

HoTTTEST

Displayed Type Theory

indexed modalities and analytic higher structures

Astra Kolomatskaia

jww

dTT: Michael Shulman

Kan in Narya: Kevin Carlson, Reed Mullanix

shapedTT: Reed Mullanix

shape categories for simplices

variations on simplex-themed shape categories

objects:

$\langle n \rangle$ for $n \in \mathbb{N}$ [*unaugmented*] or $n \in \mathbb{N} \cup \{-1\}$ [*augmented*]

regard $\langle n \rangle$ as the finite ordinal $[n + 1] \equiv \{0, \dots, n\}$

morphisms:

injective [*semi-simplicial*] or possibly not [*simplicial*]

order preserving [*no descriptor*] or possibly not [*symmetric*]

shape categories for simplices

properties:

whenever augmented: $\langle -1 \rangle$ is an initial object

whenever simplicial: $\langle 0 \rangle$ is a terminal object

the (augmented) semi-simplex category: inverse

the (augmented) symmetric semi-simplex category: inverse EI

the (augmented) simplex category: Reedy

the (augmented) symmetric simplex category: generalised Reedy

1

coherence in homotopy theory

semi-simplicial sets

we consider hset-valued presheaves on the semi-simplex category

fibred definition: a **semi-simplicial set** is a family of sets A_n for $n \geq 0$

along with maps $\partial_k : A_n \rightarrow A_{n-1}$, for $k \in \{0, \dots, n\}$

$$A_0 \begin{array}{c} \longleftarrow \\ \longleftarrow \\ \longleftarrow \end{array} A_1 \begin{array}{c} \longleftarrow \\ \longleftarrow \\ \longleftarrow \\ \longleftarrow \\ \longleftarrow \end{array} A_2 \begin{array}{c} \longleftarrow \\ \longleftarrow \\ \longleftarrow \\ \longleftarrow \\ \longleftarrow \\ \longleftarrow \\ \longleftarrow \end{array} A_3 \quad \dots$$

that satisfy

$$\partial_k \circ \partial_l = \partial_{l-1} \circ \partial_k \quad \text{for } k < l$$

semi-simplicial spaces

[in the setting of homotopy theory]

naïvely replacing sets with spaces, the identities become homotopies

$$\alpha_{k,l} : \partial_k \circ \partial_l \simeq \partial_{l-1} \circ \partial_k \quad \text{for } k < l$$

note: for $k < l < m$, there are two potentially distinct proofs that

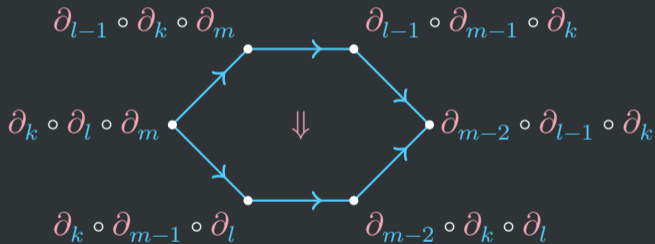
$$\partial_k \circ \partial_l \circ \partial_m \simeq \partial_{m-2} \circ \partial_{l-1} \circ \partial_k$$

coherence: higher homotopies $\beta_{k,l,m}$ with

$$\beta_{k,l,m} : \alpha_{k,l} \star \partial_m \cdot \partial_{l-1} \star \alpha_{k,m} \cdot \alpha_{l-1,m-1} \star \partial_k \simeq \partial_k \star \alpha_{l,m} \cdot \alpha_{k,m-1} \star \partial_l \cdot \partial_{m-2} \star \alpha_{k,l}$$

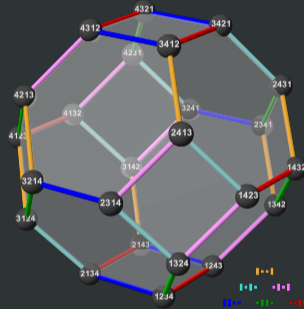
semi-simplicial spaces

picture for $\beta_{k,l,m}$ when $k < l < m$



semi-simplicial spaces

? compositions of four face maps ?



these give **permutahedral** coherences

writing these down is tricky; see [You Wouldn't Permutahedron \[Kol24\]](#)

semi-simplicial types

indexed '*moral definition*': an infinite record type with fields

$$A_0 : \text{Type}$$

$$A_1 : (x_{01} : A_0) (x_{10} : A_0) \rightarrow \text{Type}$$

$$A_2 : (x_{001} : A_0) (x_{010} : A_0) (\beta_{011} : A_1 x_{001} x_{010}) (x_{100} : A_0) (\beta_{101} : A_1 x_{001} x_{100}) \\ (\beta_{110} : A_1 x_{010} x_{100}) \rightarrow \text{Type}$$

$$A_3 : (x_{0001} : A_0) (x_{0010} : A_0) (\beta_{0011} : A_1 x_{0001} x_{0010}) (x_{0100} : A_0) (\beta_{0101} : A_1 x_{0001} x_{0100}) \\ (\beta_{0110} : A_1 x_{0010} x_{0100}) (\mathbf{f}_{0111} : A_2 x_{0001} x_{0010} \beta_{0011} x_{0100} \beta_{0101} \beta_{0110}) (x_{1000} : A_0) \\ (\beta_{1001} : A_1 x_{0001} x_{1000}) (\beta_{1010} : A_1 x_{0010} x_{1000}) (\mathbf{f}_{1011} : A_2 x_{0001} x_{0010} \beta_{0011} x_{1000} \beta_{1001} \beta_{1010}) \\ (\beta_{1100} : A_1 x_{0100} x_{1000}) (\mathbf{f}_{1101} : A_2 x_{0001} x_{0100} \beta_{0101} x_{1000} \beta_{1001} \beta_{1100}) \\ (\mathbf{f}_{1110} : A_2 x_{0010} x_{0100} \beta_{0110} x_{1000} \beta_{1010} \beta_{1100}) \rightarrow \text{Type}$$

...

semi-simplicial types

n -category cafe: Mike and I sketch a truncated equivalence

$$\tilde{A}_0 \equiv A_0$$

$$\tilde{A}_1(x, y) \equiv \sum_{(\alpha : A_1)} (\partial_1 \alpha = x) \times (\partial_0 \alpha = y)$$

$$\begin{aligned} \tilde{A}_2(x, y, z, (\alpha, p_0, q_0), (\beta, p_1, q_1), (\gamma, p_2, q_2)) &\equiv \\ \sum_{(f : A_2)} \sum_{(r_0 : \partial_1 \partial_2 f = x)} \sum_{(r_1 : \partial_0 \partial_2 f = y)} \sum_{(r_2 : \partial_0 \partial_1 f = z)} & \\ (\partial_2 f, r_0, r_1) = \tilde{A}_1(x, y) (\alpha, p_0, q_0) & \\ \times (\partial_1 f, \text{apeq } \alpha_{1,2} f \cdot r_0, r_2) = \tilde{A}_1(x, z) (\beta, p_1, q_1) & \\ \times (\partial_0 f, \text{apeq } \alpha_{0,2} f \cdot r_1, \text{apeq } \alpha_{0,1} f \cdot r_2) = \tilde{A}_1(y, z) (\gamma, p_2, q_2) & \end{aligned}$$

semi-simplicial types

moral: in the fibred formulation, coherences are necessary for describing homotopy types of n -simplices with given boundary [*a slice construction*]

in intensional type theory, constructing semi-simplicial types has been a notoriously hard problem

see Mike's 2014 n -category cafe post and my 2022 HoTTEST talk for an account of autophagy

the above account exhibits that the problem of constructing semi-simplicial types has homotopical content independent of syntactic strictness considerations

2

coherence in the categorical semantics of type theory

STLC

categorical semantics of STLC:

syntax interprets directly into any cartesian closed 1-category with given

$[\alpha]$ choices of a terminal object, products, and representing objects

$[\beta]$ objects interpreting base types

ergonomics:

adjust both the concepts of *syntactic presentation* and *model* to make instances of the former more automatically present instances of the latter

e.g. contextual CCCs [AHS95] / multicategories

objective metatheory: [Jon Sterling]

categorically, by viewing syntax solely as a freely generated model

dependent types

moving from simple to dependent Π -types shifts cartesian closed to locally cartesian closed *[Hof94]*

$$\frac{\Gamma \vdash A \text{ type} \quad \sigma : \Delta \rightarrow \Gamma}{\Delta \vdash A[\sigma] \text{ type}}$$

substitution corresponds to pullback

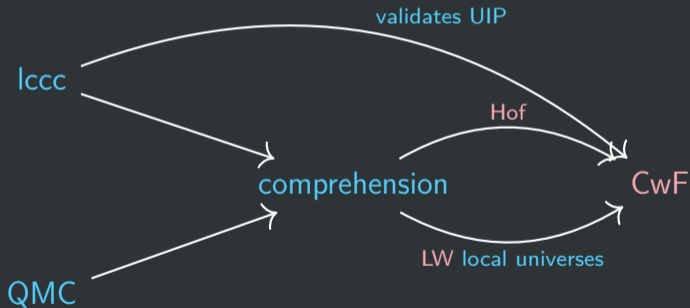
the requirement for a functorial choice of pullbacks leaves room for issues arising from a mismatch between equality and isomorphism

running notion of models: categories with families

how to model

? getting our hands on a (univalent!) model ?

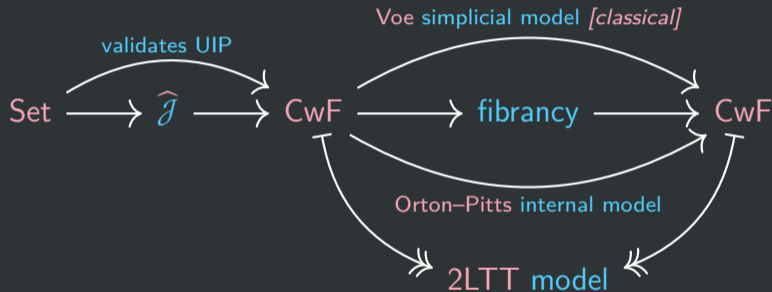
[α] from a categorical structure by strictifying



the input data is built from hsets; existing constructive methods do not handle the case of a univalent 1-category lacking an hset of objects

how to model

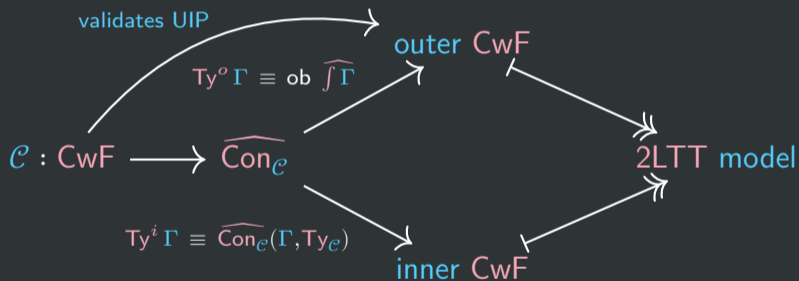
[β] start with presheaves, then impose fibrancy



these constructions are *ex nihilo* and \mathcal{J} can be an arbitrary index category for reasons that we will discuss, existing methods constrain extending this construction beyond cases starting from Set , even in cases where \mathcal{J} is Reedy

how to model

$[\gamma]$ for other 2LTT models, use Hofmann–Streicher universes



presheaves here are taken over the category of contexts; this is usually quite far away from being a prototypically *'nice'* index category

how to model

[δ] Mike has three papers on univalent diagram models:
inverse, inverse EI, and Reedy

$$\text{CwF} \xrightarrow{\text{Shu diagram model}} \text{CwF}$$

Mike's Reedy construction is scoped to diagrams valued in the univalent simplicial-set model; it does not address diagrams with arbitrary CwF values

upshot: for inverse or inverse EI index categories, Mike's constructions apply to diagrams with arbitrary CwF values *[EI means all endomorphisms are invertible]*

for the Sierpiński case, the paper does not deploy coherence theorems

these constructions hold promise for extending to homotopical settings in the absence of metatheoretic UIP

homotopy coherent models

Orton–Pitts framework constructions are carried out in an intensional type theory *with UIP*, thought of as a candidate for the internal language of a presheaf topos
the dream: *[NB quite far from currently doable]*

a fully homotopy coherent notion of model for dependent type theories inside intensional type theory without UIP

internal constructions of ∞ -syntax for univalent models, a non-hset notion of internal languages, initiality proofs, and objective metatheory

i.e. resolve sufficiently many coherence issues homotopically to enable moving past equality / isomorphism mismatch and strictness considerations

homotopy coherent models

progress:

[KS23] gives a homotopy coherent internal universal property of semi-simplicial types and builds a type theory around it – [Displayed Type Theory \[dTT\]](#)

the syntactic core of dTT interprets directly into any CwF; existence of externally-indexed sequential limits extends this core to include displayed coinductive types [e.g. SST] and \square -modal constructions; in elementary settings, we conjecture that sufficient conditions would resemble those for M-types

Mike's [Narya](#) implementation [developed with HOTT in mind] covers a non-modal fragment of dTT and appears to have no stuck terms

the syntax enables internal uniform access to higher structure; this talk includes a construction of Kan complexes in Narya

[shape theory](#) WIP

3

cubical and the Orton–Pitts framework

cube categories

cube category:

a monoidal category with a single generator $(\square, \otimes, \square^0, \square^1)$, or products thereof

\square^1 is often denoted I

cartesian when \square is monoidal cartesian

semicartesian when \square^0 is terminal

noncartesian otherwise

LOPS

have a shift operation $(- \otimes \square^1)^* : \mathbf{Set}^{\square^{\text{op}}} \rightarrow \mathbf{Set}^{\square^{\text{op}}}$

left and right adjoints are given by Day convolution and Day exponentiation

in cartesian settings $(- \otimes \square^1)^*$ is exponentiation with an interval, giving the path functor $\mathcal{P} \equiv (\mathbb{I} \rightarrow -)$ for $\mathbb{I} = \mathcal{J}(\square^1)$

[LOPS] uses crisp modal Π -types to construct fibrant universes

semicartesian settings still admit modal interval-style treatments, as conveyed by the spirit of the notation $(\mathbb{I} \multimap -)$ *[ND20]*

I am operating in the noncartesian setting

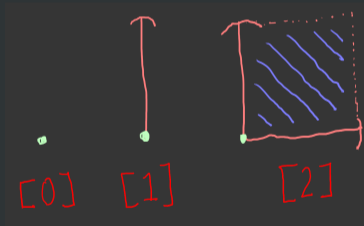


displayed type theory

unary semi-cubes

as index categories: the unary semi-cube category is isomorphic to the augmented semi-simplex category

its representable presheaves may be visualised as products of half-open intervals



we will use cube indexing for dimensions and start with 0 , instead of -1

unary semi-cubes

important:

this cube category has neither degeneracies nor symmetries

not a pathology of the arity; these exist in other unary cube categories

it is noncartesian *[no terminal object, corresponding to the lack of degeneracies]*

but it does have an initial object, which is not true of higher-arity cubes

thought: Mike and I understand the results in Evan Cavallo's thesis to make a strong case for the idea that internal parametricity requires symmetries in the presence of degeneracies

multi-modal structure

dTT is based on a multi-modal type theory [GKNB20] with **discrete** and **simplicial** modes



the following adjunctions hold in the mode theory

$$\diamond \dashv \triangle \dashv \square$$

$$\triangle \diamond \dashv \triangle \square$$

intended model

intended construction: start from an arbitrary CwF \mathcal{C} with Π -types and universes [KS23] gives an explicit construction of the CwF $\mathcal{C}^{\square^{\text{op}}}$ of \mathcal{C} -valued Reedy presheaves over the direct category \square ; this construction is written in an extensional type theory, and is feng shui'ed to have all subsequently relevant computation laws hold on the nose [no coherence theorems used]

absent other assumptions, these CwFs may be taken as the discrete and simplicial modes of a model into which, with the exception of displayed coinductive types [e.g. SST] and \square -modal formers, the syntax of dTT directly interprets

in the case that \mathcal{C} has ω -limits [record types of externally indexed infinite telescopes], this extends to allow interpretation of all dTT syntax; in elementary settings, we conjecture that it is possible to formulate sufficient conditions for the existence of displayed coinductive types that are analogous to those for M-types

intended model

$$[[dm]] \equiv \mathcal{C} \quad [[sm]] \equiv \mathcal{C}^{\square^{op}}$$

the *global* action of the modalities is given by

$\diamond : \mathcal{C}^{\square^{op}} \rightarrow \mathcal{C}$ is evaluation at \square^0 , the initial object
[NB this is not global sections]

$\triangle : \mathcal{C} \rightarrow \mathcal{C}^{\square^{op}}$ extends coskeletally
[by forming a constant diagram]

$\square : \mathcal{C}^{\square^{op}} \rightarrow \mathcal{C}$ takes the limit of the diagram

thrust of the construction

the syntax of dTT reifies, in the simplicial mode, additional structure that arises when working in a diagram model

certainly of interest is precomposition with the shift $(- \otimes \square^1)^* : \mathcal{C}^{\square^{\text{op}}} \rightarrow \mathcal{C}^{\square^{\text{op}}}$

its *global* action on simplicial mode contexts and substitutions is denoted $(-)^{\mathbb{D}}$

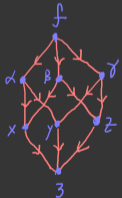
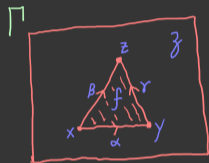
this is the *décalage* operation from simplicial homotopy theory

note: when compared with the Orton–Pitts framework treatment of

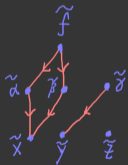
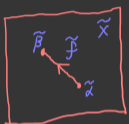
$$(- \otimes \square^1)^* : \mathbf{Set}^{\square^{\text{op}}} \rightarrow \mathbf{Set}^{\square^{\text{op}}}$$

our approach, at the very least, lets us generalise **Set** to an arbitrary CwF \mathcal{C}

décalage



\sqcap^D



display

have substitutions $\rho_{\Gamma} : \Gamma^{\mathbb{D}} \rightarrow \Gamma$

do not have substitutions $\Gamma \rightarrow \Gamma^{\mathbb{D}}$; this reflects the lack of degeneracies

attempting to localise our global '*path construction*' to a local action on fibres now presents difficulties not found in the cartesian or semicartesian cases

even restricted to a setting where the user-facing syntax applies only to closed terms, calling it a day after simply locking the operation under a crisp modality precludes the computational behaviour that we have come to expect of cubical theories; going under binders adds non-crisp variables to the context

display

the *global* action of $(- \otimes \square^1)^*$ on types in a context is given by **display**

$$\frac{\gamma : \Gamma \vdash_{\text{sm}} A \text{ } \gamma \text{ type}}{\gamma^+ : \Gamma^{\text{D}}, a : A^{\rho_{\Gamma}} \gamma^+ \vdash_{\text{sm}} A^{\text{d}} \gamma^+ a \text{ type}} \qquad \frac{\gamma : \Gamma \vdash_{\text{sm}} t \text{ } \gamma : A \text{ } \gamma}{\gamma^+ : \Gamma^{\text{D}} \vdash_{\text{sm}} t^{\text{d}} \text{ } \gamma^+ : A^{\text{d}} \gamma^+ t^{\rho_{\Gamma}}}$$

key relationship between décalage and display

$$\begin{aligned} (\gamma : \Gamma, a : A \text{ } \gamma)^{\text{D}} &\equiv (\gamma^+ : \Gamma^{\text{D}}, a : A^{\rho_{\Gamma}} \gamma^+, a' : A^{\text{d}} \gamma^+ a) \\ [\sigma, t]^{\text{D}} &\equiv [\sigma^{\text{D}}, t^{\rho_{\Delta}}, t^{\text{d}}] \end{aligned}$$

unary parametricity

display computes via unary parametricity

the following computation laws are strict in [KS23], when written Russell-style

$$\Gamma \vdash_{\text{sm}} A : \text{Type}$$

$$\Gamma^{\text{D}} \vdash_{\text{sm}} A^{\text{d}} : A^{\rho_{\Gamma}} \rightarrow \text{Type}$$

$$()_{\text{sm}}^{\text{D}} \equiv ()_{\text{sm}}$$

$$(\Gamma, x : A)^{\text{D}} \equiv (\Gamma^{\text{D}}, x : A^{\rho_{\Gamma}}, x' : A^{\text{d}} x)$$

$$x^{\text{d}} \equiv x'$$

$$(A \rightarrow B)^{\text{d}} f \equiv (x : A) \rightarrow A^{\text{d}} x \rightarrow B^{\text{d}} (f x)$$

$$\text{Type}^{\text{d}} A \equiv A \rightarrow \text{Type}$$

$$\Gamma \vdash_{\text{sm}} t : A$$

$$\Gamma^{\text{D}} \vdash_{\text{sm}} t^{\text{d}} : A^{\text{d}} t^{\rho_{\Gamma}}$$

$$[]_{\text{sm}}^{\text{D}} \equiv []_{\text{sm}}$$

$$[\sigma, t]^{\text{D}} \equiv [\sigma^{\text{D}}, t^{\rho_{\Delta}}, t^{\text{d}}]$$

$$(\lambda x. t)^{\text{d}} \equiv \lambda x. \lambda x'. t^{\text{d}}$$

$$(f a)^{\text{d}} \equiv f^{\text{d}} a a^{\text{d}}$$

iterating parametricity

iterating display probes the diagram levelwise

$A : \text{Type}$

$A^d : (\mathfrak{z}_0 : A) \rightarrow \text{Type}$

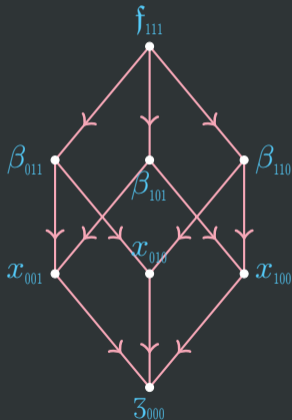
$A^{dd} : (\mathfrak{z}_{00} : A) (x_{01} : A^d \mathfrak{z}_{00}) (x_{10} : A^d \mathfrak{z}_{00}) \rightarrow \text{Type}$

$A^{ddd} : (\mathfrak{z}_{000} : A) (x_{001} : A^d \mathfrak{z}_{000}) (x_{010} : A^d \mathfrak{z}_{000})$

$(\beta_{011} : A^{dd} \mathfrak{z}_{000} x_{001} x_{010}) (x_{100} : A^d \mathfrak{z}_{000})$

$(\beta_{101} : A^{dd} \mathfrak{z}_{000} x_{001} x_{100}) (\beta_{110} : A^{dd} \mathfrak{z}_{000} x_{010} x_{100}) \rightarrow \text{Type}$

display is a backwards shift, semantically



semi-simplicial sets

Mike has an unpublished note, dated October 2012, that gives a universal characterisation of semi-simplicial *sets*

let \mathcal{S} denote the category of semi-simplicial sets

traditional slice construction: for $A : \mathcal{S}$ and $x : A_0$, define $A / x : \mathcal{S}$ by

$$(A / x)_n \equiv \{ f : A_{n+1} \mid (\partial_0 \circ \dots \circ \partial_0) f = x \}$$

face maps $\partial_k : (A / x)_n \rightarrow (A / x)_{n-1}$ derived from $\partial_k : A_{n+1} \rightarrow A_n$

projections $\rho_{A,x} : A / x \rightarrow A$ with $(\rho_{A,x})_n$ derived from $\partial_{n+1} : A_{n+1} \rightarrow A_n$

summary:

$$\left[A : \mathcal{S} \right] \rightarrow \left[A_0 : \mathbf{Set}, \{ A / x \}_{x : A_0} : \mathbf{Fam} \mathcal{S}, \{ \rho_{A,x} \}_{x : A_0} : \{ A / x \}_{x : A_0} \rightarrow \{ A \}_{* : * } \right]$$

semi-simplicial sets

a family coalgebra on $\mathcal{C} : \mathbf{Cat}$ is suitably functorial data of the form

$$\left[C : \mathcal{C} \right] \rightarrow \left[C_0 : \mathbf{Set} , \{ C/x \}_{x:C_0} : \mathbf{Fam} \mathcal{C} , \{ \rho_{C,x} \}_{x:C_0} : \{ C/x \}_{x:C_0} \rightarrow \{ C \}_{*:\star} \right]$$

note: the variable C appears in the codomain; this is really a copointed coalgebra

\mathcal{S} is universally characterised by the assignment $A \mapsto [A_0 , \{ A/x \} , \{ \rho_{A,x} \}]$

formally: equipped with the structure above, \mathcal{S} is a terminal object in the 2-category of categories equipped with family coalgebras

set indexed families are often studied in the semantics of dependent type theories

the perennial question: replace \mathbf{Set} with \mathbf{Type}

semi-simplicial types

the type `SST` of semi-simplicial types is constructed in the *simplicial mode*

coinductive characterisation: a semi-simplicial type $[A : SST]$ has zero-simplices $[Z A : Type]$ and slices $[S A : Z A \rightarrow SST^d A]$

the slices of A are semi-simplicial types *displayed* over A

`SST` is a displayed coinductive type

pseudo-Agda:

```
codata SST : Type where  
  Z : SST → Type  
  S : (A : SST) → Z A → SSTd A
```

Narya:

```
def SST : Type := codata  
[ A .z : Type  
| A .s : A .z → SSTd A ]
```

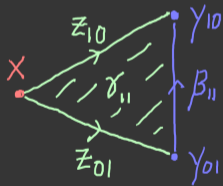
visualising the slice

n -simplices in $(A \cdot s x)$ live over n -simplices of A

$$z_1 : (A \cdot s x)_0 \rightarrow y_1$$



$$\gamma_{11} : (A \cdot s x)_1 \rightarrow y_{01} \times y_{10} \times z_{10} \times z_{01} \times \beta_{11}$$



extracting simplex types

```
def  $\sigma_0$  (A : SST) : Type :=  
  A .z
```

```
def  $\sigma^d_0$  (A : SST) (A' : SSTd A) (y1 :  $\sigma_0$  A) : Type :=  
  A' .z y1
```

```
def  $\sigma_1$  (A : SST) (x01 :  $\sigma_0$  A) (x10 :  $\sigma_0$  A) : Type :=  
   $\sigma^d_0$  A (A .s x01) x10 [≡ A .s x01 .z x10]
```

```
def  $\sigma^d_1$  (A : SST) (A' : SSTd A)  
  (y01 :  $\sigma_0$  A) (z01 :  $\sigma^d_0$  A A' y01) (y10 :  $\sigma_0$  A) (z10 :  $\sigma^d_0$  A A' y10)  
  (β11 :  $\sigma_1$  A y01 y10) : Type :=  
  A' .s y01 z01 .z y10 z10 β11
```

```
def  $\sigma_2$  (A : SST) (x001 :  $\sigma_0$  A) (x010 :  $\sigma_0$  A) (α011 :  $\sigma_1$  A x001 x010)  
  (x100 :  $\sigma_0$  A) (α101 :  $\sigma_1$  A x001 x100) (α110 :  $\sigma_1$  A x010 x100) : Type :=  
   $\sigma^d_1$  A (A .s x001) x010 α011 x100 α101 α110 [≡ A .s x001 .s x010 α011 .z x100 α101 α110]
```

5

Kan complexes in Narya

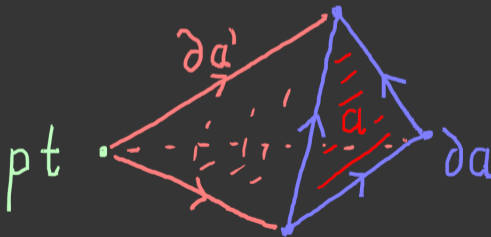
simplex boundaries and fillers [idea]

first mutually define simplex boundaries and fillers

```
def SST.○ (n : Δ ℕ) (A : SST) : Type
```

```
and SST.● (n : Δ ℕ) (A : SST) (○a : SST.○ n A) : Type
```

idea for inductive construction



simplex boundaries

for simplex boundaries

```
def SST.○ (n :  $\Delta$  ℕ) (A : SST) : Type :=  
match n  
[ zero.    $\mapsto$   $\top$   
| suc. n  $\mapsto$   
  sig  
  ( pt : A .z  
    ,  $\partial a$  : SST.○ n A  
    , a : SST.● n A  $\partial a$   
    ,  $\partial a'$  : SST.○d n A (A .s pt)  $\partial a$  ) ]
```

$SST.○^d : (n : \Delta \mathbb{N}) (A : SST) (A' : SST^d A) (\circ a : SST.○ n A) \rightarrow Type$

simplex fillers

for simplex fillers

```
and SST.● (n :  $\triangle$   $\mathbb{N}$ ) (A : SST) (○a : SST.○ n A) : Type :=  
match n  
[ zero.    $\mapsto$  A .z  
| suc. n  $\mapsto$  SST.●d n A (A .s (○a .pt)) (○a .∂a) (○a .∂a') (○a .a) ]
```

```
SST.●d : (n :  $\triangle$   $\mathbb{N}$ ) (A : SST) (A' : SSTd A) (○a : SST.○ n A)  
  (○a' : SST.○d n A A' ○a) (●a : SST.● n A ○a)  $\rightarrow$  Type
```

horns boundaries and fillers

next mutually define horn boundaries and fillers

```
def Horn.○ ( $n\ k : \triangle \mathbb{N}$ ) ( $A : \text{SST}$ ) ( $\circ a : \text{SST.}\circ\ n\ A$ ) : Type
and Horn.● ( $n\ k : \triangle \mathbb{N}$ ) ( $A : \text{SST}$ ) ( $\circ a : \text{SST.}\circ\ n\ A$ )
  ( $\bullet a : \text{SST.}\bullet\ n\ A\ \circ a$ ) ( $\Lambda : \text{Horn.}\circ\ n\ k\ A\ \circ a$ ) : Type
```

the construction is similarly combinatorial/analytic *[elided]*

Kan structures

putting these definitions together

```
def Horn (n k :  $\Delta$   $\mathbb{N}$ ) (A : SST) : Type :=  
sig  
  (  $\partial a$  : SST. $\circ$  n A  
    ,  $\Lambda a$  : Horn. $\circ$  n k A  $\partial a$ )
```

```
def Horn.data (n k :  $\Delta$   $\mathbb{N}$ ) (A : SST) ( $\Lambda$  : Horn n k A) : Type :=  
sig  
  ( face : SST. $\bullet$  n A ( $\Lambda$  . $\partial a$ )  
    , filler : Horn. $\bullet$  n k A ( $\Lambda$  . $\partial a$ ) face ( $\Lambda$  . $\Lambda a$ ))
```

```
def Kan (A : SST) : Type :=  
  (n k :  $\Delta$   $\mathbb{N}$ ) ( $\Lambda$  : Horn n k A)  $\rightarrow$  Horn.data n k A  $\Lambda$ 
```

demo

```
axiom A : SST
axiom x : A .z
axiom y : A .z
axiom  $\alpha$  : A .s x .z y
axiom z : A .z
axiom  $\beta$  : A .s x .z z
axiom  $\gamma$  : A .s y .z z
axiom f : A .s x .s y  $\alpha$  .z z  $\beta$   $\gamma$ 

def  $\Lambda\alpha\beta$  : Horn 1 0 A := ((y, (), z, ()), (x, ( $\alpha$ , ()),  $\beta$ , ()))
def  $\Lambda\alpha\gamma$  : Horn 1 1 A := ((x, (), z, ()), (y,  $\gamma$ ,  $\alpha$ ))
def  $\Lambda\beta\gamma$  : Horn 1 2 A := ((x, ()), (y, ()), (z,  $\gamma$ ,  $\beta$ ))

def  $\blacktriangle\alpha\beta$  : Horn.data 1 0 A  $\Lambda\alpha\beta$  := ( $\gamma$ , f)
def  $\blacktriangle\alpha\gamma$  : Horn.data 1 1 A  $\Lambda\alpha\gamma$  := ( $\beta$ , f)
def  $\blacktriangle\beta\gamma$  : Horn.data 1 2 A  $\Lambda\beta\gamma$  := ( $\alpha$ , f)
```



a theory of shapes / cofibrations

singular SSTs

the singular semi-simplicial types

```
def Sing (X : Type) : SST :=  
[ .z  ↦ X  
| .s  ↦ x ↦ Singd X (Path X x) ]
```

an internal statement that types are ∞ -groupoids

```
def Sing.Kan (X : Type) : Kan (Sing X) := ?
```

the ability to prove things like this is certainly desirable

displayed recursion

the usual trick is to make a displayed recursive call

```
Sing.Kand (X : Type) (X' : X → Type)
  : Kand (Sing X) (Singd X X') (Sing.Kan X)
```

the displayed recursive call gives a **displayed Kan structure** that solves filling problems of displayed horns *over prescribed downstairs fillers*

we would be done if we instead had a **relative Kan structure** that solved filling problems of displayed horns *over arbitrary downstairs fillers*

proposal: strengthen the inductive hypothesis with a Segal condition and ask for a contractible space of horn fillers

report: dTT constructions want feng shui'ed indices; fighting this is a bad idea

index control

curious situation:

we can define several higher categorical structures, but have difficulty working with them *[proofs or constructions]*

it seems that we need some more of that special juice that lets us hit coherences on the nose

shapes

in the user-facing shape DSL

$$N : \mathbb{D} \vdash \circ N \text{ shape}^N$$

$$\circ \text{zero.} \equiv \langle \rangle$$

$$\circ (\text{suc. } N) \equiv \langle \partial a : \mathbb{0} (\circ N) , a : \mathbb{0} (\bullet N) \partial a , \partial a' : \mathbb{1} (\circ N) \partial a \rangle$$

$$N : \mathbb{D} \mid \circ a : \circ N \vdash \bullet N \circ a \text{ point}^N$$

$$\circ \text{zero. } \circ a \equiv \varepsilon$$

$$\circ (\text{suc. } N) \circ a \equiv \mathbb{1} (\bullet N) (\circ a . \partial a) (\circ a . \partial a') (\circ a . a)$$

proof-relevant cofibrations

shapedTT will not have an *extensional* face lattice that checks for overlaps when forming compound shapes

shapes will instead be endowed with *intensional* computational behaviour, serving as a notion of *proof-relevant cofibration*

$$A^{[\circ \text{ (suc. } N)]} \equiv \left(\partial a : (A^{\rho_{\Gamma}})^{[\circ \ N]} , a : (A^{\rho_{\Gamma}})^{[\bullet \ N]} \partial a , \partial a' : (A^{\mathbf{d}})^{[\circ \ N]} \partial a \right)$$

this mirrors the formula defining matching objects in the simplicial model construction of *[KS23]*

big caveat to the above: we really need to avoid serialising into a linear dependency structure, and instead introduce a more general notion of telescope annotated by shape data; this encodes a more general dependency structure and plays a load-bearing role in computation dispatch *[higher dimensional syntax]*

relative Kan structures

define the notion of **relative Kan structure** of any degree

```
def Kan (N :  $\mathbb{D}$ ) ( $\partial A$  : SST [ $\circ$  N]) (A : SST [ $\bullet$  N]  $\partial A$ ) : Type :=  
  (n k :  $\triangleleft$   $\mathbb{N}$ ) ( $\partial \Lambda$  : Horn [ $\circ$  N] n k  $\partial A$ ) ( $\Lambda$  : Horn [ $\bullet$  N] n k  $\partial A$  A  $\partial \Lambda$ )  
  ( $\partial \blacktriangle$  : Horn.data [ $\circ$  N] n k  $\partial A$   $\partial \Lambda$ )  
   $\rightarrow$  Horn.data [ $\bullet$  N] n k  $\partial A$  A  $\partial \Lambda$   $\Lambda$   $\partial \blacktriangle$ 
```

strengthen the theorem

```
def Sing.Kan (N :  $\mathbb{D}$ ) ( $\partial X$  : Type [ $\circ$  N]) (X : Type [ $\bullet$  N]  $\partial X$ )  
  : Kan N (Sing [ $\circ$  N]  $\partial X$ ) (Sing [ $\bullet$  N]  $\partial X$  X) := ?
```

then: do not make a displayed recursive call, and instead make a recursive call that **raises the degree of relativity** after assembling the data on hand into a higher dimensional diagram; this gives full control over the otherwise prescribed index

simplicial homotopy theory

conceptual picture:

in simplicial homotopy theory, one is often required to generalise theorems stated in the absolute case to the relative case

[Emily Riehl](#) told me that this is because the slice construction morally wants to live at the level of morphisms and not at the level of objects

the traditional slice construction preserves the degree of relativity; it suffices to move a single step from the absolute case to the relative case

the displayed slice construction raises the degree of relativity; it becomes necessary to generalise and work across all degrees of relative cases at once

conclusion

Astra's vision for the future of dTT:

furnish tools for solving the coherences on the nose by way of indexing

break from the cubical style of performing boundary adjustments

this teases out the analytic patterns of higher categorical constructions

the gadgets built compute well

no transport hell – only index feng shui

thank you for listening to my talk

