Type theoretical approaches to opetopes

Journées Logique Homotopie Catégories

Pierre-Louis Curien[1] Cédric Ho Thanh[2] Samuel Mimram[3]

October 18th, 2018

[1]IRIF, Paris Diderot University

[2]IRIF, Paris Diderot University; this author has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement №665850

[3]LIX, École Polytechnique

This presentation informally presents the main notions and results of [CHM18] (in preparation, draft available at `chothanh.wordpress.com`).

Contents

Opetopes

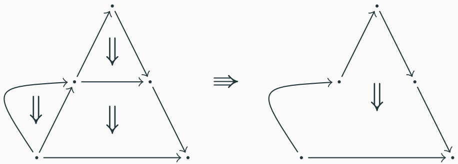
In a nutshell...

Opetopes are shapes (akin to globules, cubes, simplices, etc.) designed to represent the notion of composition in every dimension. As such, they were introduced in [BD98] to describe laws and coherence if weak higher categories.

In a nutshell...

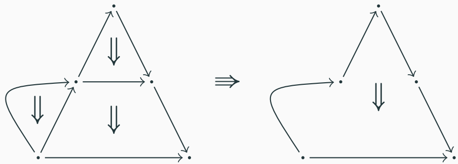
They are *pasting diagrams* where every cell is *many-to-one*.

Here is an example of a 3-opetope:



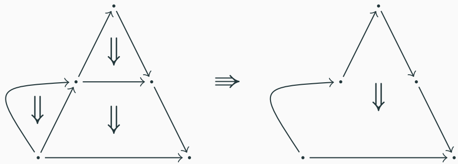
In a nutshell...

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



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

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

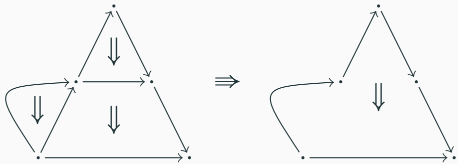


Every cell above has dimension 2, so that a 3-**opetope really is a pasting diagram of cells of dimension** 2.

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

In a nutshell...

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



Every cell 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 arrows).

We further ask those cells of dimension 1 **to be** 1**-opetopes**, i.e. pasting diagram (in a trivial way) of cells of dimension 0 (the points).

Informal definition

Definition
An n-opetope is a pasting diagram of $(n-1)$-opetopes

Informal definition

Definition
An n-opetope is a pasting diagram of $(n-1)$-opetopes i.e. a finite set of $(n-1)$-opetopes glued along $(n-2)$-opetopes.

Definition: low dimensions

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

$$\cdot$$

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*:

$$\cdot \longrightarrow \cdot$$

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, a.k.a. the arrow ∎

$$\begin{tikzcd} \cdot \arrow[r, bend left=80, ""{name=U}] \arrow[r, ""'{name=D}] & \cdot \arrow[from=U, to=D, phantom, "\Downarrow"] \end{tikzcd}$$

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, a.k.a. the arrow ■

$$\begin{tikzcd} & \cdot \arrow[dr] \arrow[d, phantom, "\Downarrow"] & \\ \cdot \arrow[ur] \arrow[rr] & {} & \cdot \end{tikzcd}$$

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, a.k.a. the arrow ∎

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

Definition: low dimensions

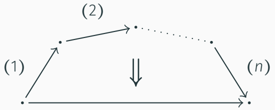
- There is a unique 0-dimensional opetope, which we'll call the *point*

$$\cdot$$

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

$$\cdot \longrightarrow \cdot$$

- 2-opetopes are pasting diagram of 1-opetopes, a.k.a. the arrow ∎



Definition: low dimensions

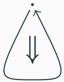
- There is a unique 0-dimensional opetope, which we'll call the *point*

$$\cdot$$

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

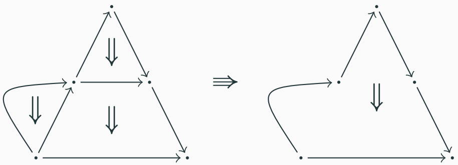
$$\cdot \longrightarrow \cdot$$

- 2-opetopes are pasting diagram of 1-opetopes, a.k.a. the arrow ∎



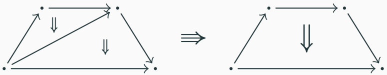
Definition: dimension 3

- 3-opetopes are pasting diagrams of 2-opetopes



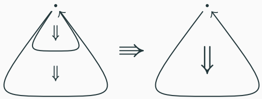
Definition: dimension 3

- 3-opetopes are pasting diagrams of 2-opetopes



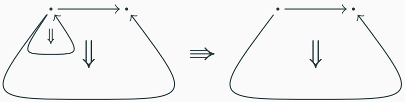
Definition: dimension 3

- 3-opetopes are pasting diagrams of 2-opetopes

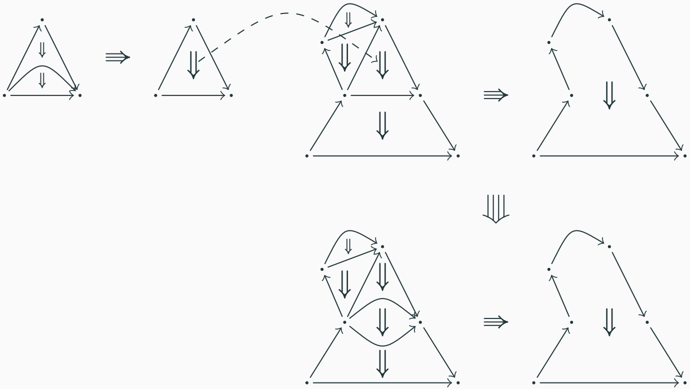


Definition: dimension 3

- 3-opetopes are pasting diagrams of 2-opetopes



- 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...

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 use T-operads [Lei04] or polynomial monads and trees [KJBM10], which are both unintuitive and difficult to manipulate.

Problem

1. The graphical approach is neither formal nor manageable for dimensions ≥ 4.
2. A formal definition either use T-operads [Lei04] or polynomial monads and trees [KJBM10], which are both unintuitive and difficult to manipulate.

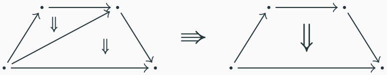
Solution

In this presentation, we give a **rough sketch** two ways to define opetopes syntactically.

The “named” approach

Idea

1. Take an opetope.



1. Take an opetope.

$$\begin{tikzcd}
& b \arrow[r, "g"] \arrow[dr, phantom, "\Downarrow\alpha", pos=0.3] & c \arrow[dr, "h"] & & & b \arrow[r, "g"] & c \arrow[dr, "h"] & \\
a \arrow[ur, "f"] \arrow[urr, "i"'] \arrow[rrr, "j"'] & & \arrow[u, phantom, "\Downarrow\beta"] & d \arrow[r, phantom, "\overset{A}{\Rrightarrow}"] & a \arrow[ur, "f"] \arrow[rrr, "j"'] & & \arrow[ul, phantom, "\Downarrow\gamma"] & d
\end{tikzcd}$$

2. Give names to everything.

Idea

1. Take an opetope.

$$
\begin{tikzcd}[ampersand replacement=\&] \& b \arrow[r, "g"] \arrow[d, phantom, "\Downarrow\alpha"] \& c \arrow[dr, "h"] \arrow[d, phantom, "\Downarrow\beta"] \& \\ a \arrow[ur, "f"] \arrow[urr, "i"] \arrow[rrr, "j"'] \& \& \& d \end{tikzcd}
\overset{A}{\Rrightarrow}
\begin{tikzcd}[ampersand replacement=\&] \& b \arrow[r, "g"] \arrow[dr, phantom, "\Downarrow\gamma"] \& c \arrow[dr, "h"] \& \\ a \arrow[ur, "f"] \arrow[rrr, "j"'] \& \& \& d \end{tikzcd}
$$

2. Give names to everything.
3. Write down the graftings:

$$A : \beta(i \leftarrow \alpha) \multimapdotbothB h(c \leftarrow g(b \leftarrow f)) \multimapdotbothB a \multimapdotbothB \emptyset.$$

4. ???
5. Profit!

Syntax

- We start with a set of variable $\mathbb{V} = \coprod_{n \in \mathbb{N}} \mathbb{V}_n$, where elements of $\mathbb{V}_n$ represent n-cells.

Syntax

- We start with a set of variable $\mathbb{V} = \coprod_{n \in \mathbb{N}} \mathbb{V}_n$, where elements of $\mathbb{V}_n$ represent n-cells.
- The set of n-terms is defined as

$$
\begin{array}{rcl}
\mathbb{T}_n & ::= & \mathbb{V}_n(\mathbb{V}_{n-1} \leftarrow \mathbb{T}_n, \cdots) \\
 & | & \underline{\mathbb{V}_{n-1}}
\end{array}
$$

Syntax

- We start with a set of variable $\mathbb{V} = \coprod_{n \in \mathbb{N}} \mathbb{V}_n$, where elements of $\mathbb{V}_n$ represent n-cells.
- The set of n-terms is defined as

$$\begin{array}{rcl} \mathbb{T}_n & ::= & \mathbb{V}_n(\mathbb{V}_{n-1} \leftarrow \mathbb{T}_n, \cdots) \\ & | & \underline{\mathbb{V}_{n-1}} \end{array}$$

Examples

For $a, b, c \in \mathbb{V}_0$, $f, g, h \in \mathbb{V}_1$,

$$a \in \mathbb{T}_0, \qquad h(a \leftarrow g, b \leftarrow f) \in \mathbb{T}_1,$$

$$f(a \leftarrow f(a \leftarrow f), a \leftarrow f, a \leftarrow f) \in \mathbb{T}_1, \qquad \underline{h} \in \mathbb{T}_2.$$

Syntax

- An n-type is a sequence of terms of the form

$$s_1 \multimap s_2 \multimap \cdots \multimap s_n \multimap s_{n+1} \multimap \varnothing,$$

  where $s_i \in \mathbb{T}_{n+1-i}$.

Syntax

- An n-type is a sequence of terms of the form

$$s_1 \multimap s_2 \multimap \cdots \multimap s_n \multimap s_{n+1} \multimap \varnothing,$$

  where $s_i \in \mathbb{T}_{n+1-i}$.
- A n-typing is an expression of the form

$$t : T$$

  where $t \in \mathbb{T}_n$ and T is an $(n-1)$-type.

Main result of the named approach

Theorem
Derivable typings in system $\mathsf{Opt}^{!}$ of the form

$$\alpha : T$$

where $\alpha \in \mathbb{V}_n$ (as opposed to just $\mathbb{T}_n$) are in bijective correspondence (up to renaming of variables) with n-opetopes.

System $\mathtt{Opt}^!$: the `point` rule

The first rule of $\mathtt{Opt}^!$ states that we may create points without any prior assumption:

$$\frac{}{\cdot}\ \mathtt{point}$$

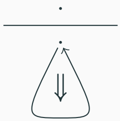
System $\mathsf{Opt}^!$: the `point` rule

The first rule of $\mathsf{Opt}^!$ states that we may create points without any prior assumption:

$$\frac{}{\cdot}\ \mathtt{point} \qquad\qquad \frac{}{X : \varnothing}\ \mathtt{point}$$

System Opt$^!$: the `degen-fill` rule

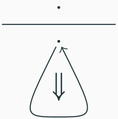
This rule takes an opetope and produces a degenerate opetope from it:



`degen-fill`

System Opt$^!$: the `degen-fill` rule

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



`degen-fill`

$$\frac{x : T}{\delta : \underline{x} \bullet\!\!-\!\!\circ\, x \bullet\!\!-\!\!\circ\, T}\ \texttt{degen-fill}$$

System Opt$^!$: the `fill` rule

This rule takes a pasting diagram (that is, a term), and creates an opetope by “filling” it:

$$
\frac{\cdot \xrightarrow{f} \cdot \xrightarrow{g} \cdot \xrightarrow{h} \cdot \xrightarrow{i} \cdot}{
\begin{tikzcd}
 & \cdot \arrow[r, "g"] & \cdot \arrow[r, "h"] \arrow[d, phantom, "\Downarrow \mu"] & \cdot \arrow[dr, "i"] & \\
\cdot \arrow[ur, "f"] \arrow[rrrr] & & {} & & \cdot
\end{tikzcd}}
\ \texttt{fill}
$$

System Opt$^!$: the `fill` rule

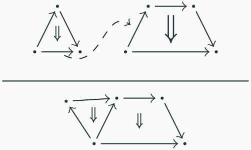
This rule takes a pasting diagram (that is, a term), and creates an opetope by “filling” it:

$$
\cfrac{\cdot \xrightarrow{f} \cdot \xrightarrow{g} \cdot \xrightarrow{h} \cdot \xrightarrow{i} \cdot}{\begin{tikzcd} & \cdot \arrow[r, "g"] & \cdot \arrow[r, "h"] \arrow[d, phantom, "\Downarrow \mu"] & \cdot \arrow[dr, "i"] & \\ \cdot \arrow[ur, "f"] \arrow[rrrr] & & & & \cdot \end{tikzcd}} \ \texttt{fill}
$$

$$
\cfrac{t : T}{\mu : t \multimapboth T} \ \texttt{fill}
$$

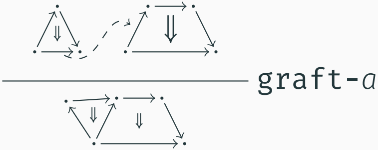
System Opt$^!$: the `graft` rule

This rules glues an opetope to a pasting diagram of the same dimension:



`graft`-a

This rules glues an opetope to a pasting diagram of the same dimension:



$$\frac{t : s \mathrel{\bullet\!\!-\!\!\circ} T \qquad x : y \mathrel{\bullet\!\!-\!\!\circ} U}{t(a \leftarrow x) : s[y/a] \mathrel{\bullet\!\!-\!\!\circ} T}\ \texttt{graft-}a$$

Example 1

Let’s derive

$$
\begin{tikzcd}
 & b \arrow[r, "g"] \arrow[d, phantom, "\Downarrow\alpha"] & c \arrow[dr, "h"] \arrow[d, phantom, "\Downarrow\beta"] & \\
a \arrow[ur, "f"] \arrow[urr, "i"'] \arrow[rrr, "j"'] & & & d
\end{tikzcd}
\quad \overset{A}{\Rrightarrow} \quad
\begin{tikzcd}
 & b \arrow[r, "g"] \arrow[d, phantom, "\Downarrow\gamma"] & c \arrow[dr, "h"] & \\
a \arrow[ur, "f"] \arrow[rrr, "j"'] & & & d
\end{tikzcd}
$$

Example 1

Let's derive

$$\begin{tikzcd} & b \arrow[r, "g"] \arrow[d, phantom, "\Downarrow\alpha"] & c \arrow[dr, "h"] \arrow[d, phantom, "\Downarrow\beta"] & \\ a \arrow[ur, "f"] \arrow[urr, "i"'] \arrow[rrr, "j"'] & & & d \end{tikzcd} \quad \overset{A}{\Rrightarrow} \quad \begin{tikzcd} & b \arrow[r, "g"] & c \arrow[dr, "h"] & \\ a \arrow[ur, "f"] \arrow[rrr, "j"'] \arrow[urr, phantom, "\Downarrow\gamma"] & & & d \end{tikzcd}$$

Derivation of α

$$\cfrac{\cfrac{\cfrac{\cfrac{}{b : \varnothing}\ \texttt{point}}{g : b \multimap \varnothing}\ \texttt{fill} \qquad \cfrac{\cfrac{}{a : \varnothing}\ \texttt{point}}{f : a \multimap \varnothing}\ \texttt{fill}}{g(b \leftarrow f) : \underbrace{b[a/b]}_{\equiv a} \multimap \varnothing}\ \texttt{graft-}b}{\alpha : g(b \leftarrow f) \multimap a \multimap \varnothing}\ \texttt{fill}$$

.

Let's derive

$$\begin{tikzcd}
 & b \arrow[r, "g"] \arrow[d, phantom, "\Downarrow\alpha"] & c \arrow[dr, "h"] & \\
a \arrow[ur, "f"] \arrow[urr, "i"'] \arrow[rrr, "j"'] & & \arrow[r, phantom, "\Downarrow\beta"] & d
\end{tikzcd}
\overset{A}{\Rrightarrow}
\begin{tikzcd}
 & b \arrow[r, "g"] \arrow[d, phantom, "\Downarrow\gamma"] & c \arrow[dr, "h"] & \\
a \arrow[ur, "f"] \arrow[rrr, "j"'] & & & d
\end{tikzcd}$$

Derivation of β

$$\dfrac{\dfrac{\dfrac{\dfrac{}{c : \varnothing}\,\texttt{point}}{h : c \bullet\!\!-\!\!\circ\, \varnothing}\,\texttt{fill} \qquad \dfrac{\dfrac{}{a : \varnothing}\,\texttt{point}}{i : a \bullet\!\!-\!\!\circ\, \varnothing}\,\texttt{fill}}{h(c \leftarrow i) : \underbrace{c[a/c]}_{\equiv a} \bullet\!\!-\!\!\circ\, \varnothing}\,\texttt{graft-}c}{\beta : h(c \leftarrow i) \bullet\!\!-\!\!\circ\, a \bullet\!\!-\!\!\circ\, \varnothing}\,\texttt{fill}$$

Example 1

Let's derive

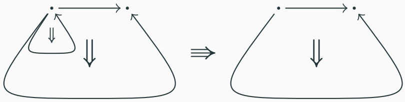
$$
\begin{tikzcd}
 & b \arrow[r, "g"] \arrow[d, phantom, "\Downarrow\alpha"] & c \arrow[dr, "h"] \arrow[d, phantom, "\Downarrow\beta"] & \\
a \arrow[ur, "f"] \arrow[urr, "i"] \arrow[rrr, "j"'] & & & d
\end{tikzcd}
\quad \overset{A}{\Rrightarrow} \quad
\begin{tikzcd}
 & b \arrow[r, "g"] \arrow[dr, phantom, "\Downarrow\gamma"] & c \arrow[dr, "h"] & \\
a \arrow[ur, "f"] \arrow[rrr, "j"'] & & & d
\end{tikzcd}
$$

And we assemble to get A

$$
\dfrac{\dfrac{\begin{matrix}\vdots \\ \beta : h(c \leftarrow i) \multimap a \multimap \varnothing\end{matrix} \qquad \begin{matrix}\vdots \\ \alpha : g(b \leftarrow f) \multimap a \multimap \varnothing\end{matrix}}{\beta(i \leftarrow \alpha) : \underbrace{h(c \leftarrow i)[g(b \leftarrow f)/i]}_{\equiv h(c \leftarrow g(b \leftarrow f))} \multimap a \multimap \varnothing}\;\texttt{graft-}i}{A : \beta(i \leftarrow \alpha) \multimap h(c \leftarrow g(b \leftarrow f)) \multimap a \multimap \varnothing}\;\texttt{fill}
$$

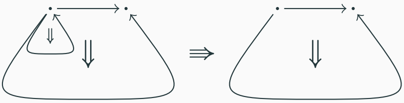
Example 2

Let's derive



Example 2

Let’s derive

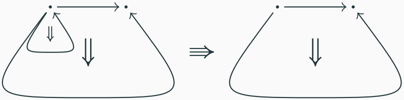


Top left part

$$\dfrac{\dfrac{}{a : \varnothing}\;\texttt{point}}{\alpha : \underline{a} \bullet\!\!-\!\!\circ\; a \bullet\!\!-\!\!\circ\; \varnothing}\;\texttt{degen-fill}$$

Example 2

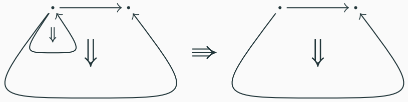
Let's derive



Bottom part

$$\dfrac{\dfrac{\dfrac{}{b : \varnothing}\,\texttt{point}}{g : b \bullet\!\!-\!\!\circ\, \varnothing}\,\texttt{fill} \qquad \dfrac{\dfrac{}{a : \varnothing}\,\texttt{point}}{f : a \bullet\!\!-\!\!\circ\, \varnothing}\,\texttt{fill}}{\dfrac{g(b \leftarrow f) : a \bullet\!\!-\!\!\circ\, \varnothing}{\beta : g(b \leftarrow f) \bullet\!\!-\!\!\circ\, a \bullet\!\!-\!\!\circ\, \varnothing}\,\texttt{fill}}\,\texttt{graft-}b$$

Let's derive



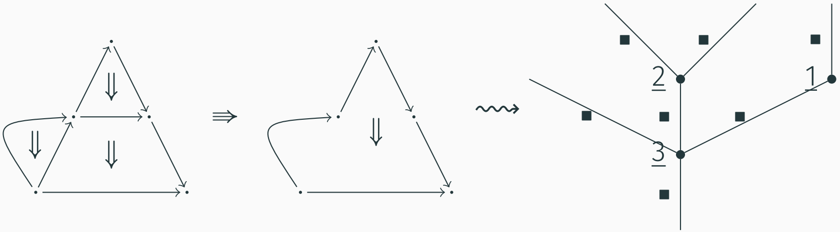
And we assemble

$$
\dfrac{\dfrac{\begin{matrix}\vdots \\ \beta : g(b \leftarrow f) \bullet\!\!-\!\!\circ\ a \bullet\!\!-\!\!\circ\ \varnothing\end{matrix} \qquad \begin{matrix}\vdots \\ \alpha : \underline{a} \bullet\!\!-\!\!\circ\ a \bullet\!\!-\!\!\circ\ \varnothing\end{matrix}}{a = b \vdash \beta(f \leftarrow \alpha) : \underbrace{g(b \leftarrow f)[\underline{a}/f]}_{\equiv g} \bullet\!\!-\!\!\circ\ a \bullet\!\!-\!\!\circ\ \varnothing}\ \texttt{graft-}f}{a = b \vdash A : \beta(f \leftarrow \alpha) \bullet\!\!-\!\!\circ\ g \bullet\!\!-\!\!\circ\ a \bullet\!\!-\!\!\circ\ \varnothing}\ \texttt{fill}
$$

The “unnamed” approach

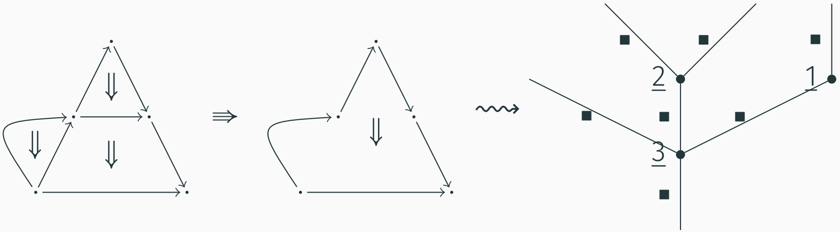
Idea

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



Idea

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



Then a cell in a pasting diagram no longer needs to have a name, it can be identified by its *address* in that tree.

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

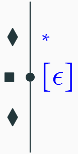
Denote by ♦ the unique 0-opetope, a.k.a. the point:

$$\cdot$$

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:

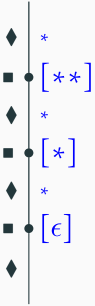


Let us add address information.

Idea: dimension 2

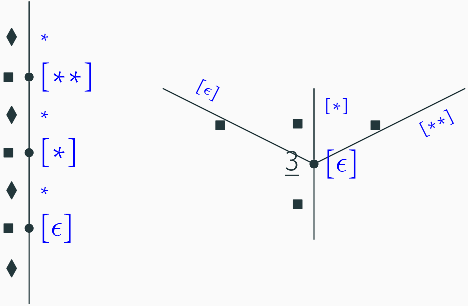
Then we can:

1. create a tree with that node representing ■



Then we can:

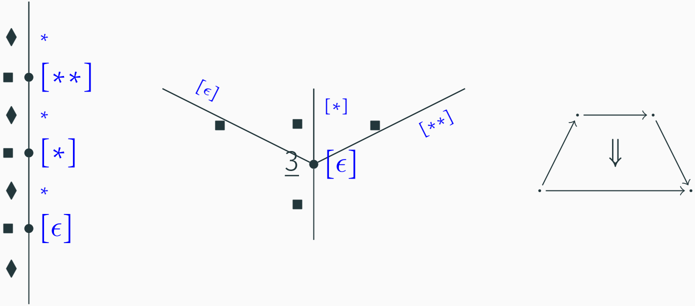
1. create a tree with that node representing ■



2. consider that tree like a node, where the input edges are the nodes of said tree

Then we can:

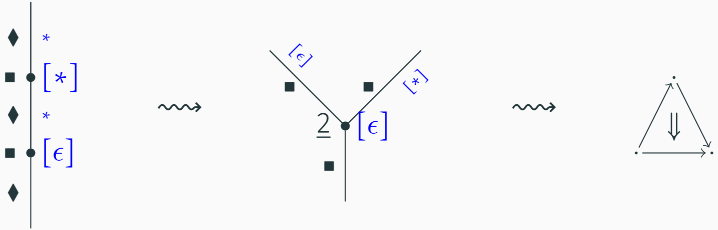
1. create a tree with that node representing ■



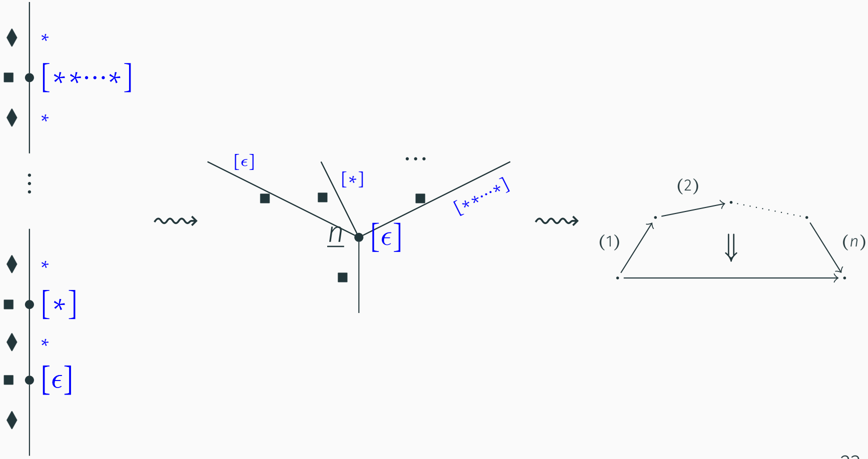
2. consider that tree like a node, where the input edges are the nodes of said tree
3. be convinced that this is a good representation of some 2-opetope!

Idea: dimension 2

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

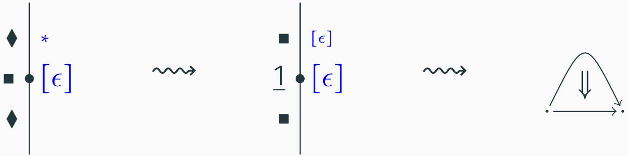


Depending on the original tree, we obtain different 2-opetopes:



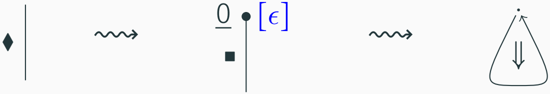
Idea: dimension 2

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

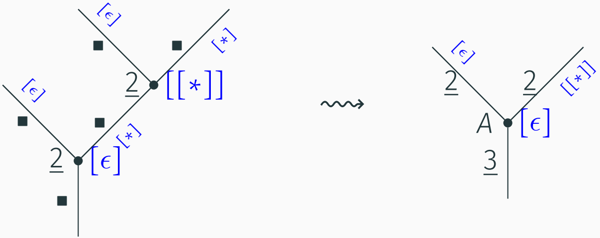


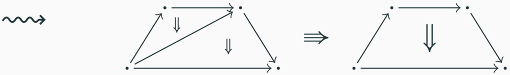
Idea: dimension 2

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

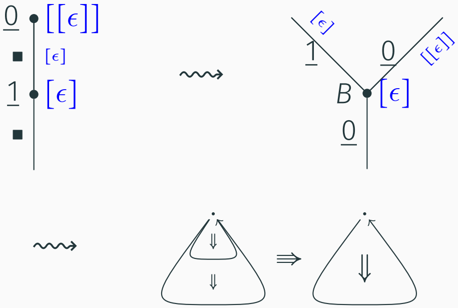


From there, repeat the process!



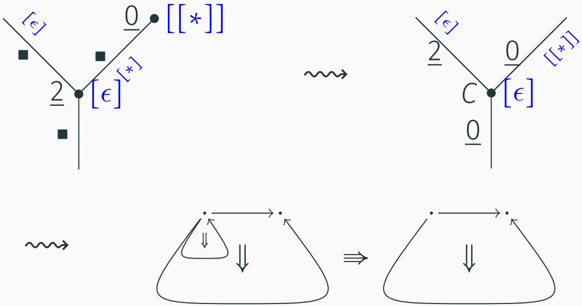


From there, repeat the process!



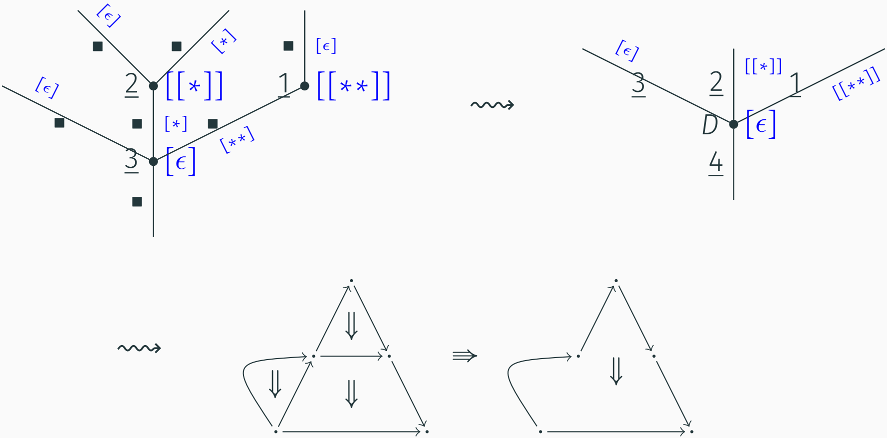
Idea: dimension 3

From there, repeat the process!



Idea: dimension 3

From there, repeat the process!

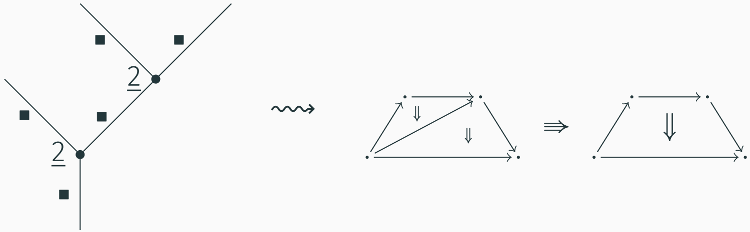


Syntax

We now want a syntactical description of such trees.

We now want a syntactical description of such trees.

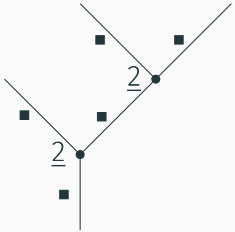
Solution



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

We now want a syntactical description of such trees.

Solution



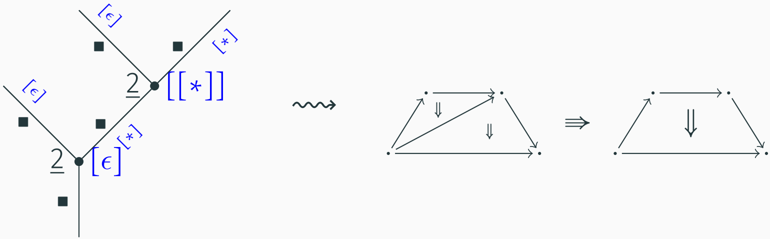
$\rightsquigarrow$

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

Syntax

We now want a syntactical description of such trees.

Solution

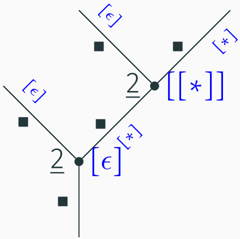


In an n-opetope, every node is decorated by $(n-1)$-opetope, but $(n-1)$-opetope does not uniquely identify a node. But addresses do! So we just need to describe a partial map

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

Syntax

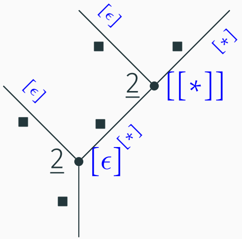
We encode opetopes recursively as follows:



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

Syntax

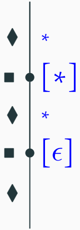
We encode opetopes recursively as follows:



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

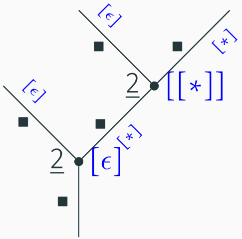
Reminder

$\underline{2} \quad =$



Syntax

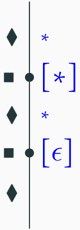
We encode opetopes recursively as follows:



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

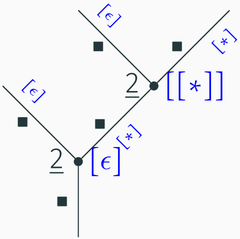
Reminder

$$\underline{2} \quad =$$



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

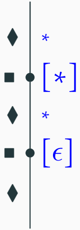
We encode opetopes recursively as follows:



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

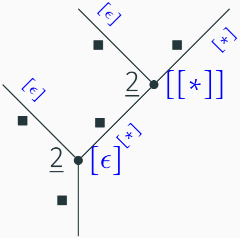
Reminder

$\underline{2}$ =



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

We encode opetopes recursively as follows:

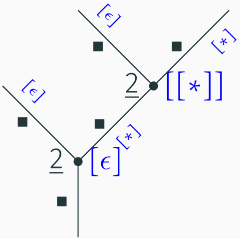


$$\rightsquigarrow \quad \begin{cases} [\epsilon] \leftarrow \begin{cases} [\epsilon] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases} \\ [[\epsilon]] \leftarrow \begin{cases} [\epsilon] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases} \end{cases}$$

Convention

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

We encode opetopes recursively as follows:

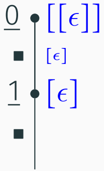


$$\rightsquigarrow \quad \begin{cases} [\epsilon] \leftarrow \begin{cases} [\epsilon] \leftarrow \begin{cases} * \leftarrow \blacklozenge \end{cases} \\ [*] \leftarrow \begin{cases} * \leftarrow \blacklozenge \end{cases} \end{cases} \\ [[\epsilon]] \leftarrow \begin{cases} [\epsilon] \leftarrow \begin{cases}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

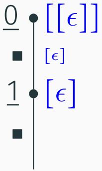
Convention

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

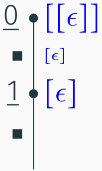
Syntax: examples



Syntax: examples



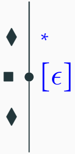
$$\rightsquigarrow \quad \begin{cases} [\epsilon] \leftarrow \underline{1} \\ [[\epsilon]] \leftarrow \underline{0} \end{cases}$$



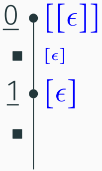
$$\rightsquigarrow \begin{cases}[\epsilon] \leftarrow \underline{1} \\ [[\epsilon]] \leftarrow \underline{0}\end{cases}$$

Reminder

$$\underline{1} =$$



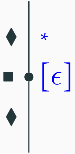
$$= \left\{ [\epsilon] \leftarrow \blacksquare \right.$$



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

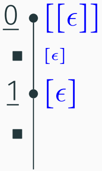
Reminder

$$\underline{1} \quad =$$



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

Syntax: examples

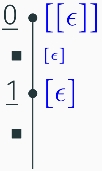


$$\rightsquigarrow \quad \begin{cases} [\epsilon] \leftarrow \begin{cases} [\epsilon] \leftarrow \blacksquare \end{cases} \\ [[\epsilon]] \leftarrow \underline{0} \end{cases}$$

Reminder

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

Syntax: examples

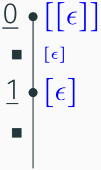


$$\rightsquigarrow \quad \begin{cases} [\epsilon] \leftarrow \begin{cases} [\epsilon] \leftarrow \begin{cases} * \leftarrow \blacklozenge \end{cases} \end{cases} \\ [[\epsilon]] \leftarrow \underline{0} \end{cases}$$

Reminder

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

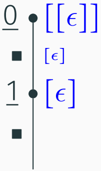
Syntax: examples



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

Reminder + convention

$\underline{0} \quad = $  $= \quad \left\{\!\!\left\{ \blacklozenge \right.\right.$

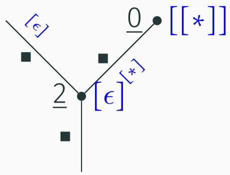


$$\rightsquigarrow \quad \begin{cases} [\epsilon] \leftarrow \left\{ [\epsilon] \leftarrow \left\{ * \leftarrow \blacklozenge \right.\right. \\ [[\epsilon]] \leftarrow \{\!\{ \blacklozenge \end{cases}$$

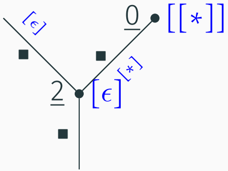
Reminder + convention

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

Syntax: examples

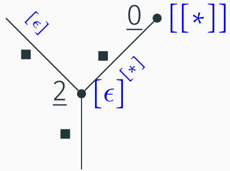


Syntax: examples



$$\rightsquigarrow \quad \begin{cases} [\epsilon] \leftarrow \underline{2} \\ [[\epsilon]] \leftarrow \underline{0} \end{cases}$$

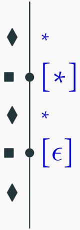
Syntax: examples



$$\rightsquigarrow \quad \begin{cases} [\epsilon] \leftarrow \underline{2} \\ [[\epsilon]] \leftarrow \underline{0} \end{cases}$$

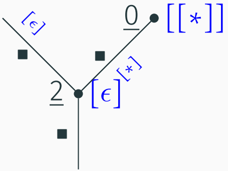
Reminder

$\underline{2} \quad = $



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

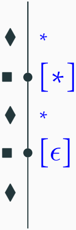
Syntax: examples



$$\rightsquigarrow \quad \begin{cases} [\epsilon] \leftarrow \begin{cases} [\epsilon] \leftarrow \blacksquare \\ [\ast] \leftarrow \blacksquare \end{cases} \\ [[\epsilon]] \leftarrow \underline{0} \end{cases}$$

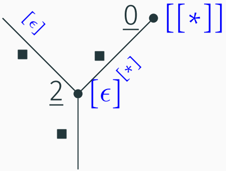
Reminder

$$\underline{2} \quad = $$



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

Syntax: examples

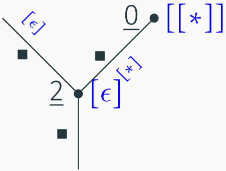


$$\rightsquigarrow \quad \begin{cases} [\epsilon] \leftarrow \begin{cases} [\epsilon] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \end{cases} \\ [[\epsilon]] \leftarrow \underline{0} \end{cases}$$

Reminder

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

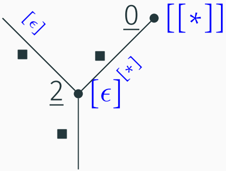
Syntax: examples



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

Reminder

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



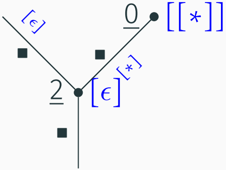
$$\rightsquigarrow \quad \begin{cases} [\epsilon] \leftarrow \begin{cases} [\epsilon] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \\ [*] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \end{cases} \\ [[\epsilon]] \leftarrow \underline{0} \end{cases}$$

Reminder

$\underline{0} \quad = $

$= \quad \left\{\left\{ \blacklozenge \right.\right.$

Syntax: examples

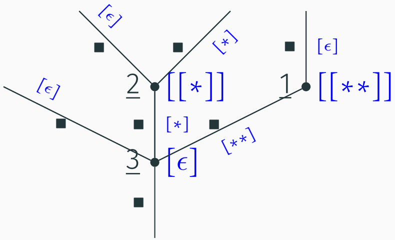


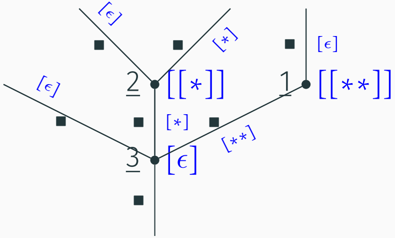
$$\rightsquigarrow \begin{cases}[\epsilon] \leftarrow \begin{cases}[\epsilon] \leftarrow \{ * \leftarrow \blacklozenge \\ [*] \leftarrow \{ * \leftarrow \blacklozenge\end{cases} \\ [[\epsilon]] \leftarrow \{\{\blacklozenge\end{cases}$$

Reminder

$\underline{0} \quad = $  $= \quad \{\{\blacklozenge$

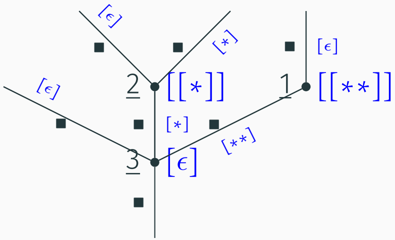
Syntax: examples





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

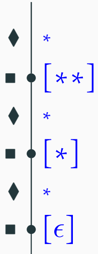
Syntax: examples



$$\rightsquigarrow \begin{cases} [\epsilon] \leftarrow \underline{3} \\ [[\epsilon]] \leftarrow \underline{1} \\ [[*]] \leftarrow \underline{2} \end{cases}$$

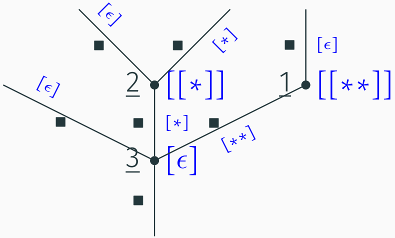
Reminder

$\underline{3}$ =



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

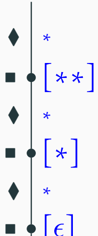
Syntax: examples



$$\rightsquigarrow \quad \begin{cases} [\epsilon] \leftarrow \begin{cases} [\epsilon] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare \end{cases} \\ [[\epsilon]] \leftarrow \underline{1} \\ [[*]] \leftarrow \underline{2} \end{cases}$$

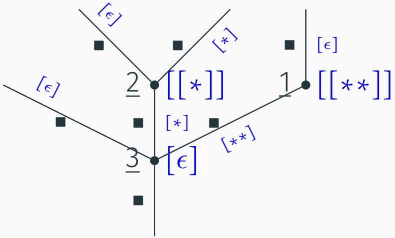
Reminder

$$\underline{3} \quad =$$



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

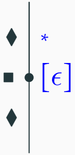
Syntax: examples



$$\rightsquigarrow \quad \begin{cases} [\epsilon] \leftarrow \begin{cases} [\epsilon] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare \end{cases} \\ [[\epsilon]] \leftarrow \underline{1} \\ [[*]] \leftarrow \underline{2} \end{cases}$$

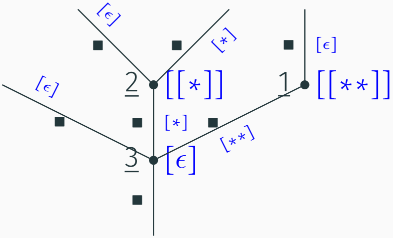
Reminder

$\underline{1}$ =



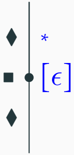
$$= \quad \Big\{ [\epsilon] \leftarrow \blacksquare$$

Syntax: examples

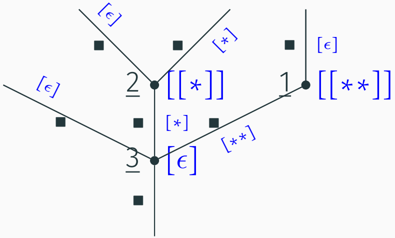


$$\rightsquigarrow \quad \begin{cases} [\epsilon] \leftarrow \begin{cases} [\epsilon] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare \end{cases} \\ [[\epsilon]] \leftarrow \left\{ [\epsilon] \leftarrow \blacksquare \right. \\ [[*]] \leftarrow \underline{2} \end{cases}$$

Reminder

$\underline{1} \quad =$  $= \quad \left\{ [\epsilon] \leftarrow \blacksquare \right.$

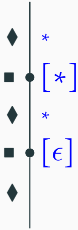
Syntax: examples



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

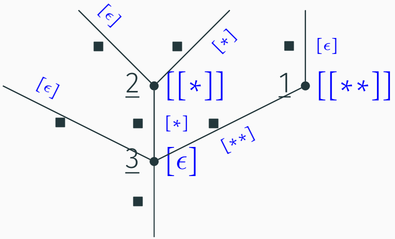
Reminder

$$\underline{2} \quad = $$



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

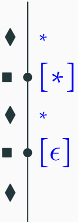
Syntax: examples



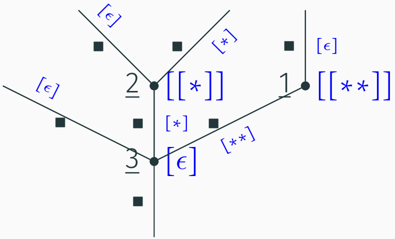
$$\rightsquigarrow \begin{cases} [\epsilon] \leftarrow \begin{cases} [\epsilon] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare \end{cases} \\ [[\epsilon]] \leftarrow \begin{cases} [\epsilon] \leftarrow \blacksquare \end{cases} \\ [[*]] \leftarrow \begin{cases}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

Reminder

$\underline{2} \quad =$



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

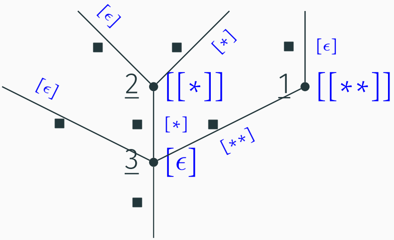


$$\rightsquigarrow \left\{ \begin{array}{l} [\epsilon] \leftarrow \left\{ \begin{array}{l} [\epsilon] \leftarrow \blacksquare \\ [*] \leftarrow \blacksquare \\ [**] \leftarrow \blacksquare \end{array} \right. \\ [[\epsilon]] \leftarrow \left\{ [\epsilon] \leftarrow \blacksquare \right. \\ [[*]] \leftarrow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

Reminder

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

Syntax: examples



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

Reminder

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

Question
Is this an opetope?

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

System Opt$^?$

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 $\mathsf{Opt}^?$

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

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

The $\mathsf{Opt}^?$ system aims to characterize preopetopes that actually are opetopes:

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

System $\mathtt{Opt}^?$: the `point` rule

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

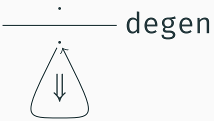
$$\frac{}{\cdot}\ \mathtt{point}$$

System Opt$^?$: the `point` rule

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

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

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



System Opt$^?$: the degen rule

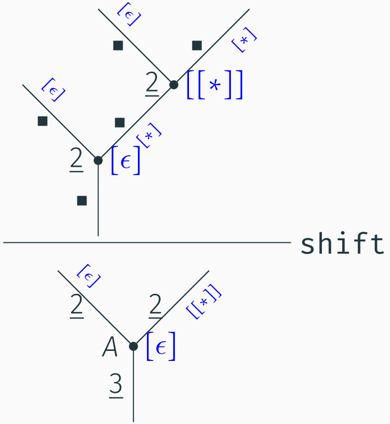
This rule takes an opetope and produces a degenerate opetope from it:

$$\frac{\cdot}{}\ \mathsf{degen}$$

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

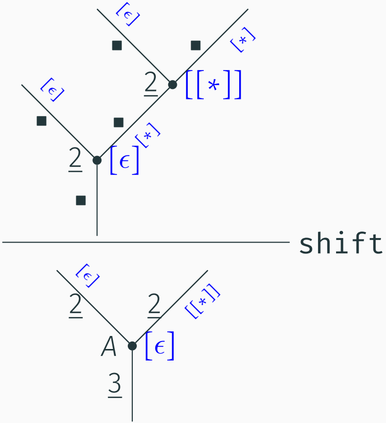
System Opt$^?$: the `shift` rule

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



System Opt?: the `shift` rule

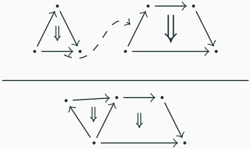
This rule takes an opetope p and produces a new opetope having a unique node, decorated in p:



$$\frac{\mathsf{p}}{\left\{[\epsilon] \leftarrow \mathsf{p}\right.}\ \mathtt{shift}$$

System Opt$^?$: the `graft` rule

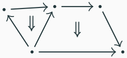
This rule glues an n-opetope q to an $(n+1)$-opetope p, the latter really just being a pasting diagram of n-opetopes:



`graft`-$[b]$

System 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:



$\mathtt{graft}\text{-}[b]$

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

(we omitted some technical assumptions that ensure this operation is geometrically meaningful)

Example

The proof tree of

```
.
```

is:

$$\dfrac{}{\blacklozenge}\ \texttt{point}$$

Example

The proof tree of

$$\cdot \longrightarrow \cdot$$

is:

$$\dfrac{\dfrac{}{\blacklozenge}\,\texttt{point}}{\left\{[\epsilon] \leftarrow \blacklozenge\right.}\,\texttt{shift}$$

Example

The proof tree of

$$\cdot \longrightarrow \cdot$$

is:

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

Example

The proof tree of

$$\begin{tikzcd} \cdot \arrow[r, bend left=60] \arrow[r] \arrow[r, phantom, bend left=30, "\Downarrow"] & \cdot \end{tikzcd}$$

is:

$$\dfrac{\dfrac{\dfrac{}{\blacklozenge}\ \mathtt{point}}{\{* \leftarrow \blacklozenge}\ \mathtt{shift}}{\{[\epsilon] \leftarrow \{* \leftarrow \blacklozenge}\ \mathtt{shift}$$

Example

The proof tree of

$$\begin{tikzcd}
 & \cdot \arrow[dr] \arrow[d, phantom, "\Downarrow"] & \\
\cdot \arrow[ur] \arrow[rr] & & \cdot
\end{tikzcd}$$

is:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\blacklozenge}\;\texttt{point}}{\{* \leftarrow \blacklozenge}\;\texttt{shift}}{\{[\epsilon] \leftarrow \{* \leftarrow \blacklozenge}\;\texttt{shift} \qquad \cdot \longrightarrow \cdot}{?}\;\texttt{graft-}[*]$$

Example

The proof tree of

$$\begin{tikzcd}[column sep=small] & \cdot \arrow[dr] \arrow[d, phantom, "\Downarrow"] & \\ \cdot \arrow[ur] \arrow[rr] & & \cdot \end{tikzcd}$$

is:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\blacklozenge}\,\texttt{point}}{\{* \leftarrow \blacklozenge}\,\texttt{shift}}{\{[\epsilon] \leftarrow \{* \leftarrow \blacklozenge}\,\texttt{shift} \qquad \dfrac{\dfrac{}{\blacklozenge}\,\texttt{point}}{\{* \leftarrow \blacklozenge}\,\texttt{shift}}{?}\,\texttt{graft-}[*]$$

The proof tree of

$$
\begin{tikzcd}
& \cdot \arrow[dr] \arrow[d, phantom, "\Downarrow"] & \\
\cdot \arrow[ur] \arrow[rr] & & \cdot
\end{tikzcd}
$$

is:

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

The proof tree of

$$\begin{tikzcd}
& \cdot \arrow[r] \arrow[d, phantom, "\Downarrow" near end, xshift=2.5em] & \cdot \arrow[dr] & \\
\cdot \arrow[ur] \arrow[rrr] & & & \cdot
\end{tikzcd}$$

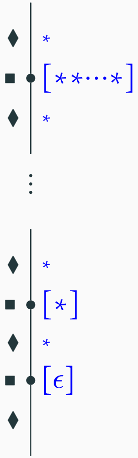
is:

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

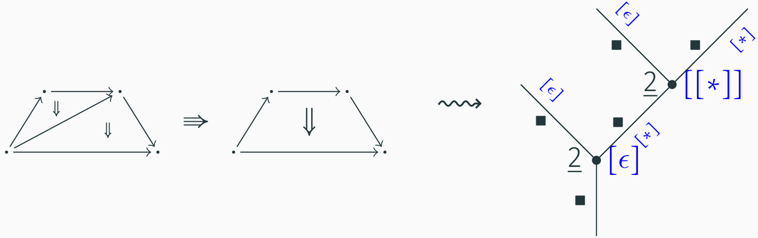
Examples

Write

$$\underline{n} = \begin{cases} [\epsilon] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \\ [*] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \\ [**] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \\ \vdots \\ [*^{n-1}] \leftarrow \left\{ * \leftarrow \blacklozenge \right. \end{cases} =$$



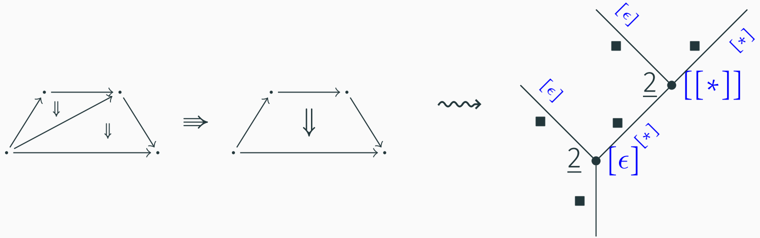
The proof tree of



is:

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

The proof tree of

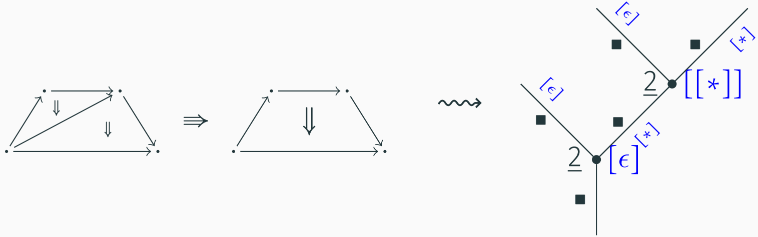


is:

$$\dfrac{\begin{array}{c}\vdots\\ \underline{2}\end{array}}{\left\{[\epsilon] \leftarrow \underline{2}\right.}\ \texttt{shift}$$

Examples

The proof tree of

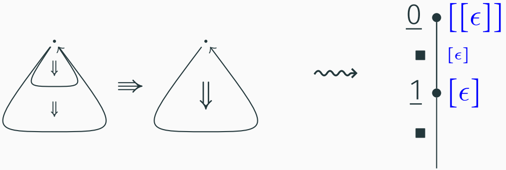


is:

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \underline{2}\end{matrix}}{\left\{[\epsilon]\leftarrow\underline{2}\right.}\,\texttt{shift}\qquad\begin{matrix}\vdots\\ \underline{2}\end{matrix}}{\begin{cases}[\epsilon]\leftarrow\underline{2}\\ [[*]]\leftarrow\underline{2}\end{cases}}\,\texttt{graft-}[[*]]$$

Example

The proof tree of

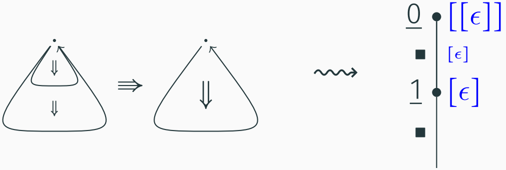


is

$$\begin{array}{c}\vdots \\ \underline{1}\end{array}$$

Example

The proof tree of

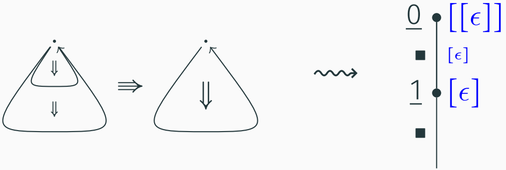


is

$$\frac{\begin{array}{c}\vdots \\ \underline{1}\end{array}}{\left\{[\epsilon] \leftarrow \underline{1}\right.}\ \texttt{shift}$$

Example

The proof tree of

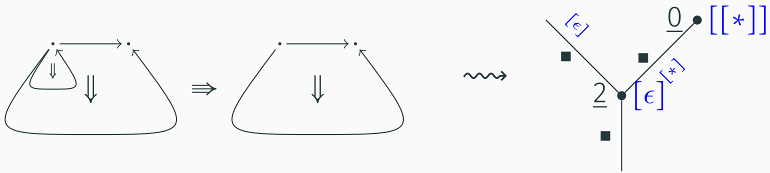


is

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \underline{1}\end{matrix}}{\left\{[\epsilon] \leftarrow \underline{1}\right.}\,\texttt{shift} \qquad \begin{matrix}\vdots\\ \underline{0}\end{matrix}}{\begin{cases}[\epsilon]\leftarrow\underline{1}\\ [[\epsilon]]\leftarrow\underline{0}\end{cases}}\,\texttt{graft-}[[\epsilon]]$$

Example

The proof tree of

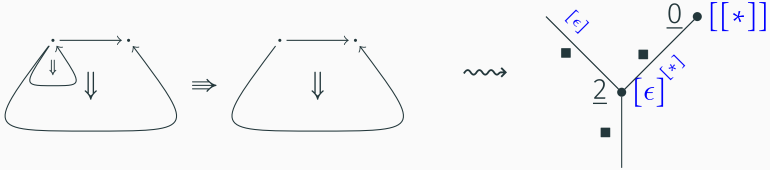


is

$$\vdots$$
$$\underline{2}$$

Example

The proof tree of

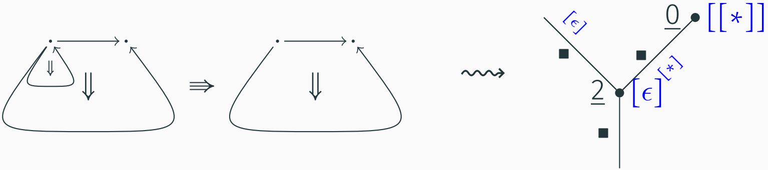


is

$$\dfrac{\begin{array}{c}\vdots\\ \underline{2}\end{array}}{\left\{[\epsilon] \leftarrow \underline{2}\right.}\ \texttt{shift}$$

Example

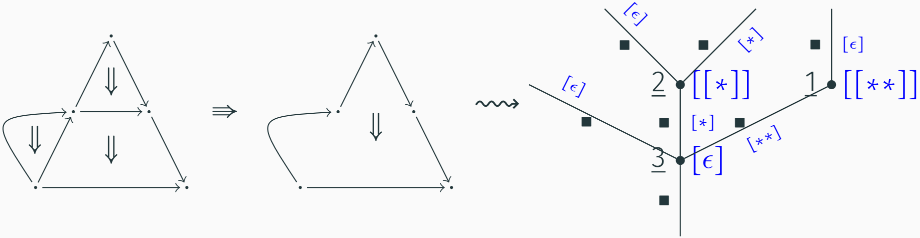
The proof tree of



is

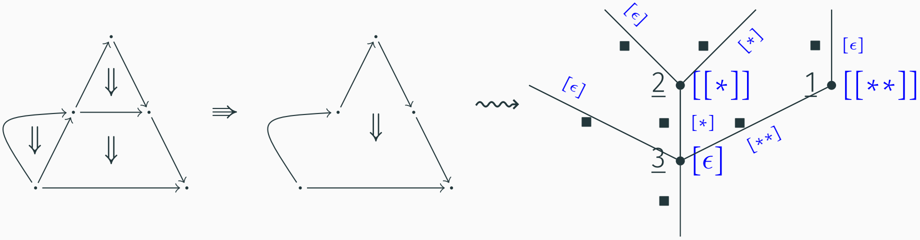
$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \underline{2}\end{array}}{\left\{[\epsilon] \leftarrow \underline{2}\right.}\;\texttt{shift} \qquad \begin{array}{c}\vdots\\ \underline{0}\end{array}}{\begin{cases}[\epsilon] \leftarrow \underline{2}\\ [[*]] \leftarrow \underline{0}\end{cases}}\;\texttt{graft-}[[*]]$$

Example



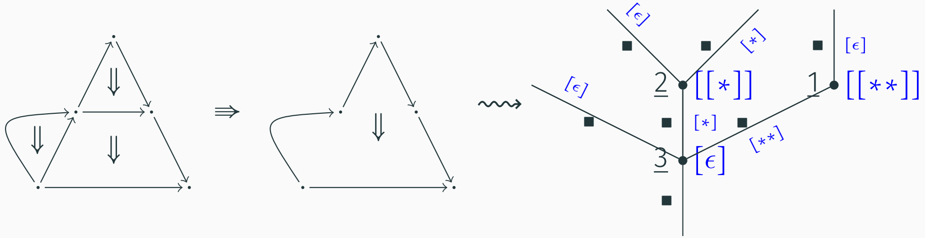
$$\vdots$$
$$\underline{3}$$

Example



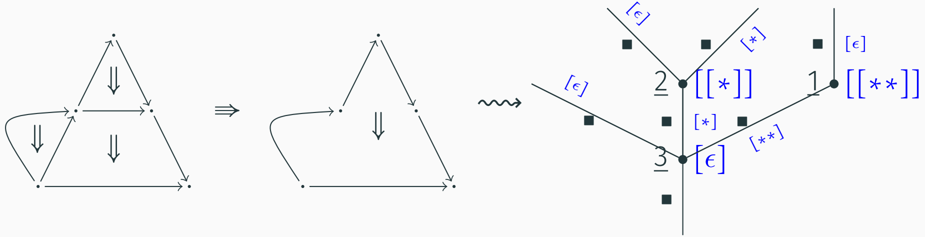
$$\dfrac{\begin{array}{c}\vdots\\ \underline{3}\end{array}}{\{[\epsilon] \leftarrow \underline{3}\}}\ \texttt{shift}$$

Example



$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \underline{3}\end{matrix}}{\{[\epsilon] \leftarrow \underline{3}}\ \texttt{shift} \qquad \begin{matrix}\vdots\\ \underline{2}\end{matrix}}{\begin{cases}[\epsilon] \leftarrow \underline{3}\\ [[*]] \leftarrow \underline{2}\end{cases}}\ \texttt{graft-}[[*]]$$

Example



$$
\dfrac{
  \dfrac{
    \dfrac{\begin{matrix}\vdots\\ \underline{3}\end{matrix}}{\{[\epsilon] \leftarrow \underline{3}}\ \mathtt{shift}
    \qquad
    \begin{matrix}\vdots\\ \underline{2}\end{matrix}
  }{
    \begin{cases}[\epsilon] \leftarrow \underline{3}\\ [[*]] \leftarrow \underline{2}\end{cases}
  }\ \mathtt{graft\text{-}}[[*]]
  \qquad
  \begin{matrix}\vdots\\ \underline{1}\end{matrix}
}{
  \begin{cases}[\epsilon] \leftarrow \underline{3}\\ [[*]] \leftarrow \underline{2}\\ [[**]] \leftarrow \underline{1}\end{cases}
}\ \mathtt{graft\text{-}}[[**]]
$$

Conclusion

Conclusion

- In this presentation, we gave two ways to define opetopes syntactically:
    1. in a "named" way, using terms and system $\mathsf{Opt}^!$;
    2. in an "unnamed" way, using preopetopes and system $\mathsf{Opt}^?$;

Conclusion

- In this presentation, we gave two ways to define opetopes syntactically:
    1. in a “named” way, using terms and system $\mathsf{Opt}^!$;
    2. in an “unnamed” way, using preopetopes and system $\mathsf{Opt}^?$;
- 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 [Ho 18], where valid proof trees are represented by certain expressions that evaluate without throwing any exception.

Conclusion

- In this presentation, we gave two ways to define opetopes syntactically:
  1. in a "named" way, using terms and system $\mathsf{Opt}^!$;
  2. in an "unnamed" way, using preopetopes and system $\mathsf{Opt}^?$;
- 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 [Ho 18], where valid proof trees are represented by certain expressions that evaluate without throwing any exception.
- In [CHM18] (see link on the first slide for a draft), we also present variants of those systems for opetopic sets.

Conclusion

- In this presentation, we gave two ways to define opetopes syntactically:
    1. in a "named" way, using terms and system $\mathsf{Opt}^{!}$;
    2. in an "unnamed" way, using preopetopes and system $\mathsf{Opt}^{?}$;
- 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 [Ho 18], where valid proof trees are represented by certain expressions that evaluate without throwing any exception.
- In [CHM18] (see link on the first slide for a draft), we also present variants of those systems for opetopic sets.
- We are experimenting with those new tools to automatically check coherence laws for an appropriate definition of opetopic ω-groupoid.

Thank you for your attention!

References i

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

Pierre-Louis Curien, Cédric Ho Thanh, and Samuel Mimram.
Type theoretical approaches for opetopes.
In preparation, 2018.

Cédric Ho Thanh.
opetopy.
`https://github.com/altaris/opetopy`, April 2018.

References ii

Joachim Kock, André Joyal, Michael Batanin, and Jean-François Mascari.
Polynomial functors and opetopes.
Advances in Mathematics, 224(6):2690–2737, 2010.

Tom Leinster.
Higher Operads, Higher Categories.
Cambridge University Press, 2004.