Weak ω-categories as models of a type theory

Thibaut Benjamin

CEA LIST

Logic and higher structures
CIRM, February 23, 2022

Dependent type theories and higher structures

HoTT and weak ω-groupoids

- In HoTT every type has associated tower of identity type

HoTT and weak ω-groupoids

- In HoTT every type has associated tower of identity type
- The iterated identity types have the structure of an ω-groupoid

HoTT and weak ω-groupoids

- In HoTT every type has associated tower of identity type

- The iterated identity types have the structure of an ω-groupoid

- This suggest a link between dependent type theories and higher structures.

Brunerie's type theory

- A minimal dependent type theory to describe weak ω-groupoids, introduced by Brunerie [2]

Brunerie's type theory

- A minimal dependent type theory to describe weak ω-groupoids, introduced by Brunerie [2]

- Types : encode the higher dimensional disks (same dependency as in HoTT)

$$\frac{\Gamma \vdash}{\Gamma \vdash \star} \qquad \frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : A}{\Gamma \vdash t =_A u}$$

Brunerie's type theory

- A minimal dependent type theory to describe weak ω-groupoids, introduced by Brunerie [2]

- Types : encode the higher dimensional disks (same dependency as in HoTT)

$$\frac{\Gamma \vdash}{\Gamma \vdash \star} \qquad \frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : A}{\Gamma \vdash t =_A u}$$

In this type theory, the identity types are not inductive, instead there is a family of term constructors that witnesses the algebraic structure.

DTT and higher structures

The correspondence between dependent type theories and higher algebraic structure follows the principle

$$\text{type dependency} \rightsquigarrow \text{higher dimensional shapes}$$
$$\text{term constructors} \rightsquigarrow \text{algebraic structure}$$

Weak ω-categories

An Overview

Weak ω-categories are higher structure with directed arrows in all levels

$\omega \neq \infty$, Lurie uses "∞-categories" for $(\infty, 1) - categories$.

Weak ω-categories are higher structure with directed arrows in all levels

$\omega \neq \infty$, Lurie uses "∞-categories" for $(\infty, 1) - categories$.

Several definitions, among which some globular ones are

- Batanin-Leinster [6] :Algebras for the initial globular operad with contractions.

An Overview

Weak ω-categories are higher structure with directed arrows in all levels

$\omega \neq \infty$, Lurie uses "∞-categories" for $(\infty, 1) - categories$.

Several definitions, among which some globular ones are

- Batanin-Leinster [6] :Algebras for the initial globular operad with contractions.
- Grothendieck-Maltsiniotis (G.-M.) [7] :Presheaves over the category Θ_∞ which preserve globular sums.

An Overview

Weak ω-categories are higher structure with directed arrows in all levels

$\omega \neq \infty$, Lurie uses "∞-categories" for $(\infty, 1) - categories$.

Several definitions, among which some globular ones are

- Batanin-Leinster [6] :Algebras for the initial globular operad with contractions.
- Grothendieck-Maltsiniotis (G.-M.) [7] :Presheaves over the category Θ_∞ which preserve globular sums.

Lots of other definitions, with other shapes.

Weak ω-categories

The Grothendieck-Maltsiniotis definition

A bit of context

- Originally proposed by Grothendieck for weak ω-groupoids [5].
  Brunerie has proved that his type theory describes exactly Grothendieck's weak ω-groupoids

A bit of context

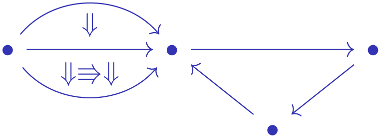
- Originally proposed by Grothendieck for weak ω-groupoids [5].
  Brunerie has proved that his type theory describes exactly Grothendieck's weak ω-groupoids
- Extended by Maltsiniotis to weak ω-categories [7]
  Intuition : enforce a privileged direction on the rules

A bit of context

- Originally proposed by Grothendieck for weak ω-groupoids [5].
  Brunerie has proved that his type theory describes exactly Grothendieck's weak ω-groupoids
- Extended by Maltsiniotis to weak ω-categories [7]
  Intuition : enforce a privileged direction on the rules
- Proven equivalent to Batanin-Leinster definition by Ara [1]

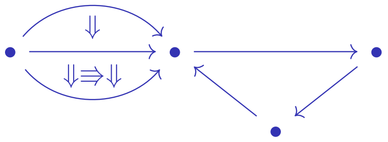
A globular definition

- Supported by globular sets



A globular definition

- Supported by globular sets



- Presheaf category whose representables are disks

Pasting schemes

Pasting schemes are globular sets that represent a "unique" composition in ω-categories.

Pasting schemes

Pasting schemes are globular sets that represent a "unique" composition in ω-categories.

- They are well ordered and without holes

  Examples : $\bullet \longrightarrow \bullet \longrightarrow \bullet$

Pasting schemes

Pasting schemes are globular sets that represent a "unique" composition in ω-categories.

- They are well ordered and without holes

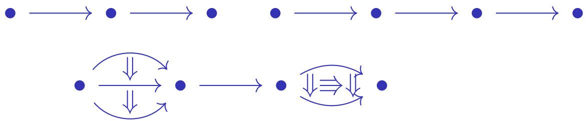
  Examples : $\bullet \longrightarrow \bullet \longrightarrow \bullet \qquad \bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet$

Pasting schemes

Pasting schemes are globular sets that represent a "unique" composition in ω-categories.

- They are well ordered and without holes

Examples :

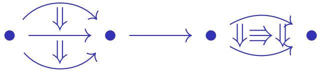


Pasting schemes

Pasting schemes are globular sets that represent a "unique" composition in ω-categories.

- They are well ordered and without holes

Examples : $\bullet \longrightarrow \bullet \longrightarrow \bullet \qquad \bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet$



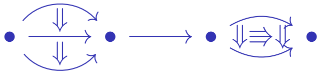
Counter-examples : $\bullet \longrightarrow \bullet \longleftarrow \bullet$

Pasting schemes

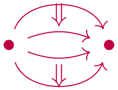
Pasting schemes are globular sets that represent a "unique" composition in ω-categories.

- They are well ordered and without holes

Examples : $\bullet \longrightarrow \bullet \longrightarrow \bullet \qquad \bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet$



Counter-examples : $\bullet \longrightarrow \bullet \longleftarrow \bullet$

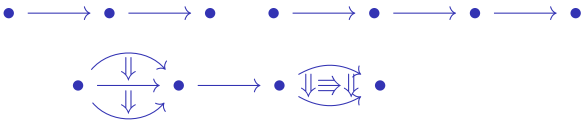


Pasting schemes

Pasting schemes are globular sets that represent a "unique" composition in ω-categories.

- They are well ordered and without holes

Examples :



Counter-examples :

Globular sums

- Globular sums formalize the idea of well-ordered and without holes

Globular sums

- Globular sums formalize the idea of well-ordered and without holes

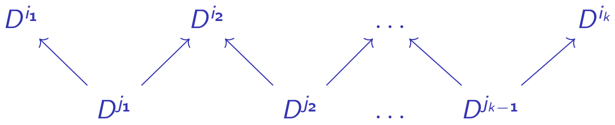
- They are defined as colimits of disks, for diagrams of the form

$$
\begin{array}{ccccccc}
D^{i_1} & & D^{i_2} & & \dots & & D^{i_k} \\
& \nwarrow \quad \nearrow & & \nwarrow \quad \nearrow & & \nwarrow \quad \nearrow & \\
& D^{j_1} & & D^{j_2} & \dots & D^{j_{k-1}} &
\end{array}
$$

Globular sums

- Globular sums formalize the idea of well-ordered and without holes

- They are defined as colimits of disks, for diagrams of the form



- The globular sums are exactly the pasting schemes.
  Define Θ_0 to the full subcategory of globular sets whose objects are the globular sums.

Adding operations/coherences

- An operation in a globular set Γ is materialized by a morphism $\Gamma \to D^k$.

Adding operations/coherences

- An operation in a globular set Γ is materialized by a morphism $\Gamma \to D^k$.

- Build the category Θ_∞ by freely adding arrows to Θ_0, according to the 2 following principles :

Adding operations/coherences

- An operation in a globular set Γ is materialized by a morphism $\Gamma \to D^k$.

- Build the category Θ_∞ by freely adding arrows to Θ_0, according to the 2 following principles :
  - **Every pasting scheme has a composition**

Adding operations/coherences

- An operation in a globular set Γ is materialized by a morphism $\Gamma \to D^k$.

- Build the category Θ_∞ by freely adding arrows to Θ_0, according to the 2 following principles :
  - **Every pasting scheme has a composition**
  - **Any two ways of composing a pasting scheme are connected by a higher cell**

Adding operations/coherences

- An operation in a globular set Γ is materialized by a morphism $\Gamma \to D^k$.

- Build the category Θ_∞ by freely adding arrows to Θ_0, according to the 2 following principles :
  - **Every pasting scheme has a composition**
  - **Any two ways of composing a pasting scheme are connected by a higher cell**

- This should remind you of "contractibility as uniqueness"

Adding operations/coherences

- An operation in a globular set Γ is materialized by a morphism $\Gamma \to D^k$.

- Build the category Θ_∞ by freely adding arrows to Θ_0, according to the 2 following principles :
  - **Every pasting scheme has a composition**
  - **Any two ways of composing a pasting scheme are connected by a higher cell**

- This should remind you of "contractibility as uniqueness"

actually one has to do inifitely many steps to build Θ_∞

Definition of weak ω-categories

Weak ω categories are globular sets which have all the compositions and coherences described in Θ_∞.

Definition of weak ω-categories

Weak ω categories are globular sets which have all the compositions and coherences described in Θ_∞.

- We can define them as presheaves over the category Θ_∞.

Definition of weak ω-categories

Weak ω categories are globular sets which have all the compositions and coherences described in Θ_∞.

- We can define them as presheaves over the category Θ_∞.

- We need to require those presheaves to preserve globular sums, to avoid having too much shapes allowed.

The type theory CaTT

Intuition

- Introduced by Finster and Mimram [4]
  Intuition : It defines the following "pushout"

$$\begin{array}{ccc}
\text{Grothendieck's } \omega\text{-groupoids} & \xrightarrow{\text{direction}} & \text{G.-M. } \omega\text{-categories} \\
\text{type theory} \downarrow & & \downarrow \text{type theory} \\
\text{Brunerie's type theory} & \xrightarrow[\text{direction}]{} & \text{CaTT}
\end{array}$$

The type theory CaTT

Dependent type theories and their categorical semantics

Building blocks of a dependent type theory (DTT)

A dependent type theory $\mathcal{T}$ has syntactic objects :

- Contexts $\Gamma, \Delta, \ldots$: lists of pairs of variables and types

$$\Gamma \vdash$$

Building blocks of a dependent type theory (DTT)

A dependent type theory $\mathcal{T}$ has syntactic objects :

- Contexts $\Gamma, \Delta, \ldots$: lists of pairs of variables and types

$$\Gamma \vdash$$

- Types $A, B, \ldots$: constructed with type constructors

$$\Gamma \vdash A$$

Building blocks of a dependent type theory (DTT)

A dependent type theory $\mathcal{T}$ has syntactic objects :

- Contexts $\Gamma, \Delta, \dots$: lists of pairs of variables and types

$$\Gamma \vdash$$

- Types $A, B, \dots$: constructed with type constructors

$$\Gamma \vdash A$$

- Terms : $t, u, \dots$: constructed with term constructors

$$\Gamma \vdash t : A$$

Building blocks of a dependent type theory (DTT)

A dependent type theory $\mathcal{T}$ has syntactic objects :

- Contexts $\Gamma, \Delta, \ldots$: lists of pairs of variables and types

$$\Gamma \vdash$$

- Types $A, B, \ldots$: constructed with type constructors

$$\Gamma \vdash A$$

- Terms : $t, u, \ldots$: constructed with term constructors

$$\Gamma \vdash t : A$$

- Substitutions $\gamma, \delta, \ldots$: lists of pairs of variables and terms

$$\Delta \vdash \gamma : \Gamma$$

Structure of a dependent type theories

The dependent type theories the structure in common

- A variable is a valid term in a context :

$$\frac{\Gamma \vdash \qquad (x, A) \in \Gamma}{\Gamma \vdash x : A}$$

Structure of a dependent type theories

The dependent type theories the structure in common

- A variable is a valid term in a context :

$$\frac{\Gamma \vdash \qquad (x, A) \in \Gamma}{\Gamma \vdash x : A}$$

- The substitutions act on terms and types

$$\frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A}{\Delta \vdash A[\gamma]} \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash t : A}{\Delta \vdash t[\gamma] : A[\gamma]}$$

Structure of a dependent type theories

The dependent type theories the structure in common

- A variable is a valid term in a context :

$$\frac{\Gamma \vdash \qquad (x, A) \in \Gamma}{\Gamma \vdash x : A}$$

- The substitutions act on terms and types

$$\frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A}{\Delta \vdash A[\gamma]} \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash t : A}{\Delta \vdash t[\gamma] : A[\gamma]}$$

- Contexts can be extended with types

$$\frac{\Gamma \vdash \qquad \Gamma \vdash A \qquad x \notin \Gamma}{\Gamma, x : A \vdash}$$

Structure of a dependent type theories

The dependent type theories the structure in common

- A variable is a valid term in a context :

$$\frac{\Gamma \vdash \qquad (x, A) \in \Gamma}{\Gamma \vdash x : A}$$

- The substitutions act on terms and types

$$\frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A}{\Delta \vdash A[\gamma]} \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash t : A}{\Delta \vdash t[\gamma] : A[\gamma]}$$

- Contexts can be extended with types

$$\frac{\Gamma \vdash \qquad \Gamma \vdash A \qquad x \notin \Gamma}{\Gamma, x : A \vdash}$$

- Substitutions can be extended with terms

$$\frac{\Delta \vdash \gamma : \Gamma \qquad \Gamma \vdash A \qquad \Delta \vdash t : A[\gamma] \qquad x \notin \Gamma}{\Delta \vdash \langle \gamma, x \mapsto t \rangle : (\Gamma, x : A)}$$

Categories with families (CwF)

Categories with families are a categorical formulation of the structure of DTT

- Introduced by Dybjer [3]

Categories with families (CwF)

Categories with families are a categorical formulation of the structure of DTT

- Introduced by Dybjer [3]
- Build a category : objects=context, morphisms=substitutions
  We call it the syntactic category of $\mathcal{T}$ and denote it $\mathcal{S}_{\mathcal{T}}$

Categories with families (CwF)

Categories with families are a categorical formulation of the structure of DTT

- Introduced by Dybjer [3]
- Build a category : objects=context, morphisms=substitutions
  We call it the syntactic category of $\mathcal{T}$ and denote it $\mathcal{S}_{\mathcal{T}}$
- It has a terminal element
  The empty context

Categories with families (CwF)

Categories with families are a categorical formulation of the structure of DTT

- Introduced by Dybjer [3]
- Build a category : objects=context, morphisms=substitutions
  We call it the syntactic category of $\mathcal{T}$ and denote it $\mathcal{S}_{\mathcal{T}}$
- It has a terminal element
  The empty context
- Define $\mathsf{Ty}_\Gamma = \{\text{types in } \Gamma\}$, $\mathsf{Tm}_\Gamma^A = \{\text{terms of type } A \text{ in } \Gamma\}$
  Ty is a presheaf over $\mathcal{S}_{\mathcal{T}}$, Tm is a presheaf over El(Ty)

Categories with families (CwF)

Categories with families are a categorical formulation of the structure of DTT

- Introduced by Dybjer [3]
- Build a category : objects=context, morphisms=substitutions
  We call it the syntactic category of $\mathcal{T}$ and denote it $\mathcal{S}_{\mathcal{T}}$
- It has a terminal element
  The empty context
- Define $\mathsf{Ty}_\Gamma = \{\text{types in } \Gamma\}$, $\mathsf{Tm}_\Gamma^A = \{\text{terms of type } A \text{ in } \Gamma\}$
  Ty is a presheaf over $\mathcal{S}_{\mathcal{T}}$, Tm is a presheaf over El(Ty)
- A context extension : given a context Γ and a type $\Gamma \vdash A$, an object of $\mathcal{S}_{\mathcal{T}}$
  Defines a functor $\mathsf{El}(\mathsf{Ty}) \to \mathcal{S}_{\mathcal{T}}$ characterized by a universal property

Categories with families (CwF)

Categories with families are a categorical formulation of the structure of DTT

- Introduced by Dybjer [3]
- Build a category : objects=context, morphisms=substitutions
  We call it the syntactic category of $\mathcal{T}$ and denote it $\mathcal{S}_{\mathcal{T}}$
- It has a terminal element
  The empty context
- Define $\mathsf{Ty}_\Gamma = \{\text{types in } \Gamma\}$, $\mathsf{Tm}_\Gamma^A = \{\text{terms of type } A \text{ in } \Gamma\}$
  Ty is a presheaf over $\mathcal{S}_{\mathcal{T}}$, Tm is a presheaf over El(Ty)
- A context extension : given a context Γ and a type $\Gamma \vdash A$, an object of $\mathcal{S}_{\mathcal{T}}$
  Defines a functor $\mathsf{El}(\mathsf{Ty}) \to \mathcal{S}_{\mathcal{T}}$ characterized by a universal property

A CwF is the collection of all this data

This presentation follows the style of Awodey's natural models

Categorical semantics

The models are a way to incarnate the axioms defining a dependent type theory in sets

Categorical semantics

The models are a way to incarnate the axioms defining a dependent type theory in sets

- There is a CwF structure on the category of sets

Categorical semantics

The models are a way to incarnate the axioms defining a dependent type theory in sets

- There is a CwF structure on the category of sets
- A model of the theory $\mathcal{T}$ is a morphism of CwF $\mathcal{S}_{\mathcal{T}} \to \mathrm{Set}$

The type theory CaTT

Presentation of the theory

The theory GSeTT

- Start with describing the type dependancies : higher dimensional shapes

The theory GSeTT

- Start with describing the type dependancies : higher dimensional shapes

- Same as Brunerie's Type Theory and Identity types in HoTT

$$\frac{\Gamma \vdash}{\Gamma \vdash \star} \qquad \frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : A}{\Gamma \vdash t \underset{A}{\longrightarrow} u}$$

change the name to emphasize directionality

The theory GSeTT

- Start with describing the type dependancies : higher dimensional shapes

- Same as Brunerie's Type Theory and Identity types in HoTT

$$\frac{\Gamma \vdash}{\Gamma \vdash \star} \qquad \frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : A}{\Gamma \vdash t \underset{A}{\longrightarrow} u}$$

change the name to emphasize directionality

- Denote GSeTT the theory with just these type constructors

The theory GSeTT

- Start with describing the type dependancies : higher dimensional shapes

- Same as Brunerie's Type Theory and Identity types in HoTT

$$\frac{\Gamma \vdash}{\Gamma \vdash \star} \qquad\qquad \frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : A}{\Gamma \vdash t \underset{A}{\longrightarrow} u}$$

  change the name to emphasize directionality

- Denote GSeTT the theory with just these type constructors

- $\mathcal{S}_{\mathsf{GSeTT}}$ is the opposite of finite globular sets
  For instance, the following context and globular sets are in correspondence

  `(x:*, y:*, z:*, f:x->y, g: y->z)` $\qquad \overset{x}{\bullet} \longrightarrow \overset{y}{\bullet} \longleftarrow \overset{z}{\bullet}$

Ps-contexts

ps-contexts are context that represent pasting schemes

Ps-contexts

ps-contexts are context that represent pasting schemes

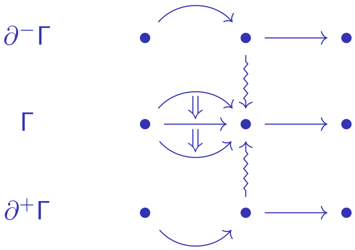
- We introduce the judgment $\Gamma \vdash_{\mathsf{ps}}$ to recognize them
  To simplify the recognition, we require the ps-context to be in a specific order

Ps-contexts

ps-contexts are context that represent pasting schemes

- We introduce the judgment $\Gamma \vdash_{ps}$ to recognize them
  To simplify the recognition, we require the ps-context to be in a specific order

- Each ps-context Γ has a source $\partial^{-}\Gamma$ and a target $\partial^{+}\Gamma$



The theory CaTT

To the theory CaTT, add term constructors corresponding to the two principle expressing that the "space" of composition of each pasting scheme is "contractible".

The theory CaTT

To the theory CaTT, add term constructors corresponding to the two principle expressing that the "space" of composition of each pasting scheme is "contractible".

- **Each pasting has a composition**

$$\frac{\Gamma \vdash_{\mathsf{ps}} \qquad \partial^- \Gamma \vdash t : A \qquad \partial^+ \Gamma \vdash u : A}{\Gamma \vdash \mathsf{op}_{\Gamma, t \underset{A}{\longrightarrow} u} : t \underset{A}{\longrightarrow} u}$$

$$\begin{aligned} \mathrm{Var}(t : A) &= \mathrm{Var}(\partial^-(\Gamma)) \\ \mathrm{Var}(u : A) &= \mathrm{Var}(\partial^+(\Gamma)) \end{aligned}$$

The theory CaTT

To the theory CaTT, add term constructors corresponding to the two principle expressing that the "space" of composition of each pasting scheme is "contractible".

- **Each pasting has a composition**

$$\frac{\Gamma \vdash_{\mathsf{ps}} \qquad \partial^- \Gamma \vdash t : A \qquad \partial^+ \Gamma \vdash u : A}{\Gamma \vdash \mathsf{op}_{\Gamma, t \underset{A}{\longrightarrow} u} : t \underset{A}{\longrightarrow} u} \qquad \begin{array}{ll} \mathrm{Var}(t : A) & = \mathrm{Var}(\partial^-(\Gamma)) \\ \mathrm{Var}(u : A) & = \mathrm{Var}(\partial^+(\Gamma)) \end{array}$$

- **Every two compositions of the same pasting scheme are related**

$$\frac{\Gamma \vdash_{\mathsf{ps}} \qquad \Gamma \vdash t : A \qquad \Gamma \vdash u : A}{\Gamma \vdash \mathsf{coh}_{\Gamma, t \underset{A}{\longrightarrow} u} : t \underset{A}{\longrightarrow} u} \qquad \begin{array}{ll} \mathrm{Var}(t : A) & = \mathrm{Var}(\Gamma) \\ \mathrm{Var}(u : A) & = \mathrm{Var}(\Gamma) \end{array}$$

Applying operations and coherences

- We have explained how to compute terms in ps-contexts

Applying operations and coherences

- We have explained how to compute terms in ps-contexts

- We get terms in generic context by action of substitutions
  Hence relax the previous rules to have

$$\frac{\Gamma \vdash_{\mathsf{ps}} \qquad \partial^{-}\Gamma \vdash t : A \qquad \partial^{+}\Gamma \vdash u : A \qquad \Delta \vdash \gamma : \Gamma}{\Delta \vdash \mathsf{op}_{\Gamma, t \underset{A}{\to} u}[\gamma] : t[\gamma] \to u[\gamma]}$$

$$\frac{\Gamma \vdash_{\mathsf{ps}} \qquad \Gamma \vdash t : A \qquad \Gamma \vdash u : A \qquad \Delta \vdash \gamma : \Gamma}{\Delta \vdash \mathsf{coh}_{\Gamma, t \underset{A}{\to} u}[\gamma] : t[\gamma] \to u[\gamma]}$$

(keeping the side condition)

Examples of derivation

- Composition :
  Consider the ps-context
  $\Gamma_c = (x : \star, y : \star, f : x \to y, z : \star, g : y \to z)$.

Examples of derivation

- Composition :
  Consider the ps-context
  $\Gamma_c = (x : \star, y : \star, f : x \to y, z : \star, g : y \to z)$.
  We have $\partial^- \Gamma_c = (x : \star)$ and $\partial^+ \Gamma_c = (z : \star)$.

Examples of derivation

- Composition :
  Consider the ps-context
  $\Gamma_c = (x : \star, y : \star, f : x \to y, z : \star, g : y \to z)$.
  We have $\partial^- \Gamma_c = (x : \star)$ and $\partial^+ \Gamma_c = (z : \star)$.
  So we deduce the term $\Gamma_c \vdash \mathrm{op}_{\Gamma_c, x \to z} : x \to z$ (denoted $\mathtt{comp}$).

Examples of derivation

- Composition :
  Consider the ps-context
  $\Gamma_c = (x : \star, y : \star, f : x \to y, z : \star, g : y \to z)$.
  We have $\partial^- \Gamma_c = (x : \star)$ and $\partial^+ \Gamma_c = (z : \star)$.
  So we deduce the term $\Gamma_c \vdash \mathrm{op}_{\Gamma_c, x \to z} : x \to z$ (denoted $\mathtt{comp}$).

- Associativity :
  Consider the ps-context
  $\Gamma_a = (x : \star, y : \star, f : x \to y, z : \star, g : y \to z, w : \star, h : z \to w)$.

Examples of derivation

- Composition :
  Consider the ps-context
  $\Gamma_c = (x : \star, y : \star, f : x \to y, z : \star, g : y \to z)$.
  We have $\partial^- \Gamma_c = (x : \star)$ and $\partial^+ \Gamma_c = (z : \star)$.
  So we deduce the term $\Gamma_c \vdash \mathrm{op}_{\Gamma_c, x \to z} : x \to z$ (denoted $\mathtt{comp}$).

- Associativity :
  Consider the ps-context
  $\Gamma_a = (x : \star, y : \star, f : x \to y, z : \star, g : y \to z, w : \star, h : z \to w)$.
  We have the type
  $\Gamma_a \vdash \mathtt{comp}(f, \mathtt{comp}(g, h)) \to \mathtt{comp}(\mathtt{comp}(f, g), h)$.

Examples of derivation

- Composition :
  Consider the ps-context
  $\Gamma_c = (x : \star, y : \star, f : x \to y, z : \star, g : y \to z)$.
  We have $\partial^- \Gamma_c = (x : \star)$ and $\partial^+ \Gamma_c = (z : \star)$.
  So we deduce the term $\Gamma_c \vdash \mathsf{op}_{\Gamma_c, x \to z} : x \to z$ (denoted $\mathtt{comp}$).

- Associativity :
  Consider the ps-context
  $\Gamma_a = (x : \star, y : \star, f : x \to y, z : \star, g : y \to z, w : \star, h : z \to w)$.
  We have the type
  $\Gamma_a \vdash \mathtt{comp}(f, \mathtt{comp}(g, h)) \to \mathtt{comp}(\mathtt{comp}(f, g), h)$.
  Both sides use all the variables of Γ_a (some are implicit).

Examples of derivation

- Composition :
  Consider the ps-context
  $\Gamma_c = (x : \star, y : \star, f : x \to y, z : \star, g : y \to z)$.
  We have $\partial^- \Gamma_c = (x : \star)$ and $\partial^+ \Gamma_c = (z : \star)$.
  So we deduce the term $\Gamma_c \vdash \mathrm{op}_{\Gamma_c, x \to z} : x \to z$ (denoted $\mathtt{comp}$).

- Associativity :
  Consider the ps-context
  $\Gamma_a = (x : \star, y : \star, f : x \to y, z : \star, g : y \to z, w : \star, h : z \to w)$.
  We have the type
  $\Gamma_a \vdash \mathtt{comp}(f, \mathtt{comp}(g, h)) \to \mathtt{comp}(\mathtt{comp}(f, g), h)$.
  Both sides use all the variables of Γ_a (some are implicit).
  So we deduce the term $\Gamma_c \vdash \mathrm{op}_{\Gamma_c, x \to z} : x \to z$ (denoted $\mathtt{comp}$).

The type theory CaTT

Semantics of the theory

The subcategory of ps-contexts

Define PS : the full subcategory of $\mathcal{S}_{\mathsf{CaTT}}$ whose objects are the ps-contexts

The subcategory of ps-contexts

Define PS : the full subcategory of $\mathcal{S}_{\mathsf{CaTT}}$ whose objects are the ps-contexts

Theorem (B., Finster, Mimram)

The category PS is equivalent to $\Theta_\infty^{\mathrm{op}}$

Models of the theory

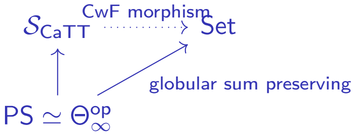
Theorem (B., Finster, Mimram)

The models of CaTT are equivalent to the G.-M. weak ω-categories

Models of the theory

Theorem (B., Finster, Mimram)

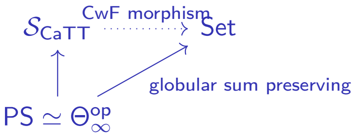
The models of CaTT are equivalent to the G.-M. weak ω-categories



Models of the theory

Theorem (B., Finster, Mimram)

The models of CaTT are equivalent to the G.-M. weak ω-categories



Proved by showing the initiality theorem for the theory CaTT.

The syntactic category

- The category $\mathcal{S}_{\mathsf{CaTT}}$ naturally appears in our presentation
  What is its significance ?

The syntactic category

- The category $\mathcal{S}_{\mathsf{CaTT}}$ naturally appears in our presentation
  What is its significance ?

- The Yoneda embedding provides an inclusion of $\mathcal{S}_{\mathsf{CaTT}}^{\mathsf{op}}$ into its models the weak ω-categories.
  Conjecture : the syntactic category is the opposite of the subcategory of the weak ω-categories freely by finite computads, for an appropriate notion of computad.

The syntactic category

- The category $\mathcal{S}_{\mathsf{CaTT}}$ naturally appears in our presentation
  What is its significance ?

- The Yoneda embedding provides an inclusion of $\mathcal{S}_{\mathsf{CaTT}}^{\mathrm{op}}$ into its models the weak ω-categories.
  Conjecture : the syntactic category is the opposite of the subcategory of the weak ω-categories freely by finite computads, for an appropriate notion of computad.

- There is work conducted around this conjecture and extension of CaTT.
  Ongoing work related to this question and CaTT by Finster, Vicary, Markakis, Rice

Thank you !

References I

Dimitri Ara.
Sur les ∞-groupoïdes de Grothendieck et une variante ∞-catégorique.
PhD thesis, Université Paris 7, 2010.

Guillaume Brunerie.
On the homotopy groups of spheres in homotopy type theory.
arXiv preprint arXiv :1606.05916, 2016.

Peter Dybjer.
Internal Type Theory.
In *Types for Proofs and Programs. TYPES 1995*, pages 120–134. Springer, Berlin, Heidelberg, 1996.

References II

Eric Finster and Samuel Mimram.
A Type-Theoretical Definition of Weak ω-Categories.
In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12, 2017.

Alexander Grothendieck.
Pursuing stacks.
Unpublished manuscript, 1983.

Tom Leinster.
Higher operads, higher categories, volume 298.
Cambridge University Press, 2004.

Georges Maltsiniotis.
Grothendieck ∞-groupoids, and still another definition of ∞-categories.
Preprint arXiv:1009.2331, 2010.