Can we implement coercion and composition?

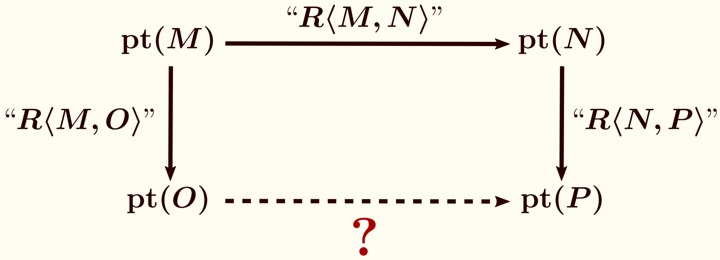
Cubical inductive types: semantics

- What are the values of a higher inductive type?



Cubical inductive types: semantics

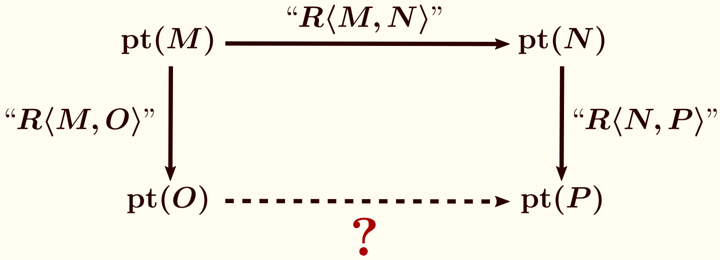
- What are the values of a higher inductive type?



- Unless R is symmetric and transitive, ? may not exist

Cubical inductive types: semantics

- What are the values of a higher inductive type?



- Unless R is symmetric and transitive, ? may not exist

  ⇒ Must revise our choice of values

Cubical inductive types: semantics

- Idea: freely add homogeneous composition values

Cubical inductive types: semantics

- Idea: freely add homogeneous composition values

$$\mathrm{pt}(M)\ \mathbf{val} \qquad \qquad \mathrm{rel}(M, N, P, x)\ \mathbf{val}$$

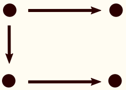
$$\begin{pmatrix} \mathrm{rel}(M, N, P, 0) \longmapsto \mathrm{pt}(M) \\ \mathrm{rel}(M, N, P, 1) \longmapsto \mathrm{pt}(N) \end{pmatrix}$$

Cubical inductive types: semantics

- Idea: freely add homogeneous composition values

$$\mathrm{pt}(M)\ \mathbf{val} \qquad \mathrm{rel}(M,N,P,x)\ \mathbf{val}$$

$$\begin{pmatrix} \mathrm{rel}(M,N,P,0) \longmapsto \mathrm{pt}(M) \\ \mathrm{rel}(M,N,P,1) \longmapsto \mathrm{pt}(N) \end{pmatrix}$$

hcom () **val**

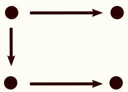
(+ boundary reductions)

Cubical inductive types: semantics

- Idea: freely add homogeneous composition values

$$\mathrm{pt}(M)\ \textbf{val} \qquad\qquad \mathrm{rel}(M,N,P,x)\ \textbf{val}$$

$$\left(\begin{array}{l}\mathrm{rel}(M,N,P,0) \longmapsto \mathrm{pt}(M)\\ \mathrm{rel}(M,N,P,1) \longmapsto \mathrm{pt}(N)\end{array}\right)$$

hcom () **val**

(+ boundary reductions)

- Eliminator maps $\mathbf{hcom}$ values to $\mathbf{hcom}$s in the target

Cubical inductive types: semantics

- Implement coercion by cases

Cubical inductive types: semantics

- Implement coercion by cases

$$\mathbf{coe}^{r \rightsquigarrow s}_{x.\mathbf{quo}(A,R)}(\mathbf{pt}(M)) \longmapsto \mathbf{pt}(\mathbf{coe}^{r \rightsquigarrow s}_{x.A}(M))$$

Cubical inductive types: semantics

- Implement coercion by cases

$$\mathbf{coe}^{r \rightsquigarrow s}_{x.\mathbf{quo}(A,R)}(\mathbf{pt}(M)) \longmapsto \mathbf{pt}(\mathbf{coe}^{r \rightsquigarrow s}_{x.A}(M))$$

$$\mathbf{coe}^{r \rightsquigarrow s}_{x.\mathbf{quo}(A,R)}(\mathbf{rel}(M, N, P, y)) \longmapsto \mathbf{rel}(\mathbf{coe}^{r \rightsquigarrow s}_{x.A}(M), \cdots, \cdots, y)$$

Cubical inductive types: semantics

- Implement coercion by cases

$$\mathbf{coe}^{r \rightsquigarrow s}_{x.\mathbf{quo}(A,R)}(\mathbf{pt}(M)) \longmapsto \mathbf{pt}(\mathbf{coe}^{r \rightsquigarrow s}_{x.A}(M))$$

$$\mathbf{coe}^{r \rightsquigarrow s}_{x.\mathbf{quo}(A,R)}(\mathbf{rel}(M, N, P, y)) \longmapsto \mathbf{rel}(\mathbf{coe}^{r \rightsquigarrow s}_{x.A}(M), \cdots, \cdots, y)$$

$$\mathbf{coe}^{r \rightsquigarrow s}_{x.\mathbf{quo}(A,R)}(\mathbf{hcom}(\cdots)) \longmapsto \mathbf{hcom}(\mathbf{coe}^{r \rightsquigarrow s}_{x.\mathbf{quo}(A,R)}(\cdots))$$

Cubical inductive types: semantics

- Implement coercion by cases

$$\mathrm{coe}^{r\rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(\mathrm{pt}(M)) \longmapsto \mathrm{pt}(\mathrm{coe}^{r\rightsquigarrow s}_{x.A}(M))$$

$$\mathrm{coe}^{r\rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(\mathrm{rel}(M,N,P,y)) \longmapsto \mathrm{rel}(\mathrm{coe}^{r\rightsquigarrow s}_{x.A}(M),\cdots,\cdots,y)$$

$$\mathrm{coe}^{r\rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(\mathrm{hcom}(\cdots)) \longmapsto \mathrm{hcom}(\mathrm{coe}^{r\rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(\cdots))$$

- Why not free coercion values?

$$\frac{A \in \mathcal{U}\ [\Psi, x] \qquad R \in A \times A \to \mathcal{U}\ [\Psi, x] \qquad M \in \mathrm{quo}(A,R)\langle r/x\rangle\ [\Psi]}{\mathrm{coe}^{r\rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(M) \in \mathrm{quo}(A,R)\langle s/x\rangle\ [\Psi]}$$

Cubical inductive types: semantics

- Implement coercion by cases

$$\mathrm{coe}^{r\rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(\mathrm{pt}(M)) \longmapsto \mathrm{pt}(\mathrm{coe}^{r\rightsquigarrow s}_{x.A}(M))$$

$$\mathrm{coe}^{r\rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(\mathrm{rel}(M,N,P,y)) \longmapsto \mathrm{rel}(\mathrm{coe}^{r\rightsquigarrow s}_{x.A}(M),\cdots,\cdots,y)$$

$$\mathrm{coe}^{r\rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(\mathrm{hcom}(\cdots)) \longmapsto \mathrm{hcom}(\mathrm{coe}^{r\rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(\cdots))$$

- Why not free coercion values?

$$\frac{\boxed{A \in \mathcal{U}\ [\Psi, x]} \qquad \boxed{R \in A \times A \to \mathcal{U}\ [\Psi, x]} \qquad M \in \mathrm{quo}(A,R)\langle r/x\rangle\ [\Psi]}{\mathrm{coe}^{r\rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(M) \in \mathrm{quo}(A,R)\langle s/x\rangle\ [\Psi]}$$

size trouble!

Cubical inductive types: semantics

- Implement coercion by cases

$$\mathrm{coe}^{r \rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(\mathrm{pt}(M)) \longmapsto \mathrm{pt}(\mathrm{coe}^{r \rightsquigarrow s}_{x.A}(M))$$

$$\mathrm{coe}^{r \rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(\mathrm{rel}(M, N, P, y)) \longmapsto \mathrm{rel}(\mathrm{coe}^{r \rightsquigarrow s}_{x.A}(M), \cdots, \cdots, y)$$

$$\mathrm{coe}^{r \rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(\mathrm{hcom}(\cdots)) \longmapsto \mathrm{hcom}(\mathrm{coe}^{r \rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(\cdots))$$

- Why not free coercion values?

$$\frac{\boxed{A \in \mathcal{U}\ [\Psi, x]} \quad \boxed{R \in A \times A \to \mathcal{U}\ [\Psi, x]} \quad M \in \mathrm{quo}(A, R)\langle r/x \rangle\ [\Psi]}{\mathrm{coe}^{r \rightsquigarrow s}_{x.\mathrm{quo}(A,R)}(M) \in \mathrm{quo}(A, R)\langle s/x \rangle\ [\Psi]}$$

size trouble!

→ cf. Lumsdaine & Shulman

Cubical inductive types: semantics

Theorem (Canonicity).
Any term in a HIT evaluates to a value.

Cubical inductive types: semantics

Theorem (Canonicity).
Any term in a HIT evaluates to a value.

- Are we satisfied with our choice of values?

Cubical inductive types: semantics

Theorem (Canonicity).
Any term in a HIT evaluates to a value.

- Are we satisfied with our choice of values?

  → any $\mathbf{int}$ evaluates to $\mathbf{pos}$, $\mathbf{neg}$, $\mathbf{seg}$, or $\mathbf{hcom}$

Cubical inductive types: semantics

Theorem (Canonicity).
Any term in a HIT evaluates to a value.

- Are we satisfied with our choice of values?
  - → any $\mathbf{int}$ evaluates to $\mathbf{pos}$, $\mathbf{neg}$, $\mathbf{seg}$, or $\mathbf{hcom}$
  - → Angiuli, Favonia, & Harper:
    zero-d compositions can be excluded

Cubical inductive types: semantics

Theorem (Canonicity).
Any term in a HIT evaluates to a value.

⥀ Are we satisfied with our choice of values?

→ any $\mathbf{int}$ evaluates to $\mathbf{pos}$, $\mathbf{neg}$, $\mathbf{seg}$, or $\mathbf{hcom}$

→ Angiuli, Favonia, & Harper:
zero-d compositions can be excluded

→ any zero-d $\mathbf{int}$ evaluates to $\mathbf{pos}$ or $\mathbf{neg}$ ✓

Indexed inductive types

Indexed inductive types

- Simultaneously inductively defined family of types

Indexed inductive types

- Simultaneously inductively defined family of types
  - → Vectors of a given length

$$A : \mathcal{U} \vdash \textbf{data}\ \mathbf{vec}(n : \mathbf{nat})\ \textbf{where}$$
$$\mid \mathbf{nil} : \mathbf{vec}(0)$$
$$\mid \mathbf{cons}(n : \mathbf{nat}, a : A, v : \mathbf{vec}(n)) : \mathbf{vec}(n+1)$$

Indexed inductive types

- Simultaneously inductively defined family of types
  - → Vectors of a given length

$$A : \mathcal{U} \vdash \textbf{data}\ \mathrm{vec}(n : \mathrm{nat})\ \textbf{where}$$
$$\mid \mathrm{nil} : \mathrm{vec}(0)$$
$$\mid \mathrm{cons}(n : \mathrm{nat}, a : A, v : \mathrm{vec}(n)) : \mathrm{vec}(n+1)$$

  - → Identity types

$$A : \mathcal{U} \vdash \textbf{data}\ \mathrm{id}(a_0 : A, a_1 : A)\ \textbf{where}$$
$$\mid \mathrm{refl}(a : A) : \mathrm{id}(a, a)$$

Indexed inductive types

- What are the values of an indexed inductive type?

$$A : \mathcal{U} \vdash \mathbf{data}\ \mathrm{id}(a_0 : A, a_1 : A)\ \mathbf{where}$$
$$\mid \mathrm{refl}(a : A) : \mathrm{id}(a, a)$$

Indexed inductive types

- What are the values of an indexed inductive type?

$$A : \mathcal{U} \vdash \textbf{data}\ \mathrm{id}(a_0 : A, a_1 : A)\ \textbf{where}$$
$$\mid \mathrm{refl}(a : A) : \mathrm{id}(a, a)$$

- Not only refl...

$$\frac{P \in \mathrm{Path}_A(M, N)\ [\Psi]}{\mathrm{coe}^{0 \rightsquigarrow 1}_{x.\mathrm{id}(A,M,P@x)}(\mathrm{refl}(M)) \in \mathrm{id}(A, M, N)\ [\Psi]}$$

Indexed inductive types

- What are the values of an indexed inductive type?

$$A : \mathcal{U} \vdash \textbf{data}\ \mathrm{id}(a_0 : A, a_1 : A)\ \textbf{where}$$
$$\mid \mathrm{refl}(a : A) : \mathrm{id}(a, a)$$

- Not only **refl**...

$$\frac{P \in \mathrm{Path}_A(M, N)\ [\Psi]}{\mathrm{coe}^{0 \rightsquigarrow 1}_{x.\mathrm{id}(A,M,P@x)}(\mathrm{refl}(M)) \in \mathrm{id}(A, M, N)\ [\Psi]}$$

- Add free coercion values for coercion between indices

Indexed inductive types

- What are the values of an indexed inductive type?

$$A : \mathcal{U} \vdash \textbf{data}\ \mathbf{id}(a_0 : A, a_1 : A)\ \textbf{where}$$
$$\mid \mathbf{refl}(a : A) : \mathbf{id}(a, a)$$

- Not only $\mathbf{refl}$...

$$\frac{P \in \mathrm{Path}_A(M, N)\ [\Psi]}{\mathrm{coe}^{0 \rightsquigarrow 1}_{x.\mathrm{id}(A,M,P@x)}(\mathrm{refl}(M)) \in \mathrm{id}(A, M, N)\ [\Psi]}$$

- Add free coercion values for coercion between indices

  → Coercion in parameters still reduces

Indexed inductive types

- What are the values of an indexed inductive type?

$$A : \mathcal{U} \vdash \textbf{data}\ \mathrm{id}(a_0 : A, a_1 : A)\ \textbf{where}$$
$$\mid \mathrm{refl}(a : A) : \mathrm{id}(a, a)$$

- Not only refl...

$$\frac{P \in \mathrm{Path}_A(M, N)\ [\Psi]}{\mathrm{coe}^{0 \rightsquigarrow 1}_{x.\mathrm{id}(A,M,P@x)}(\mathrm{refl}(M)) \in \mathrm{id}(A, M, N)\ [\Psi]}$$

- Add free coercion values for coercion between indices
  - → Coercion in parameters still reduces
  - → Size depends on size of indices, but not parameters

Cubical type theory and the future of HITs

Cubical type theory and the future of HITs

- Gracefully scaling to higher-d is essential for practice

Cubical type theory and the future of HITs

- Gracefully scaling to higher-d is essential for practice
  - → where does cubical type theory have models?

Cubical type theory and the future of HITs

- Gracefully scaling to higher-d is essential for practice
  - → where does cubical type theory have models?

- Going further
  - → inductive-inductive types (Hugunin 2019)
  - → implementation (`redtt`, Cubical Agda)
  - → more flexible schemata