Key idea: Type theory is a *syntactic* and *computational* formal system which describes $\infty$-groupoids (that is, higher categories with all invertible morphisms).

Solves problems of coherence by *disallowing* statements which would violate it.

What would a syntactic description of higher *categories* look like?
Opetopes

- Idea: focus on a model of higher category theory which is already syntactic in nature.
- Opetopic models of higher categories are based on higher dimensional trees as opposed to simplices.
- They can be thought of (roughly) as the continuation of the sequence
  
  character, string, tree, ... 

- Fix an alphabet $\Sigma = \{a, b, c, \ldots\}$
An Example Expression

Syntactic

Geometric

Eric Finster
Higher Dimensional Syntax
An Example Expression

Syntactic

Geometric

a
b

b

a

An Example Expression

Syntactic

Geometric
An Example Expression

Syntactic

Geometric

Eric Finster
Higher Dimensional Syntax
An Example Expression

Syntactic

Geometric

Diagram showing the syntactic and geometric representations of an example expression.
An Example Expression

Syntactic

Geometric
An Example Expression

Syntactic

Geometric
An Example Expression

Syntactic

Geometric
An Example Expression

Syntactic

Geometric
An Example Expression

Syntactic

Geometric
An Example Expression

Syntactic

Geometric

Eric Finster
Higher Dimensional Syntax
An Example Expression

Syntactic

Geometric

Higher Dimensional Syntax
An Example Expression

Syntactic

Geometric

Eric Finster
Higher Dimensional Syntax
An Example Expression

**Syntactic**

**Geometric**
An Example Expression

**Syntactic**

**Geometric**
An Example Expression

Syntactic

Geometric

Eric Finster
Higher Dimensional Syntax
Special Forms

Web
Special Forms

Pasting Diagram
Special Forms

Frame
Special Forms

Cell
Unlabelled cell diagrams are called *opetopes*

Taking “opetopic expressions” to be our fundamental syntactic unit, what should derivations rules look like?
A Pasting Diagram
Composition

A Pasting Diagram

It's Composite
Example: the free category on a graph
Example: the free category on a graph

Axioms

Pasting Diagram

Term

Eric Finster | Higher Dimensional Syntax
Example: the free category on a graph

Axioms

Pasting Diagram
Example: the free category on a graph

### Axioms

- \( G \quad a \)
- \( G \quad b \)
- \( f \)
- \( g \)

### Pasting Diagram

### Term
Given a frame:

\[
\begin{array}{c}
X \\
\begin{array}{c}
\begin{array}{c}
\frame{a}{b}{c} \\
\frame{f}{g}{h}
\end{array}
\end{array}
\end{array}
\]

One has a new type.
Given a frame:

One has a new type.

Eric Finster
Higher Dimensional Syntax
Given a frame:

One has a new type.

\[
\begin{array}{c}
X \\
\text{a} \\
\text{b} \\
\text{c} \\
f \\
g \\
h \\
k \\
\ldots
\end{array}
\]

\[
X \ a \ b = \text{Id}_{(a,b)}
\]

Eric Finster
Higher Dimensional Syntax
Given a frame:

One has a new type.

One has a new type.

Given a frame:

One has a new type.

Given a frame:
Example: Binary Trees

data BT where
  R : BT
  B : BT -> BT -> BT
Example: Binary Trees

data BT where
  R : BT
  B : BT -> BT -> BT
Example: Binary Trees

data BT where
  R : BT
  B : BT -> BT -> BT
Example: Binary Trees

data BT where
  R : BT
  B : BT -> BT -> BT

Eric Finster  Higher Dimensional Syntax
Example: Binary Trees

data BT where
  R : BT
  B : BT -> BT -> BT
data BT where
  R : BT
  B : BT -> BT -> BT
Example: Binary Trees

data BT where
R : BT
B : BT -> BT -> BT
Example: Binary Trees

```haskell
data BT where
  R : BT
  B : BT -> BT -> BT
```

Eric Finster
Higher Dimensional Syntax
Example: Vectors

data Vec (A : Set) : Nat -> Set where
  Nil : Vec A 0
  Cons : A -> Vec A n -> Vec A (n + 1)
Open Problems

- Find a theory of higher functions. (Higher λ-calculus??)
- What are cofree/coinductive definitions?
- Semantic Theorems