A sequent calculus for opetopes

CT 2019

Pierre-Louis Curien[1] **Cédric Ho Thanh**[1] Samuel Mimram[2]

July 9th, 2019

[1]IRIF, Paris University

[2]LIX, École Polytechnique

This presentation informally presents some of the main notions and results of [Curien et al., 2019] `arXiv:1903.05848`, namely a **“unnamed” syntax for opetopes**, and a sequent calculus $\mathsf{Opt}^?$.

Contents

Opetopes

In a nutshell...

Opetopes are shapes (akin to globules, cubes, simplices, dendrices, etc.) designed to represent the notion of composition in every dimension. As such, they were introduced in [Baez and Dolan, 1998] to describe laws and coherence in weak higher categories.

In a nutshell...

Opetopes are shapes (akin to globules, cubes, simplices, dendrices, etc.) designed to represent the notion of composition in every dimension. As such, they were introduced in [Baez and Dolan, 1998] to describe laws and coherence in weak higher categories.

They have been actively studied over the recent years in [Hermida et al., 2002], [Cheng, 2003], [Leinster, 2004], [Kock et al., 2010] and applied to the theory of polygraphs in [Ho Thanh, 2018a].

In a nutshell...

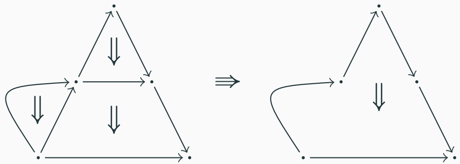
Opetopes are shapes (akin to globules, cubes, simplices, dendrices, etc.) designed to represent the notion of composition in every dimension. As such, they were introduced in [Baez and Dolan, 1998] to describe laws and coherence in weak higher categories.

They have been actively studied over the recent years in [Hermida et al., 2002], [Cheng, 2003], [Leinster, 2004], [Kock et al., 2010] and applied to the theory of polygraphs in [Ho Thanh, 2018a].

A first syntactic account of opetopes has been tried in [Hermida et al., 2002], but does not seem usable for any computation.

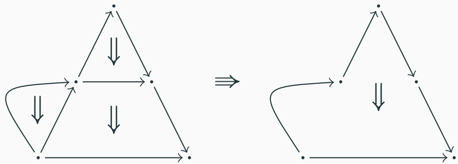
Informal definition

They are **pasting diagrams** where every cell is **many-to-one** i.e. many inputs, one output. Here is an example of a 3-opetope:



Informal definition

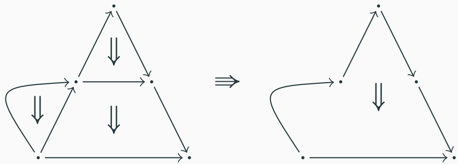
They are **pasting diagrams** where every cell is **many-to-one** i.e. many inputs, one output. Here is an example of a 3-opetope:



Every cell denoted by a $\Downarrow$ above has dimension 2, so that a 3-**opetope really is a pasting diagram of cells of dimension** 2.

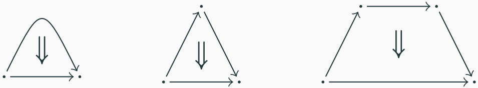
Informal definition

They are **pasting diagrams** where every cell is **many-to-one** i.e. many inputs, one output. Here is an example of a 3-opetope:

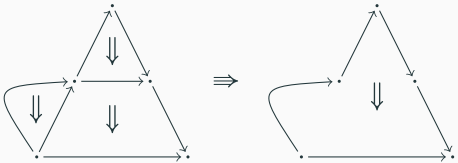


Every cell denoted by a $\Downarrow$ above has dimension 2, so that a 3**-opetope really is a pasting diagram of cells of dimension** 2.

We further ask those cells of dimension 2 **to be** 2**-opetopes**, i.e. pasting diagram of cells of dimension 1 (the simple arrows $\rightarrow$).



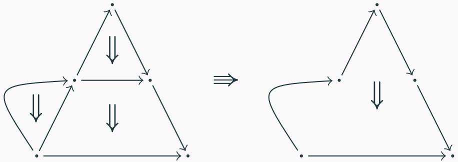
Informal definition



Definition

An n-**dimensional opetope** (or just n-**opetope**) is a pasting diagram of $(n-1)$-opetopes,

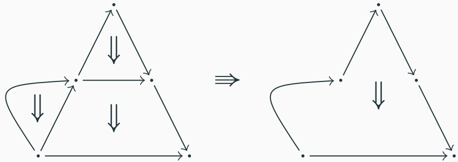
Informal definition



Definition

An n-**dimensional opetope** (or just n-**opetope**) is a pasting diagram of $(n-1)$-opetopes, i.e. a finite set of $(n-1)$-opetopes glued along $(n-2)$-opetopes,

Informal definition



Definition

An n-**dimensional opetope** (or just n-**opetope**) is a pasting diagram of $(n-1)$-opetopes, i.e. a finite set of $(n-1)$-opetopes glued along $(n-2)$-opetopes, in a “well-defined manner”.

Definition: low dimensions

- There is a unique 0-dimensional opetope, which we'll call the **point**:

$$\bullet$$

Definition: low dimensions

- There is a unique 0-dimensional opetope, which we'll call the **point**:

$$\cdot$$

- There is a unique 1-opetope, the **arrow**:

$$\cdot \longrightarrow \cdot$$

Definition: low dimensions

- There is a unique 0-dimensional opetope, which we'll call the **point**:

$$\cdot$$

- There is a unique 1-opetope, the **arrow**:

$$\cdot \longrightarrow \cdot$$

- 2-opetopes are pasting diagram of 1-opetopes:

$$\mathbf{3} \qquad = \qquad \begin{tikzcd} & \cdot \arrow[r] \arrow[dr, phantom, "\Downarrow"] & \cdot \arrow[dr] & \\ \cdot \arrow[ur] \arrow[rrr] & & & \cdot \end{tikzcd}$$

Definition: low dimensions

- There is a unique 0-dimensional opetope, which we'll call the **point**:

$$\cdot$$

- There is a unique 1-opetope, the **arrow**:

$$\cdot \longrightarrow \cdot$$

- 2-opetopes are pasting diagram of 1-opetopes:

$$\mathbf{2} \qquad = \qquad \begin{tikzcd} & \cdot \arrow[dr] & \\ \cdot \arrow[ur] \arrow[rr] \arrow[rr, phantom, shift left=3ex, "\Downarrow"] & & \cdot \end{tikzcd}$$

Definition: low dimensions

- There is a unique 0-dimensional opetope, which we'll call the **point**:

$$\cdot$$

- There is a unique 1-opetope, the **arrow**:

$$\cdot \longrightarrow \cdot$$

- 2-opetopes are pasting diagram of 1-opetopes:

$$\mathbf{1} \qquad = \qquad \begin{tikzcd} \cdot \arrow[r, bend left=80] \arrow[r] \arrow[r, phantom, bend left=40, "\Downarrow"] & \cdot \end{tikzcd}$$

Definition: low dimensions

- There is a unique 0-dimensional opetope, which we'll call the **point**:

$$\cdot$$

- There is a unique 1-opetope, the **arrow**:

$$\cdot \longrightarrow \cdot$$

- 2-opetopes are pasting diagram of 1-opetopes:

$$\mathsf{n} \qquad = \qquad \begin{tikzcd} & \cdot \arrow[r, "(n-1)"] & \cdot & \cdots & \cdot \arrow[dr, "(1)"] & \\ \cdot \arrow[ur, "(n)"] \arrow[rrrrr] & & \arrow[u, phantom, "\Downarrow"] & & & \cdot \end{tikzcd}$$

Definition: low dimensions

- There is a unique 0-dimensional opetope, which we'll call the **point**:

$$\cdot$$

- There is a unique 1-opetope, the **arrow**:

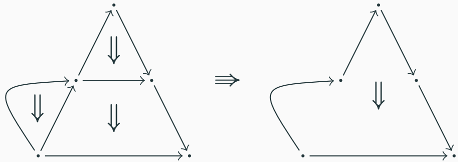
$$\cdot \longrightarrow \cdot$$

- 2-opetopes are pasting diagram of 1-opetopes:

$$\mathbf{0} \qquad = $$

Definition: dimension 3

- 3-opetopes are pasting diagrams of 2-opetopes

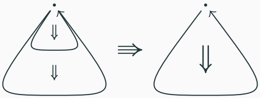


- 3-opetopes are pasting diagrams of 2-opetopes

$$
\begin{tikzcd}
 & \cdot \arrow[rr] & & \cdot \arrow[dr] & \\
\cdot \arrow[ur] \arrow[urrr] \arrow[rrrr] \arrow[ur, phantom, shift left=6, "\Downarrow"] & & \arrow[ur, phantom, shift right=4, "\Downarrow"] & & \cdot
\end{tikzcd}
\Rrightarrow
\begin{tikzcd}
 & \cdot \arrow[rr] \arrow[dr, phantom, "\Downarrow"] & & \cdot \arrow[dr] & \\
\cdot \arrow[ur] \arrow[rrrr] & & {} & & \cdot
\end{tikzcd}
$$

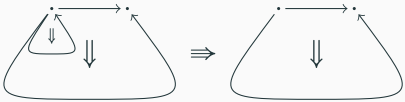
Definition: dimension 3

- 3-opetopes are pasting diagrams of 2-opetopes



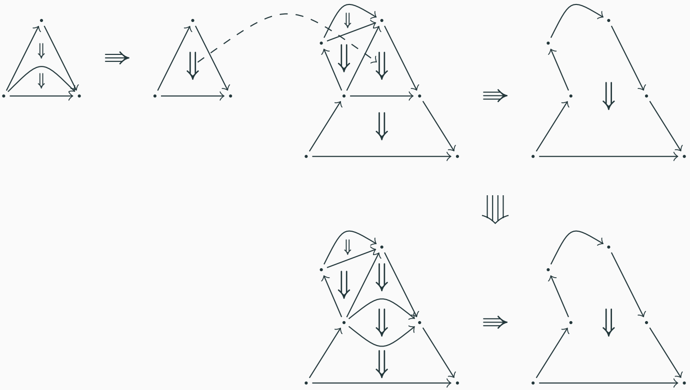
Definition: dimension 3

- 3-opetopes are pasting diagrams of 2-opetopes



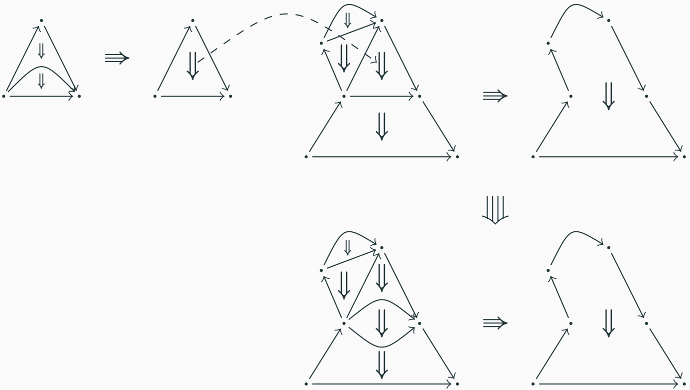
Definition: dimension 4

- The induction goes on: 4-opetopes are pasting diagrams of 3-opetopes:



Definition: dimension 4

- The induction goes on: 4-opetopes are pasting diagrams of 3-opetopes:



This is getting out of hand…

Motivation

Problem

1. The graphical approach is neither formal nor manageable for dimensions ≥ 4.

Motivation

Problem

1. The graphical approach is neither formal nor manageable for dimensions ≥ 4.
2. A formal definition either uses T-operads [Leinster, 2004] or polynomial monads and trees [Kock et al., 2010], which as is, are not suited for automated computations.

Motivation

Problem

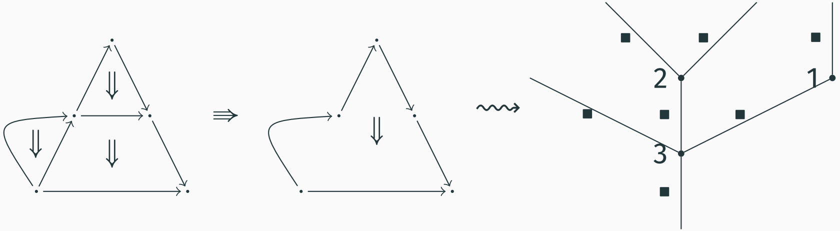
1. The graphical approach is neither formal nor manageable for dimensions ≥ 4.
2. A formal definition either uses T-operads [Leinster, 2004] or polynomial monads and trees [Kock et al., 2010], which as is, are not suited for automated computations.

Solution

In this presentation, we give a way to define opetopes syntactically.

Syntax

Since opetopes are pasting diagrams whose cells are **many-to-one**, they can be represented as trees:



Idea: dimension 0 and 1

Denote by $\blacklozenge$ the unique 0-opetope, a.k.a. the point:

$$\cdot$$

Idea: dimension 0 and 1

Denote by $\blacklozenge$ the unique 0-opetope, a.k.a. the point:

$$\cdot$$

and by $\blacksquare$ the unique 1-opetope, a.k.a. the arrow:

$$\cdot \longrightarrow \cdot$$

Idea: dimension 0 and 1

Denote by $\blacklozenge$ the unique 0-opetope, a.k.a. the point:

$$\cdot$$

and by $\blacksquare$ the unique 1-opetope, a.k.a. the arrow:

$$\cdot \longrightarrow \cdot$$

We can represent $\blacksquare$ as a node of a tree as follows:

Idea: dimension 0 and 1

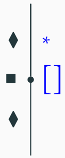
Denote by ♦ the unique 0-opetope, a.k.a. the point:

$$\cdot$$

and by ■ the unique 1-opetope, a.k.a. the arrow:

$$\cdot \longrightarrow \cdot$$

We can represent ■ as a node of a tree as follows:

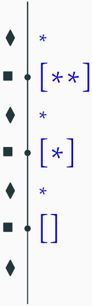


Let us add address information.

Idea: dimension 2

Then we can:

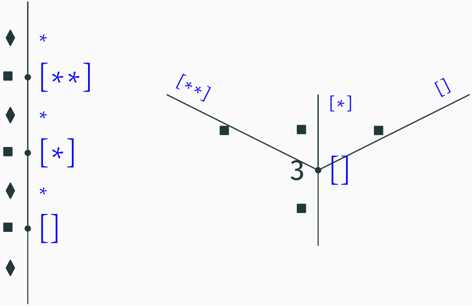
1. create a tree with that corolla representing ■



Idea: dimension 2

Then we can:

1. create a tree with that corolla representing ■

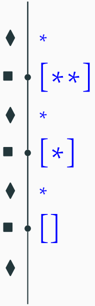


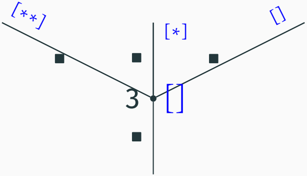
2. consider that tree as a corolla, where the input edges are the nodes

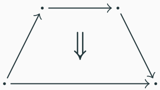
Idea: dimension 2

Then we can:

1. create a tree with that corolla representing ■

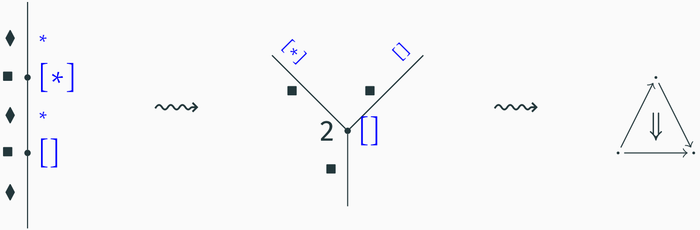






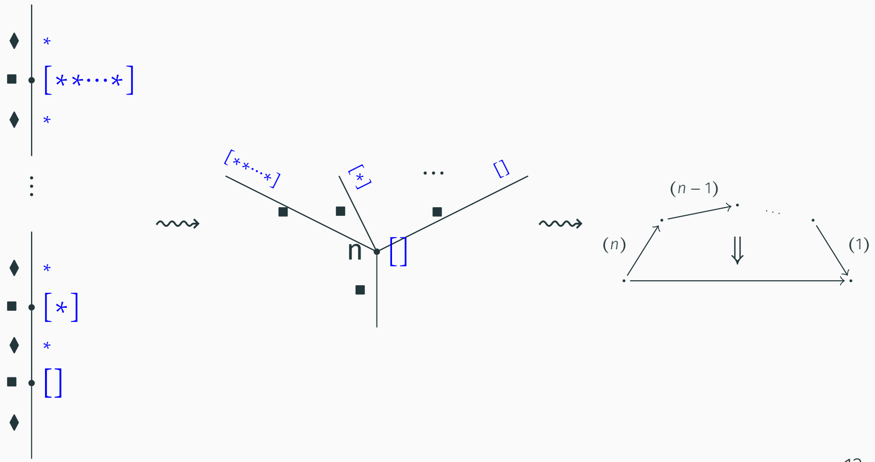
2. consider that tree as a corolla, where the input edges are the nodes
3. be convinced that this is a good representation of some 2-opetope!

Depending on the original tree, we obtain different 2-opetopes:



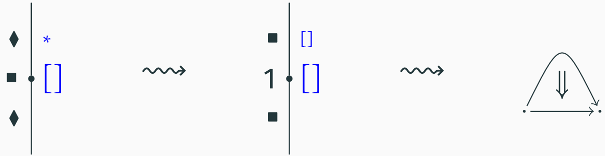
Idea: dimension 2

Depending on the original tree, we obtain different 2-opetopes:

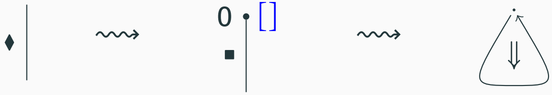


Idea: dimension 2

Depending on the original tree, we obtain different 2-opetopes:

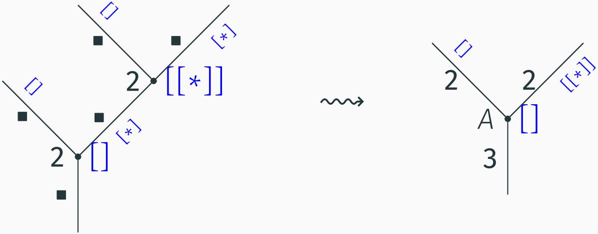


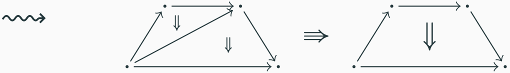
Depending on the original tree, we obtain different 2-opetopes:



Idea: dimension 3

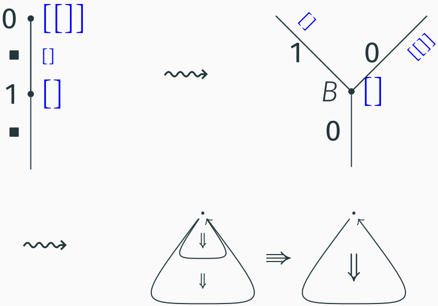
From there, repeat the process!



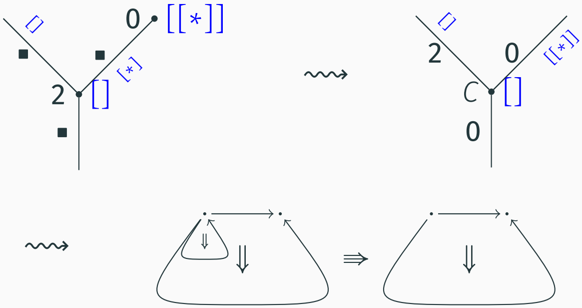


Idea: dimension 3

From there, repeat the process!

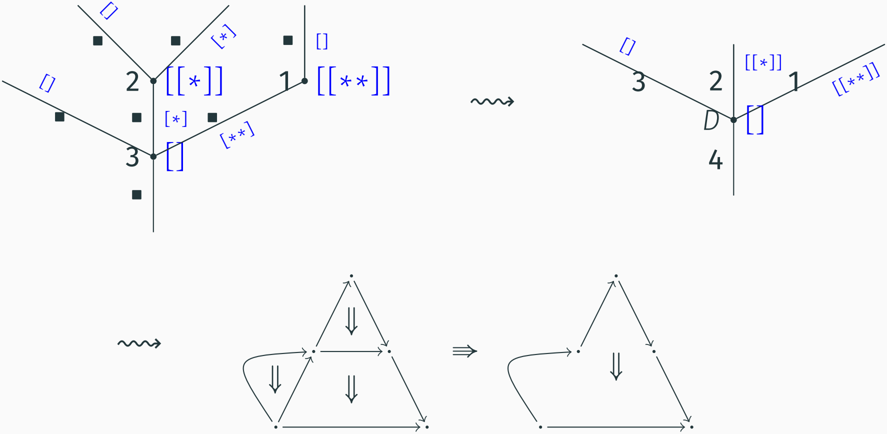


From there, repeat the process!



Idea: dimension 3

From there, repeat the process!

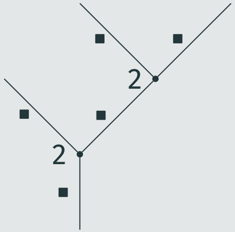


Syntax

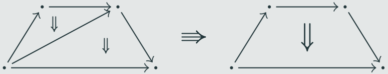
We now want a syntactic description of such trees.

We now want a syntactic description of such trees.

Solution



$\rightsquigarrow$

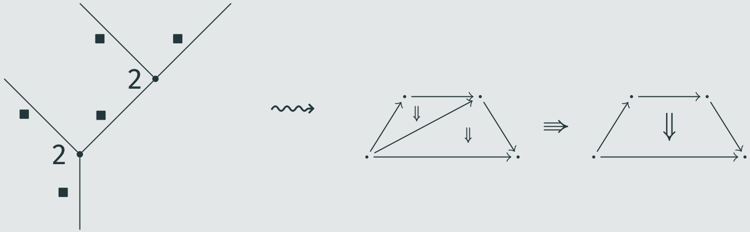


In an n-opetope, every node is decorated by $(n-1)$-opetope,

Syntax

We now want a syntactic description of such trees.

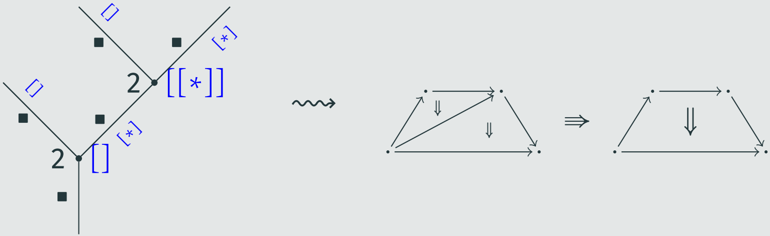
Solution



In an n-opetope, every node is decorated by $(n-1)$-opetope, but $(n-1)$-opetope does not uniquely identify a node.

We now want a syntactic description of such trees.

Solution

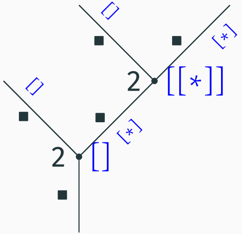


In an n-opetope, every node is decorated by $(n-1)$-opetope, but $(n-1)$-opetope does not uniquely identify a node. But addresses do! So we just need to describe a partial map

$$\mathbb{A} \longrightarrow \mathbb{O}_{n-1}.$$

Syntax

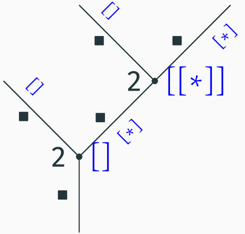
We encode opetopes recursively as follows:



$$\rightsquigarrow \quad \begin{cases} [] \leftarrow 2 \\ [[*]] \leftarrow 2 \end{cases}$$

Syntax

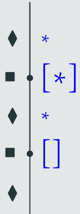
We encode opetopes recursively as follows:



$$\rightsquigarrow \quad \begin{cases} [\,] \leftarrow 2 \\ [[*]] \leftarrow 2 \end{cases}$$

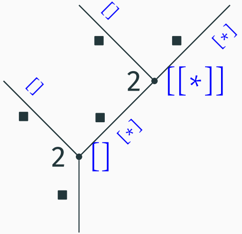
Reminder

$2 \quad =$



Syntax

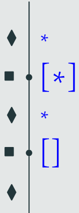
We encode opetopes recursively as follows:



$$\rightsquigarrow \quad \begin{cases} [] \leftarrow 2 \\ [[*]] \leftarrow 2 \end{cases}$$

Reminder

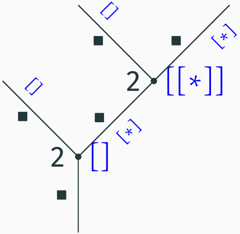
$$2 \quad = $$



$$= \quad \begin{cases} [] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases}$$

Syntax

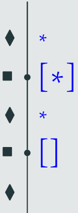
We encode opetopes recursively as follows:



$$\rightsquigarrow \begin{cases} [] \leftarrow \begin{cases} [] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases} \\ [[*]] \leftarrow \begin{cases} [] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases} \end{cases}$$

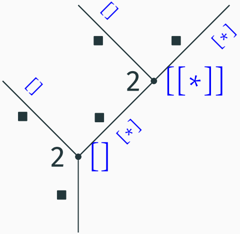
Reminder

$$2 \quad =$$



$$= \begin{cases} [] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases}$$

We encode opetopes recursively as follows:



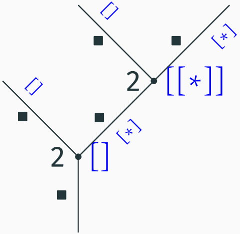
$$\rightsquigarrow \quad \begin{cases} [] \leftarrow \begin{cases} [] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases} \\ [[*]] \leftarrow \begin{cases} [] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases} \end{cases}$$

Convention

$$\blacksquare \quad = \quad \begin{cases} * \leftarrow \blacklozenge \end{cases}$$

Syntax

We encode opetopes recursively as follows:

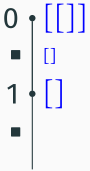


$$\rightsquigarrow \quad \begin{cases} [] \leftarrow \begin{cases} [] \leftarrow \begin{cases} * \leftarrow \blacklozenge \end{cases} \\ [*] \leftarrow \begin{cases} * \leftarrow \blacklozenge \end{cases} \end{cases} \\ [[*]] \leftarrow \begin{cases} [] \leftarrow \begin{cases} * \leftarrow \blacklozenge \end{cases} \\ [*] \leftarrow \begin{cases} * \leftarrow \blacklozenge \end{cases} \end{cases} \end{cases}$$

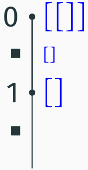
Convention

$$\blacksquare \quad = \quad \begin{cases} * \leftarrow \blacklozenge \end{cases}$$

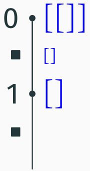
Syntax: examples



Syntax: examples



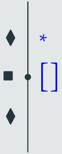
$$\rightsquigarrow \quad \begin{cases} [\,] \leftarrow 1 \\ [[\,]] \leftarrow 0 \end{cases}$$



$$\rightsquigarrow \quad \begin{cases} [\,] \leftarrow 1 \\ [[\,]] \leftarrow 0 \end{cases}$$

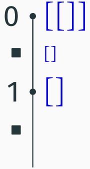
Reminder

$1 \quad = $



$= \quad \left\{ [\,] \leftarrow \blacksquare \right.$

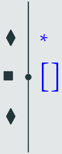
Syntax: examples



$$\rightsquigarrow \quad \left\{ \begin{array}{l} [\,] \leftarrow \left\{ [\,] \leftarrow \blacksquare \right. \\ [[\,]] \leftarrow 0 \end{array} \right.$$

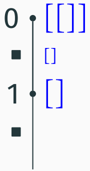
Reminder

$$1 \quad = $$



$$= \quad \left\{ [\,] \leftarrow \blacksquare \right.$$

Syntax: examples



$$\rightsquigarrow \quad \begin{cases} [\,] \leftarrow \begin{cases} [\,] \leftarrow \blacksquare \end{cases} \\ [[\,]] \leftarrow 0 \end{cases}$$

Reminder

$$\blacksquare \quad = \quad \begin{cases} * \leftarrow \blacklozenge \end{cases}$$

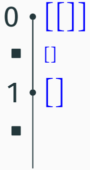
Syntax: examples

$$\begin{array}{r|l} 0 & [[\,]] \\ \blacksquare & [\,] \\ 1 & [\,] \\ \blacksquare & \end{array} \quad \rightsquigarrow \quad \begin{cases} [\,] \leftarrow \{ [\,] \leftarrow \{ * \leftarrow \blacklozenge \\ [[\,]] \leftarrow 0 \end{cases}$$

Reminder

$$\blacksquare \quad = \quad \{ * \leftarrow \blacklozenge$$

Syntax: examples

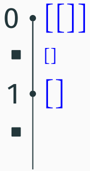


$$\rightsquigarrow \quad \left\{ \begin{array}{l} [\,] \leftarrow \left\{ [\,] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \right. \\ [[\,]] \leftarrow 0 \end{array} \right.$$

Reminder + convention

$$0 \quad = \quad \blacklozenge \Big| \quad = \quad \{\{\blacklozenge$$

Syntax: examples

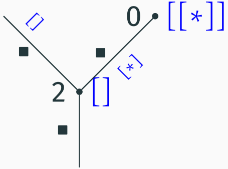


$$\rightsquigarrow \quad \left\{\begin{array}{l} [\,] \leftarrow \left\{ [\,] \leftarrow \left\{ * \leftarrow \blacklozenge \right.\right. \\ [[\,]] \leftarrow \{\!\{ \blacklozenge \end{array}\right.$$

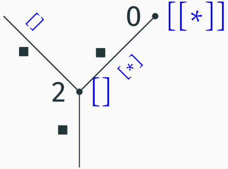
Reminder + convention

$$0 \quad = \quad \blacklozenge \Big| \quad = \quad \{\!\{ \blacklozenge$$

Syntax: examples

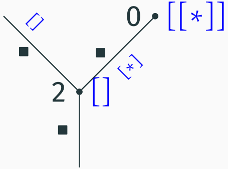


Syntax: examples



$$\rightsquigarrow \quad \begin{cases} [\,] \leftarrow 2 \\ [[*]] \leftarrow 0 \end{cases}$$

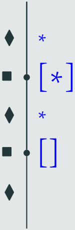
Syntax: examples



$$\rightsquigarrow \quad \begin{cases} [\,] \leftarrow 2 \\ [[*]] \leftarrow 0 \end{cases}$$

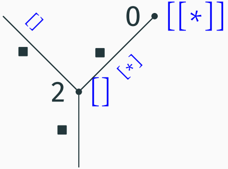
Reminder

$2 =$



$$= \begin{cases} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases}$$

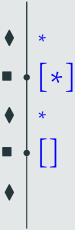
Syntax: examples



$$\rightsquigarrow \quad \left\{\begin{array}{l} [\,] \leftarrow \left\{\begin{array}{l} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{array}\right. \\ [[*]] \leftarrow 0 \end{array}\right.$$

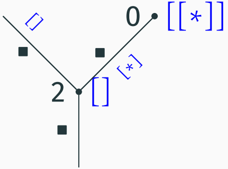
Reminder

$$2 \quad =$$



$$= \quad \left\{\begin{array}{l} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{array}\right.$$

Syntax: examples

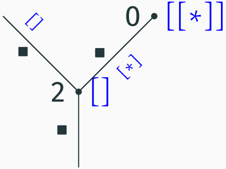


$$\rightsquigarrow \quad \begin{cases} [\,] \leftarrow \begin{cases} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases} \\ [[*]] \leftarrow 0 \end{cases}$$

Reminder

$$\blacksquare \quad = \quad \begin{cases} * \leftarrow \blacklozenge \end{cases}$$

Syntax: examples

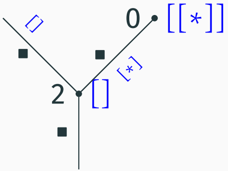


$$\rightsquigarrow \quad \begin{cases} [\,] \leftarrow \begin{cases} [\,] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \\ [*] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \end{cases} \\ [[*]] \leftarrow 0 \end{cases}$$

Reminder

$$\blacksquare \quad = \quad \left\{ * \leftarrow \blacklozenge \right.$$

Syntax: examples

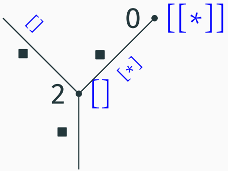


$$\rightsquigarrow \quad \begin{cases} [\,] \leftarrow \begin{cases} [\,] \leftarrow \{ * \leftarrow \blacklozenge \\ [*] \leftarrow \{ * \leftarrow \blacklozenge \end{cases} \\ [[*]] \leftarrow 0 \end{cases}$$

Reminder

$0 =$  $= \{\{\blacklozenge$

Syntax: examples

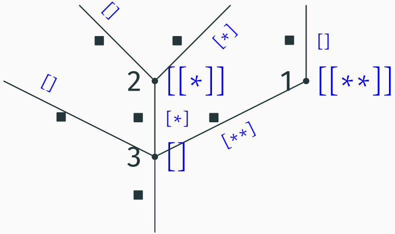


$$\rightsquigarrow \quad \begin{cases} [\,] \leftarrow \begin{cases} [\,] \leftarrow \{ * \leftarrow \blacklozenge \\ [*] \leftarrow \{ * \leftarrow \blacklozenge \end{cases} \\ [[*]] \leftarrow \{\!\{ \blacklozenge \end{cases}$$

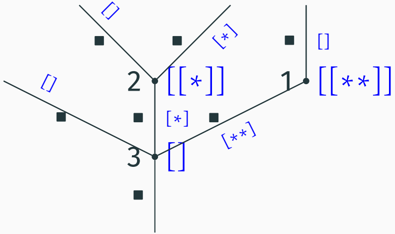
Reminder

$$0 \quad = \quad \blacklozenge \Big| \quad = \quad \{\!\{ \blacklozenge$$

Syntax: examples

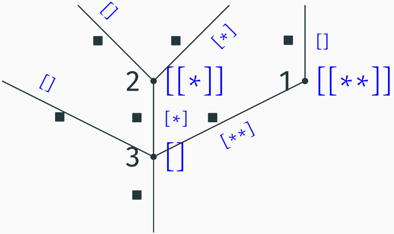


Syntax: examples



$$\rightsquigarrow \quad \begin{cases} [\,] \leftarrow 3 \\ [[\ast]] \leftarrow 2 \\ [[\ast\ast]] \leftarrow 1 \end{cases}$$

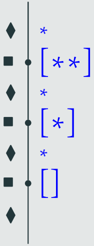
Syntax: examples



$$\rightsquigarrow \quad \begin{cases} [\,] \leftarrow 3 \\ [[*]] \leftarrow 2 \\ [[**]] \leftarrow 1 \end{cases}$$

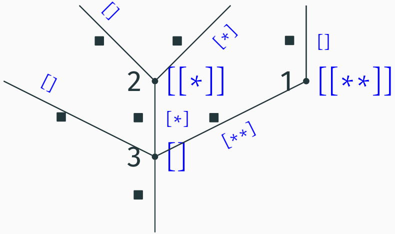
Reminder

$3 \quad =$



$$= \quad \begin{cases} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare \end{cases}$$

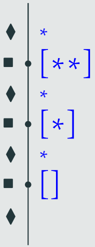
Syntax: examples



$$\rightsquigarrow \begin{cases} [\,] \leftarrow \begin{cases} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare \end{cases} \\ [[*]] \leftarrow 2 \\ [[**]] \leftarrow 1 \end{cases}$$

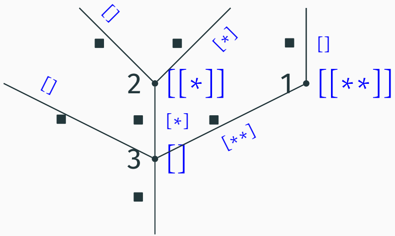
Reminder

$$3 \quad =$$



$$= \begin{cases} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare \end{cases}$$

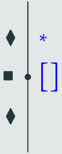
Syntax: examples



$$\rightsquigarrow \left\{ \begin{array}{l} [\,] \leftarrow \left\{ \begin{array}{l} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare \end{array} \right. \\ [[*]] \leftarrow 2 \\ [[**]] \leftarrow 1 \end{array} \right.$$

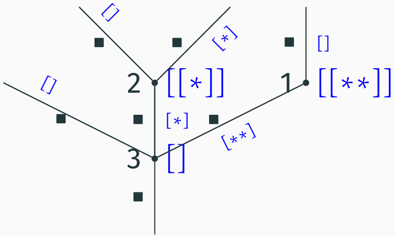
Reminder

$$1 \quad = $$



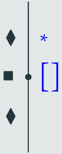
$$= \quad \left\{ [\,] \leftarrow \blacksquare \right.$$

Syntax: examples

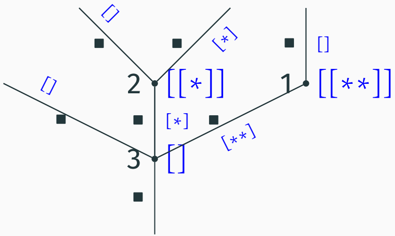


$$\rightsquigarrow \begin{cases} [] \leftarrow \begin{cases}[] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare\end{cases} \\ [[*]] \leftarrow 2 \\ [[**]] \leftarrow \begin{cases}[] \leftarrow \blacksquare\end{cases} \end{cases}$$

Reminder

$1 =$  $= \begin{cases}[] \leftarrow \blacksquare\end{cases}$

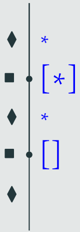
Syntax: examples



$$\rightsquigarrow \begin{cases} [\,] \leftarrow \begin{cases} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare \end{cases} \\ [[*]] \leftarrow 2 \\ [[**]] \leftarrow \begin{cases} [\,] \leftarrow \blacksquare \end{cases} \end{cases}$$

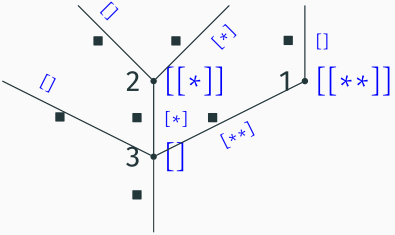
Reminder

$$2 =$$



$$= \begin{cases} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases}$$

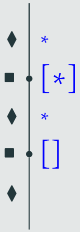
Syntax: examples



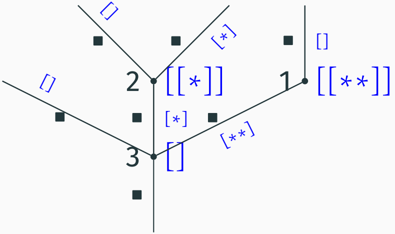
$$\rightsquigarrow \begin{cases} [\,] \leftarrow \begin{cases} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare \end{cases} \\ [[*]] \leftarrow \begin{cases} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases} \\ [[**]] \leftarrow \left\{ [\,] \leftarrow \blacksquare \right. \end{cases}$$

Reminder

$$2 \quad =$$



$$= \begin{cases} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases}$$

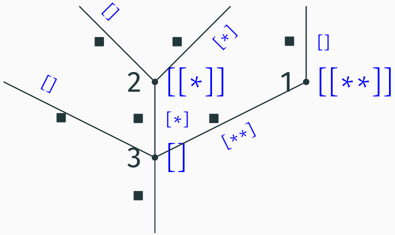


$$\rightsquigarrow \quad \left\{ \begin{array}{l} [\,] \leftarrow \left\{ \begin{array}{l} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare \end{array} \right. \\ [[*]] \leftarrow \left\{ \begin{array}{l} [\,] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{array} \right. \\ [[**]] \leftarrow \left\{ [\,] \leftarrow \blacksquare \right. \end{array} \right.$$

Reminder

$$\blacksquare \quad = \quad \left\{ * \leftarrow \blacklozenge \right.$$

Syntax: examples



$$\leadsto \begin{cases} [] \leftarrow \begin{cases} [] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \\ [*] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \\ [**] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \end{cases} \\ [[*]] \leftarrow \begin{cases} [] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \\ [*] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \end{cases} \\ [[**]] \leftarrow \left\{ [] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \right. \end{cases}$$

Reminder

$$\blacksquare \quad = \quad \left\{ * \leftarrow \blacklozenge \right.$$

Syntax

Question

Is this an opetope?

$$
\left\{
\begin{array}{l}
[] \leftarrow \left\{ \begin{array}{l} [*] \leftarrow \blacklozenge \\ [**] \leftarrow \blacklozenge \\ [***] \leftarrow \blacklozenge \end{array} \right. \\
[**] \leftarrow \big\{ [] \leftarrow \big\{ [] \leftarrow \big\{ [] \leftarrow \big\{ [] \leftarrow \big\{ [] \leftarrow \big\{ [] \leftarrow \big\{ [] \leftarrow \big\{ [] \leftarrow \blacklozenge \\
[***] \leftarrow \left\{ \begin{array}{l}
[] \leftarrow \left\{ \begin{array}{l} [] \leftarrow \big\{ [] \leftarrow \blacklozenge \\ [*] \leftarrow \blacklozenge \\ [**] \leftarrow \blacklozenge \end{array} \right. \\
[[]] \leftarrow \big\{ [] \leftarrow \big\{ * \leftarrow \blacklozenge \\
[[[]]] \leftarrow \left\{ \begin{array}{l} [[[*]]] \leftarrow \big\{ * \leftarrow \blacklozenge \\ [*] \leftarrow \big\{ * \leftarrow \blacklozenge \end{array} \right.
\end{array} \right. \\
[[***]] \leftarrow \blacklozenge
\end{array}
\right.
$$

$\mathrm{Opt}^?$: a sequent calculus for opetopes

System Opt$^?$

The set of preopetopes $\mathbb{P}$ is defined by the following grammar:

$$
\begin{aligned}
\mathbb{P} \quad ::= \quad & \blacklozenge \\
\mid \quad & \begin{cases} \mathbb{A} \leftarrow \mathbb{P} \\ \vdots \\ \mathbb{A} \leftarrow \mathbb{P} \end{cases} \\
\mid \quad & \{\!\{ \mathbb{P}
\end{aligned}
$$

The set of preopetopes $\mathbb{P}$ is defined by the following grammar:

$$
\begin{array}{rcl}
\mathbb{P} & ::= & \blacklozenge \\
& | & \begin{cases} \mathbb{A} \leftarrow \mathbb{P} \\ \vdots \\ \mathbb{A} \leftarrow \mathbb{P} \end{cases} \\
& | & \{\!\{ \mathbb{P}
\end{array}
$$

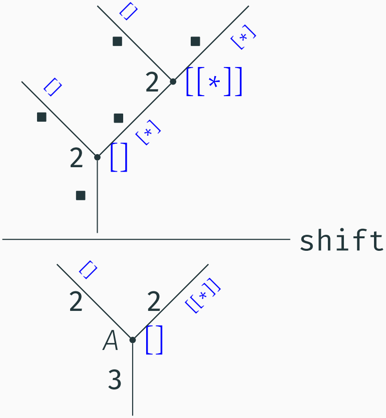
System $\mathsf{Opt}^?$ aims to characterize preopetopes that actually are opetopes.

The first rule of Opt$^?$ states that we may create points without any prior assumption:

$$\frac{}{\cdot}\ \texttt{point} \qquad\qquad \frac{}{\blacklozenge}\ \texttt{point}$$

System $\mathrm{Opt}^?$: the `shift` rule

This rule takes an opetope p and produces a new opetope having a unique node, decorated in p:



$$\dfrac{\mathsf{p}}{\{[] \leftarrow \mathsf{p}}\ \mathtt{shift}$$

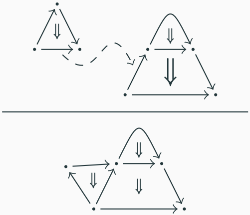
This rule takes an opetope and produces a degenerate opetope from it:

$$\dfrac{\cdot}{\qquad\qquad}\ \mathsf{degen}$$

$$\dfrac{\mathsf{p}}{\{\!\{\mathsf{p}}\ \mathsf{degen}$$

System $\mathfrak{Opt}^?$: the `graft` rule

This rule glues an n-opetope q to an $(n+1)$-opetope p, the latter really just being a pasting diagram of n-opetopes, and "glues" them together:



$$\dfrac{\begin{cases}[a_1] \leftarrow \mathsf{r}_1 \\ \vdots \\ [a_k] \leftarrow \mathsf{r}_k\end{cases} \qquad\qquad \mathsf{q}}{\begin{cases}[a_1] \leftarrow \mathsf{r}_1 \\ \vdots \\ [a_k] \leftarrow \mathsf{r}_k \\ [b] \leftarrow \mathsf{q}\end{cases}}\ \mathtt{graft}\text{-}[b]$$

Main result

Theorem

Derivable preopetopes in system $\mathsf{Opt}^?$ are in bijective correspondence with opetopes.

Examples

Examples

The proof tree of

$$\blacklozenge \quad = \quad .$$

is

$$\frac{}{\blacklozenge}\ \texttt{point}$$

Examples

The proof tree of

$$\blacksquare \quad = \quad \cdot \longrightarrow \cdot$$

is

$$\dfrac{\dfrac{}{\blacklozenge}\;\texttt{point}}{\{[\,] \leftarrow \blacklozenge}\;\texttt{shift}$$

Examples

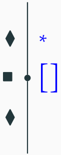
The proof tree of

$$\blacksquare \quad = \quad \cdot \longrightarrow \cdot$$

is

$$\dfrac{\dfrac{}{\blacklozenge}\ \texttt{point}}{\{* \leftarrow \blacklozenge}\ \texttt{shift}$$

The proof tree of

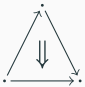
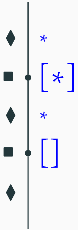
$1 \quad =$  $\rightsquigarrow$ 

is

$$\dfrac{\dfrac{\dfrac{}{\blacklozenge}\ \texttt{point}}{\{ * \leftarrow \blacklozenge}\ \texttt{shift}}{\{[\,] \leftarrow \{ * \leftarrow \blacklozenge}\ \texttt{shift}$$

Examples

The proof tree of

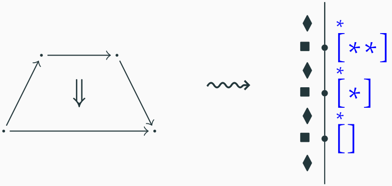
$2 \quad =$  $\rightsquigarrow$ 

is

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{}{\blacklozenge}\;\mathtt{point}
    }{
      \{ * \leftarrow \blacklozenge
    }\;\mathtt{shift}
  }{
    \{ [\,] \leftarrow \{ * \leftarrow \blacklozenge
  }\;\mathtt{shift}
  \qquad
  \dfrac{
    \dfrac{}{\blacklozenge}\;\mathtt{point}
  }{
    \{ [\,] \leftarrow \blacklozenge
  }\;\mathtt{shift}
}{
  \left\{
  \begin{array}{l}
  [\,] \leftarrow \{ * \leftarrow \blacklozenge \\
  [*] \leftarrow \{ * \leftarrow \blacklozenge
  \end{array}
  \right.
}\;\mathtt{graft\text{-}}[*]
$$

Examples

The proof tree of

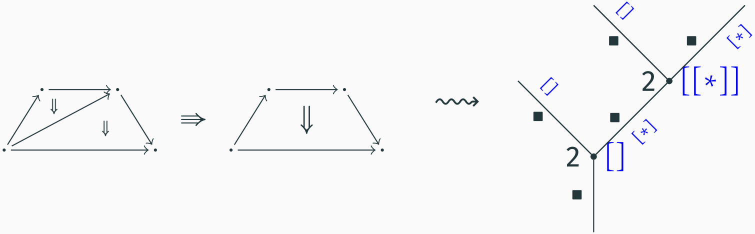
$3 =$ 

is

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\blacklozenge}\,\texttt{point}}{\{* \leftarrow \blacklozenge}\,\texttt{shift}}{\{[] \leftarrow \{* \leftarrow \blacklozenge}\,\texttt{shift} \qquad \dfrac{\dfrac{}{\blacklozenge}\,\texttt{point}}{\{[] \leftarrow \blacklozenge}\,\texttt{shift}}{\begin{cases} [] \leftarrow \{* \leftarrow \blacklozenge \\ [*] \leftarrow \{* \leftarrow \blacklozenge \end{cases}}\,\texttt{graft-}[*] \qquad \dfrac{\dfrac{}{\blacklozenge}\,\texttt{point}}{\{[] \leftarrow \blacklozenge}\,\texttt{shift}}{\begin{cases} [] \leftarrow \{* \leftarrow \blacklozenge \\ [*] \leftarrow \{* \leftarrow \blacklozenge \\ [**] \leftarrow \{* \leftarrow \blacklozenge \end{cases}}\,\texttt{graft-}[**]$$

Examples

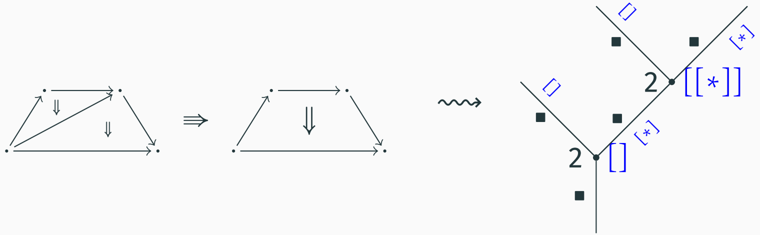
The proof tree of



is:

$$\begin{array}{c}\vdots\\ 2\end{array}$$

The proof tree of

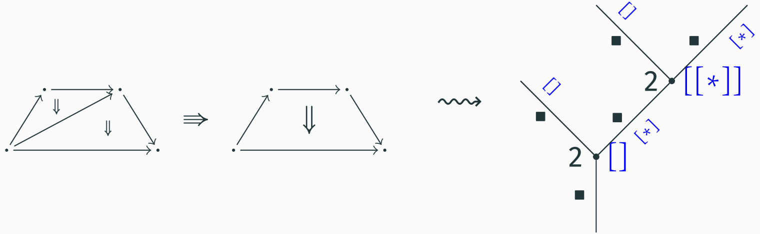


is:

$$\frac{\begin{array}{c}\vdots\\ 2\end{array}}{\{[\,] \leftarrow 2}\ \texttt{shift}$$

Examples

The proof tree of

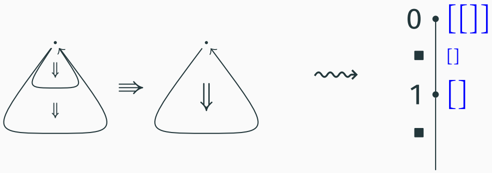


is:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ 2\end{array}}{\left\{[] \leftarrow 2\right.}\,\mathtt{shift} \qquad \begin{array}{c}\vdots\\ 2\end{array}}{\begin{cases}[] \leftarrow 2\\ [[*]] \leftarrow 2\end{cases}}\,\mathtt{graft\text{-}[[*]]}$$

Example

The proof tree of

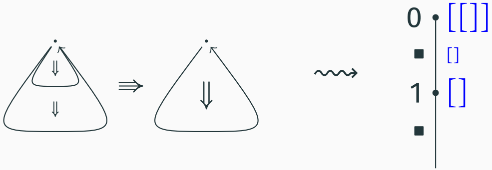


is

$$\vdots$$
$$1$$

Example

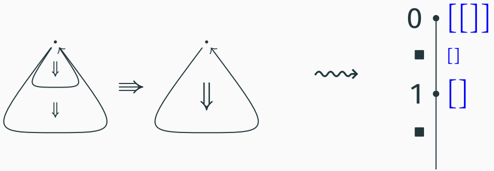
The proof tree of



is

$$\dfrac{\vdots \\ 1}{\{[] \leftarrow 1}\ \texttt{shift}$$

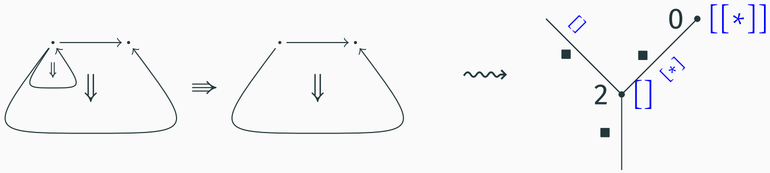
The proof tree of



is

$$\cfrac{\cfrac{\begin{array}{c}\vdots\\ 1\end{array}}{\left\{[] \leftarrow 1\right.}\,\texttt{shift} \qquad \begin{array}{c}\vdots\\ 0\end{array}}{\left\{\begin{array}{l}[] \leftarrow 1\\ [[]] \leftarrow 0\end{array}\right.}\,\texttt{graft-[[]]}$$

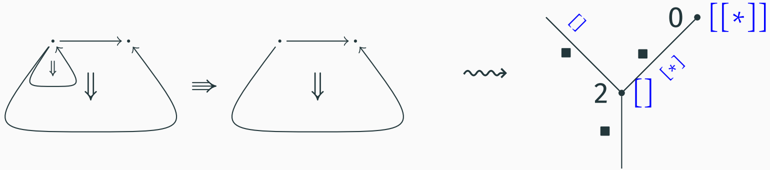
The proof tree of



is

$$\begin{array}{c}\vdots\\ 2\end{array}$$

The proof tree of

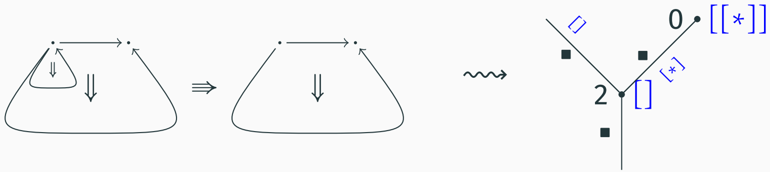


is

$$\frac{\begin{array}{c}\vdots\\ 2\end{array}}{\{[] \leftarrow 2}\ \mathtt{shift}$$

Examples

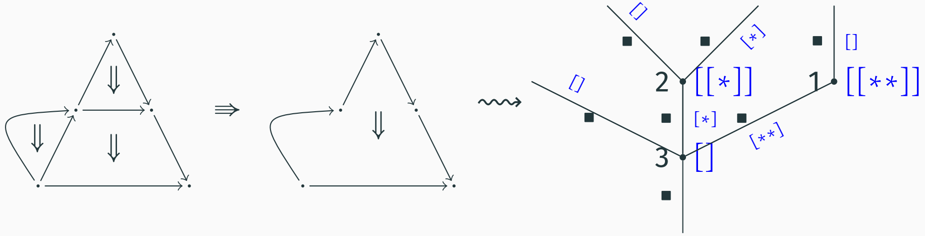
The proof tree of



is

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ 2\end{array}}{\left\{[] \leftarrow 2\right.}\,\texttt{shift} \qquad \begin{array}{c}\vdots\\ 0\end{array}}{\left\{\begin{array}{l}[] \leftarrow 2\\ [[*]] \leftarrow 0\end{array}\right.}\,\texttt{graft-}[[*]]$$

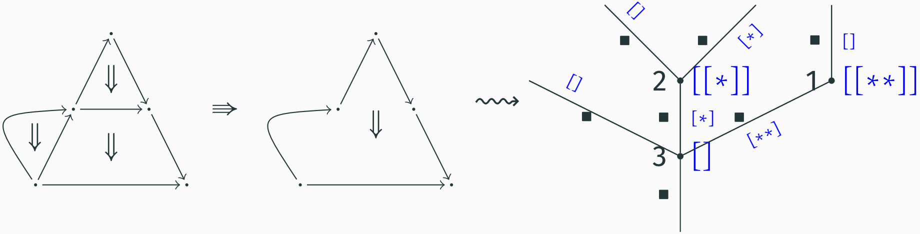
Examples



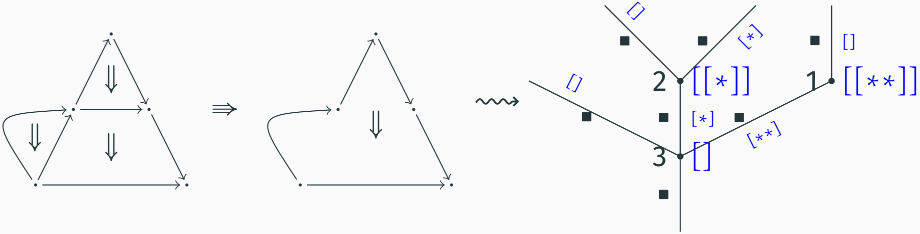
$\vdots$

3

Examples

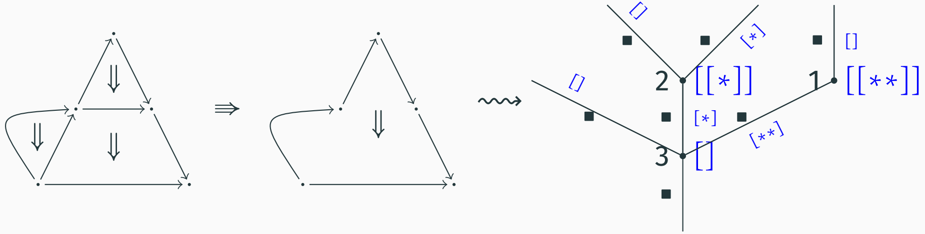


$$\frac{\begin{matrix}\vdots \\ 3\end{matrix}}{\{[] \leftarrow 3\}}\ \texttt{shift}$$



$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ 3\end{array}}{\{[] \leftarrow 3}\,\mathtt{shift} \qquad \begin{array}{c}\vdots\\ 2\end{array}}{\left\{\begin{array}{l}[] \leftarrow 3\\ [[*]] \leftarrow 2\end{array}\right.}\,\mathtt{graft\text{-}[[*]]}$$

Examples



$$\cfrac{\cfrac{\cfrac{\cfrac{\vdots}{3}}{\begin{cases}[] \leftarrow 3\end{cases}}\;\texttt{shift} \qquad \cfrac{\vdots}{2}}{\begin{cases}[] \leftarrow 3\\ [[*]] \leftarrow 2\end{cases}}\;\texttt{graft-[[*]]} \qquad \cfrac{\vdots}{1}}{\begin{cases}[] \leftarrow 3\\ [[*]] \leftarrow 2\\ [[**]] \leftarrow 1\end{cases}}\;\texttt{graft-[[**]]}$$

Conclusion

Conclusion

- In this presentation, we gave a “unnamed” way to decribe opetopes using terms and system $\mathrm{Opt}^?$.

Conclusion

- In this presentation, we gave a “unnamed” way to decribe opetopes using terms and system $\mathsf{Opt}^?$.
- In [Curien et al., 2019] `arXiv:1903.05848` we also present variants of this system for **opetopic sets**.

Conclusion

- In this presentation, we gave a “unnamed” way to decribe opetopes using terms and system $\mathsf{Opt}^?$.
- In [Curien et al., 2019] `arXiv:1903.05848` we also present variants of this system for **opetopic sets**.
- We are experimenting with those new tools to automatically check coherence laws for an appropriate definition of opetopic ∞-**groupoid**.

The various constructs and algorithms can be easily™ implemented, and opetopes amount to valid proof trees. An example implementation in Python 3 is available at `github.com/altaris/opetopy`, where valid proof trees are represented by certain expressions that evaluate without throwing any exception. For example:

$$\frac{\mathtt{p}}{\{[] \leftarrow \mathtt{p}}\ \mathtt{shift}$$

```
def shift(seq: Sequent) -> Sequent:
    n = seq.source.dimension
    ctx = Context(n + 1)
    for a in seq.source.nodeAddresses():
        ctx += (a.shift(), a)
    return Sequent(
        ctx,
        Preopetope.fromDictOfPreopetopes(
            {Address.epsilon(n):
            ↪   seq.source}
        ),
        seq.source
    )
```

Thank you for your attention!

Baez, J. C. and Dolan, J. (1998).
Higher-dimensional algebra. III. *n*-categories and the algebra of opetopes.
Advances in Mathematics, 135(2):145–206.

Cheng, E. (2003).
The category of opetopes and the category of opetopic sets.
Theory and Applications of Categories, 11:No. 16, 353–374.

Curien, P.-L., Ho Thanh, C., and Mimram, S. (2019).
Syntactic approaches for opetopes.
`arXiv:1903.05848 [math.CT]`.

References ii

Hermida, C., Makkai, M., and Power, J. (2002).
On weak higher-dimensional categories. I. 3.
Journal of Pure and Applied Algebra, 166(1-2):83–104.

Ho Thanh, C. (2018a).
The equivalence between opetopic sets and many-to-one polygraphs.
`arXiv:1806.08645 [math.CT]`.

Ho Thanh, C. (2018b).
opetopy.
`https://github.com/altaris/opetopy`.

Kock, J., Joyal, A., Batanin, M., and Mascari, J.-F. (2010).
Polynomial functors and opetopes.
Advances in Mathematics, 224(6):2690–2737.

Leinster, T. (2004).
Higher Operads, Higher Categories.
Cambridge University Press.