Opetopes and opetopic sets

(Picardie, juin 2021)

GAMBINO - KOCK 2013 (after JOYAL)

≈ (Grothendieck construction for) arities

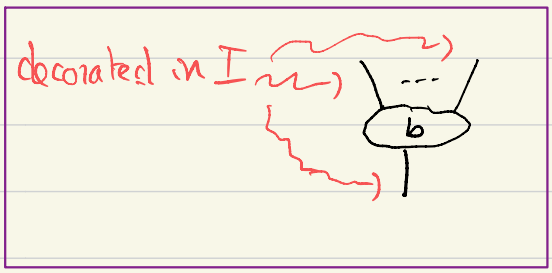
$$I \xleftarrow{\delta} E \xrightarrow{p} B \xrightarrow{t} I$$

operations

sorts

$E_b = p^{-1}(b)$

Polynomial (endo) functor



(polynomial for short)

$\Gamma \vdash b : t(b)$

$\Gamma = \{ x : \delta(x) \mid x \in E_b \}$

B Grothendieck construction for Cns Mi

$$\begin{aligned}
&\mathsf{Idx} : \mathbb{M} \to \mathcal{U} \\
&\mathsf{Cns} : (M : \mathbb{M}) \to \mathsf{Idx}\, M \to \mathcal{U} \\
&\mathsf{Pos} : (M : \mathbb{M})\ \{i : \mathsf{Idx}\, M\} \to \mathsf{Cns}\, M\, i \to \mathcal{U} \\
&\mathsf{Typ} : (M : \mathbb{M})\ \{i : \mathsf{Idx}\, M\}\ (c : \mathsf{Cns}\, M i) \\
&\qquad \to \mathsf{Pos}\, M\, c \to \mathsf{Idx}\, M
\end{aligned}$$

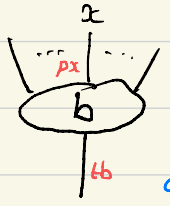
$\mathsf{Idx}\ M \rightsquigarrow I$

$\mathsf{Cns}\ M\ i \rightsquigarrow \{b \mid t(b) = i\}$

$\mathsf{Pos}\ M\ b \rightsquigarrow E_b$

$\mathsf{Typ}\ M\ b\ x \rightsquigarrow \rho x$

Id_X	provides	I
Cns	provides	B and t
Pos	provides	E and p
Typ	provides	δ



ports (or colours) CAN BE REPEATED

Think of $\Gamma = \{ \cdots\ x : \rho x \cdots \}$ as a record

positions DISTINCT

Functor induced by a polynomial

Every polynomial M induces a functor $[\_] : (\mathsf{Idx}\, M \to \mathcal{U}) \to (\mathsf{Idx}\, M \to \mathcal{U})$ called its *extension* given by

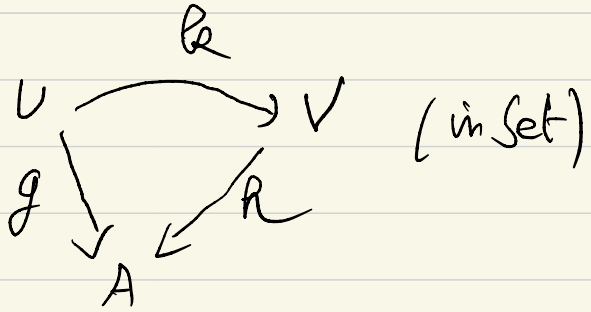
$$[M]\, X\, i = \sum_{(c : \mathsf{Cns}\, M\, i)} (p : \mathsf{Pos}\, M\, c) \to X\, (\mathsf{Typ}\, M\, c\, p)$$

Why is this functor called polynomial?

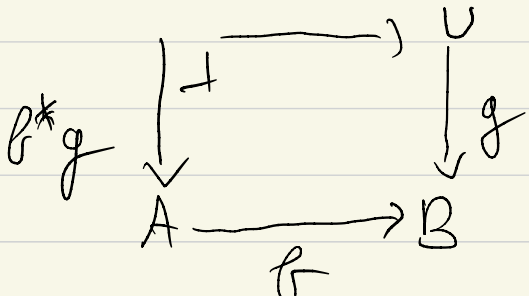
We need a little detour to explain the locally cartesian closed structure of Set

The locally cartesian closed structure of Set

- We fix a set A. We define Set/A [slice category]:
  - objects = pairs $(U, g: U \to A)$
  - morphisms = comm. triangles



- f induces $f^*: Set/B \to Set/A$ by pull-back

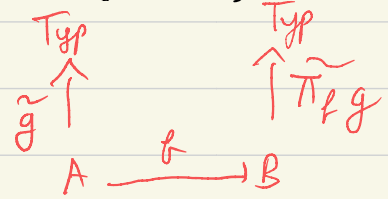


(choice of pb!)

- f^* has both a left and right adjoint $\Sigma_f \dashv f^* \dashv \Pi_f$
  - Σ_f is composition: $\Sigma_f g = f \circ g$, but...
  - Π_f is not described via Grothendieck deconstruction - construction:

for $g: U \to A$, let $\tilde{g}\, a = g^{-1}(a)$. Then

$$\widetilde{\Pi_f g}\, b = \prod_{\{a \mid fa = b\}} \tilde{g}\, a$$



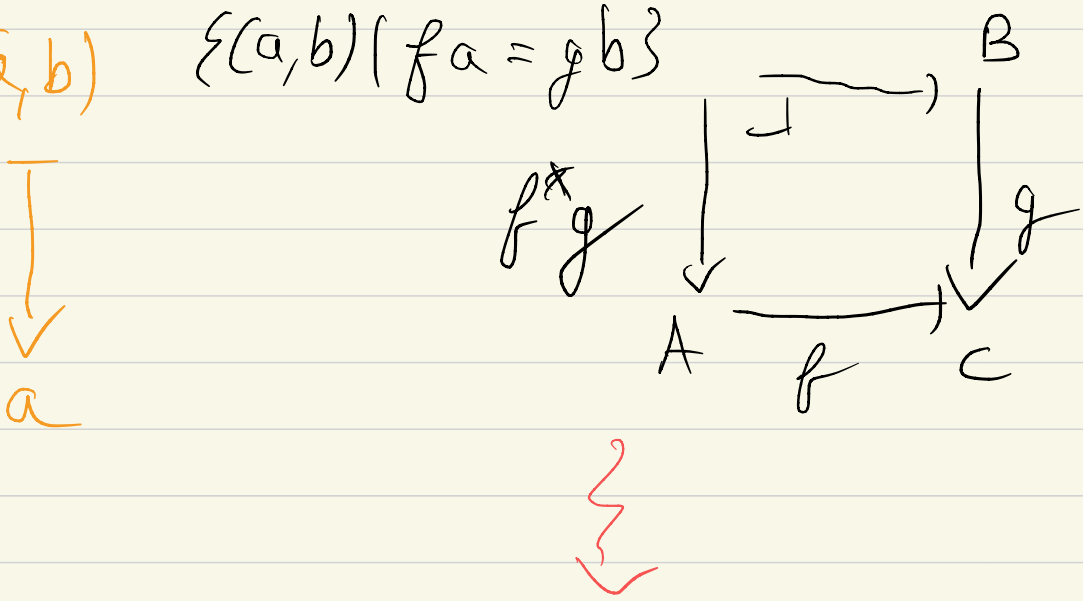
- Using the same angle, we have

$$\widetilde{\Sigma_f g}\, b = \sum_{\{a \mid fa = b\}} \tilde{g}\, a$$

Pullbacks too ... (in Set)

can be described nicely under the same angle

(a,b) $\quad \{(a,b) \mid fa = gb\}$



$$\widetilde{f^*g}\, a = \tilde{g}(fa)$$

Back to polynomial functors

A polynomial functor $I \xleftarrow{s} E \xrightarrow{p} B \xrightarrow{t} I$ induces a functor

$$\mathbf{Set}/I \xrightarrow{s^*} \mathbf{Set}/E \xrightarrow{\Pi_p} \mathbf{Set}/B \xrightarrow{\Sigma_t} \mathbf{Set}/I$$

Viewing $g : U \to I$ as a family $i \mapsto g^{-1}(i)$ (= X), we get (clearly)

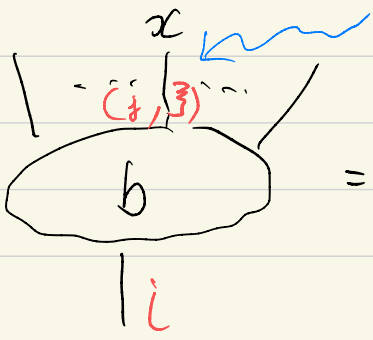
- by s^*: $x \mapsto g^{-1}(sx)$
- by Π_p: $b \mapsto \prod_{\{x \mid px = b\}} g^{-1}(sx)$
- by Σ_t: $i \mapsto \sum_{\{b \mid tb = i\}} \prod_{\{x \mid px = b\}} g^{-1}(sx)$

SUM of MONOMIALS

$\mathsf{Cns}\, M\, i$ $\quad$ $\mathsf{Pos}\, M\, b$ $\quad$ $X(\mathsf{Typ}\, M\, b\, x)$

EXTENSION of M

$$[M]\, X\, i = \sum_{(c : \mathsf{Cns}\, M\, i)} (p : \mathsf{Pos}\, M\, c) \to X\, (\mathsf{Typ}\, M\, c\, p)$$



where $\begin{cases} sx = j \\ 3 \in X_j \end{cases}$

= operations of M with inputs decorated by elements of X

A category of polynomials

The category $\mathbf{Pol}(I)$ has

- polynomial functors over I as objects
- morphisms are pairs (f_1, f_2) s.t.

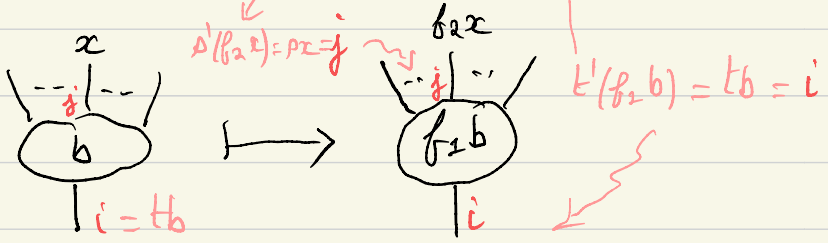
"operation part"

"arity part"

$$\begin{array}{ccccccc} P & I & \xleftarrow{s} & E & \xrightarrow{p} & B & \xrightarrow{t} & I \\ & \| & & f_2 \downarrow & \lrcorner & \downarrow f_1 & & \| \\ P' & I & \xleftarrow{s'} & E' & \xrightarrow[p']{} & B' & \xrightarrow{t'} & I \end{array}$$

pull-back!

The p.b. means: for all $b \in B$ $\quad E_b \simeq E'_{f_1(b)}$



Morphisms maps operations to operations of equipotent arity and respect sources

In FINSTER-ALLIOUX-SOZEAU (FAS)'s setting:

- $f_1 : \prod_{i:I} \mathrm{Cns}\, P_i \to \mathrm{Cns}\, P'_i$
- $f_2 : \prod_{i:I} \prod_{b : \mathrm{Cns}\, P_i} \prod_{x : \mathrm{Pos}\, P_b} \mathrm{Pos}\, P'(f_1\, i\, b)$
- with $\mathrm{Typ}\, P'(f_1\, i\, b)(f_2\, i\, b\, x) \rightsquigarrow \mathrm{Typ}\, P\, b\, x$

A monoidal structure on polynomials

- Let P, P' be polynomials. We define the operations of $P \circ P'$ as

where
- b is an operation of P
- For each $x \in E_b$ ($= \mathrm{Pos}\, Pb$), c_x is an operation of P' s.t. $t'(c_x) = s(x)$
- $t(b(\cdots x \leftarrow c_x \cdots)) = t(b)$

Hence

$$\mathrm{Ops}(P \circ P')\, i = \{ b(\cdots x \leftarrow c_x \cdots) \mid b \in \mathrm{Ops}\, P\, i,\ \forall x \in \mathrm{Pos}\, Pb,\ c_x \in \mathrm{Ops}\, P'(\mathrm{Typ}\, Pb\, x) \}$$
$$= \sum_{b \in \mathrm{Ops} P i} \prod_{x \in \mathrm{Pos} Pb} \mathrm{Ops}\, P'(\mathrm{Typ}\, Pb\, x) = [P](\mathrm{Ops}\, P')\, i$$

i.e., operations of $P \circ P'$ can be equivalently described as
- two-level trees (P at first level, P' at second level)
- operations of P decorated by operations in P'

- $\mathrm{Pos}(P \circ P')[b(\cdots, x \leftarrow c_x, \cdots)] = \sum_{x \in \mathrm{Pos}\, Pb} \mathrm{Pos}\, P' c_x$ (2)
- $\mathrm{Typ}(P \circ P')(b(\cdots, x \leftarrow c_x, \cdots))(x, y) = \mathrm{Typ}\, P'\, c_x\, y$ (3)

((b, δ); δx)

(1) singleton i.e. T

- The unit is $I \xleftarrow{id} I \xrightarrow{id} I \xrightarrow{id} I$ (in particular $\forall i$, $\mathrm{Pos}\, I\, i = \{i\}$)

This gives $\mathbb{P}\mathrm{oly}(I)$ the structure of a monoidal category

More widely, there is a bicategory whose
- objects are sets $I, J, \ldots$
- 1-morphisms are polynomials $I \leftarrow B \rightarrow E \rightarrow J$
- 2-morphisms are defined like p.6

A polynomial monad is a monoid object in $\mathbb{P}\mathrm{oly}(I)$

Polynomial monad

Thus, a polynomial monad is the data of

- a polynomial functor M
- a unity $\eta : I \to M$ and a multiplication $\mu : M \circ M \to M$ satisfying the monoid laws

viewed as operation in $I \leftarrow I \to I \to I$

$$\begin{aligned}
&\eta : (M : \mathbb{M})\ (i : \mathsf{Idx}\, M) \to \mathsf{Cns}\, M\, i \\
&\mu : (M : \mathbb{M})\ \{i : \mathsf{Idx}\, M\}\ (c : \mathsf{Cns}\, M\, i) \\
&\quad \to (\delta : (p : \mathsf{Pos}\, M\, c) \to \mathsf{Cns}\, M\, (\mathsf{Typ}\, M\, c\, p)) \\
&\quad \to \mathsf{Cns}\, M\, i
\end{aligned}$$

By uncurrying: $\mu M \{i\} : [M](\mathsf{Cns}\, M)\, i \to \mathsf{Cns}\, M\, i$

$\mathsf{Cns}\,(M \circ M)\, i$

So far for the "operation part" of η, μ

Since η, μ respect arities we have, by ❶ and ❷ (p. 7), that

- $\mathsf{Pos}\, M\, (\eta\, M\, i)$ behaves like T
- $\mathsf{Pos}\, M\, (\mu\, M\, c\, \delta)$ behaves like $\sum_{(p : \mathsf{Pos}\, M\, c)} \mathsf{Pos}\, (M\, c\, \delta)$

whence a description via the intro-elim rules of T, Σ-types:

$$\begin{aligned}
\eta\text{-pos} &: (M : \mathbb{M})\ (i : \mathsf{Idx}\, M) \to \mathsf{Pos}\, M\, (\eta\ M\, i) \\
\eta\text{-pos-elim} &: (M : \mathbb{M})\ (i : \mathsf{Idx}\, M) \\
&\quad \to (X : (p : \mathsf{Pos}\, M\, (\eta\ M\, i)) \to \mathcal{U}) \\
&\quad \to (u : X\ (\eta\text{-pos}\, M\, i)) \\
&\quad \to (p : \mathsf{Pos}\, M\, (\eta\ M\, i)) \to X\ p
\end{aligned}$$

$$\begin{aligned}
\mu\text{-pos} &: (M : \mathbb{M})\ \{i : \mathsf{Idx}\, M\}\ \{c : \mathsf{Cns}\, M\, i\} \\
&\quad \to \{\delta : (p : \mathsf{Pos}\, M\, c) \to \mathsf{Cns}\, M\, (\mathsf{Typ}\, M\, c\, p)\} \\
&\quad \to (p : \mathsf{Pos}\, M\, c) \to (q : \mathsf{Pos}\, M\, (\delta\ p)) \\
&\quad \to \mathsf{Pos}\, M\, (\mu\ M\, c\, \delta)
\end{aligned}$$

$$\begin{aligned}
\mu\text{-pos-fst} &: (M : \mathbb{M})\ \{i : \mathsf{Idx}\, M\}\ \{c : \mathsf{Cns}\, M\, i\} \\
&\quad \to \{\delta : (p : \mathsf{Pos}\, M\, c) \to \mathsf{Cns}\, M\, (\mathsf{Typ}\, M\, c\, p)\} \\
&\quad \to \mathsf{Pos}\, M\ (\mu\ M\, c\, \delta) \to \mathsf{Pos}\, M\, c \\
\mu\text{-pos-snd} &: (M : \mathbb{M})\ \{i : \mathsf{Idx}\, M\}\ \{c : \mathsf{Cns}\, M\, i\} \\
&\quad \to \{\delta : (p : \mathsf{Pos}\, M\, c) \to \mathsf{Cns}\, M\, (\mathsf{Typ}\, M\, c\, p)\} \\
&\quad \to (p : \mathsf{Pos}\, M\, (\mu\ M\, c\, \delta)) \\
&\quad \to \mathsf{Pos}\, M\, (\delta\ (\mu\text{-pos-fst}\, M\, p))
\end{aligned}$$

+ computation rules

$$\eta\text{-pos-elim}\ M\, i\, X\, u\, (\eta\text{-pos}\, M\, i) \rightsquigarrow u$$

$$\begin{aligned}
\mu\text{-pos-fst}\, M\, (\mu\text{-pos}\, M\, p\, q) &\rightsquigarrow p \quad (\mu\text{-pos-fst}) \\
\mu\text{-pos-snd}\, M\, (\mu\text{-pos}\, M\, p\, q) &\rightsquigarrow q \quad (\mu\text{-pos-snd}) \\
\mu\text{-pos}\, M\, (\mu\text{-pos-fst}\, M\, p)\, (\mu\text{-pos-snd}\, M\, p) &\rightsquigarrow p \quad (\mu\text{-pos-}\eta)
\end{aligned}$$

≃ surjective pairing

What about the "arity part" of η, μ?

We have that μ-pos is the arity part of μ_2.

Therefore, the rewriting rule of p.6

$$\mathrm{Typ}\ P'\ (f_1\ i\ b)\ (f_2\ i\ b\ x) \rightsquigarrow \mathrm{Typ}\ P\ b\ x$$ ●$_0$

instantiates as (using surjective pairing ●$_4$ p.8):

$$\mathrm{Typ}\ M\ (\mu M\ c\ \delta)\ p \rightsquigarrow \mathrm{Typ}\ (M \circ M)\ (c, \delta)\ (x, y)$$

(μ-Pos-fst M p, μ-Pos-snd M p)

Now, instantiating

$$\mathrm{Typ}\ (P \circ P')\ (c\ (\cdots, x \Leftarrow c_x, \cdots))\ (x, y) = \mathrm{Typ}\ P'\ c_x\ y$$ ●$_3$ (p.7), we get

(c, δ) — c_x = δx

$$\mathrm{Typ}\ (M \circ M)\ (c, \delta)\ (x, y) = \mathrm{Typ}\ M\ (\delta x)\ y$$, yielding

$$\mathrm{Typ}\ M\ (\mu\ M\ c\ \delta)\ p \rightsquigarrow \mathrm{Typ}\ M\ (\delta\ (\mu\text{-pos-fst}\ M\ p))\ (\mu\text{-pos-snd}\ M\ p)$$

expressing that μ respects sorts

Similarly (up to the fact that for "Agdaic" reasons FAS does not include an η-rule for η-pos)

$$\mathrm{Typ}\ M\ (\eta\ M\ i)\ p \rightsquigarrow i$$

expresses that η respects sorts

The monoid laws of M

$$\mu\ M\ c\ (\lambda\ p \to \eta\ M\ (\mathsf{Typ}\ M\ c\ p)) \rightsquigarrow c \qquad (\mu\text{-}\eta\text{-r})$$
$$\mu\ M\ (\eta\ M\ i)\ \delta \rightsquigarrow \delta\ (\eta\text{-}\mathsf{pos}\ M\ i) \qquad (\mu\text{-}\eta\text{-l})$$
$$\mu\ M\ (\mu\ M\ c\ \delta)\ \epsilon \rightsquigarrow \qquad (\mu\text{-}\mu)$$
$$\mu\ M\ c\ (\lambda\ p \to\ \mu\ M\ (\delta\ p)\ (\lambda\ q \to \epsilon\ (\mu\text{-}\mathsf{pos}\ M\ p\ q)))$$

Relation to operads

A polynomial monad is the same thing as a coloured operad with a free action of the symmetric group. This means that the reindexing after composition (encoded by μ-pos) is not necessarily planar. If it is, then we have a non-symmetric operad.

In operadic notation, we write

- $\eta M i = id_i$
- $\mu M b \delta = b \circ \{ \cdots x \leftarrow \delta x \cdots \}$

And the above laws are reformulated as (first try)

$$b \circ \{ \cdots x \leftarrow id_{\mathsf{Typ} M b x} \cdots \} = b$$

$$id_i \circ \{ \bullet \leftarrow b \} = b \quad (\text{where } \mathsf{Pos}\ id_i = \{\bullet\})$$

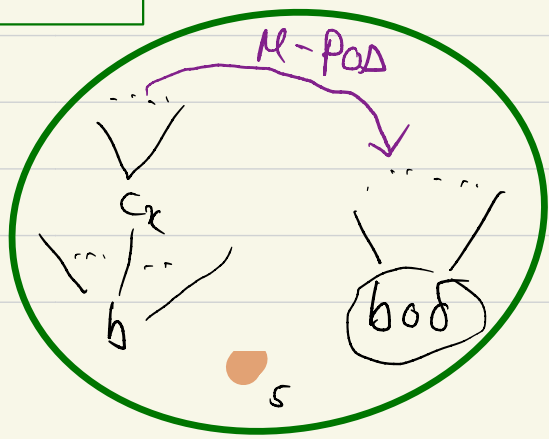
$$(b \circ \{ \cdots x \leftarrow c_x \cdots \}) \circ \{ \cdots (x,y) \leftarrow d_{x,y} \} = b \circ \{ \cdots x \leftarrow (c_x \circ \{ \cdots, y \leftarrow d_{x,y}, \cdots \}) \}$$

cf. substitution lemma!

More careful:

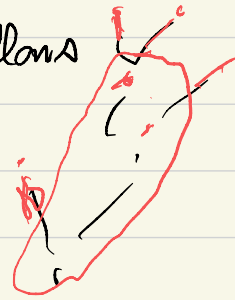
$$(b \circ \{ \cdots x \leftarrow c_x \cdots \}) \circ \{ \cdots p \leftarrow d_p \cdots \} =$$

$$b \{ \cdots x \leftarrow c_x \circ \{ \cdots, y \leftarrow \delta_{\mu\text{-}\mathsf{Pos} M b \delta x y} \}$$



The Star construction (implicit in FAS)

This construction builds the free polynomial monad P^* over a polynomial functor P. Its operations are trees, defined here as terms. The set I of sorts is fixed, i.e. $Idx\ P = Idx\ P^*$. We define $T_i = Cns\ P^*\ i$ as follows



$$T_i = \langle i \rangle \mid \Pi_i \qquad \Pi_i = b\{\dots x \leftarrow T_{Typ\ P\ b\ x} \dots\}$$

(i in I; b in $Cns\ P\ i$)

ie

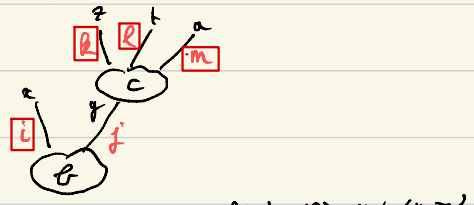
$$Cns\ P^*\ i = I + \sum_{b : Cns\ P\ i} \prod_{x : Pos\ P\ b} Cns\ P^*\ (Typ\ P\ b\ x)$$

The positions of a tree are its **leaves**

$$Pos\ P^*\ \langle i \rangle = \{i\}$$

$$Pos\ P^*\ (b, \delta) = \sum_{x : Pos\ P\ b} Pos\ P^*\ \delta x$$

$\langle i \rangle$



$b\{x \leftarrow \langle i \rangle, y \leftarrow c\{z \leftarrow \langle e \rangle, t \leftarrow \langle e \rangle, u \leftarrow \langle m \rangle\}$

$$Typ\ P^*\ \langle i \rangle\ i = i$$

$$Typ\ P^*\ (b, \delta)\ (x, y) = Typ\ P^*\ \delta x\ y$$

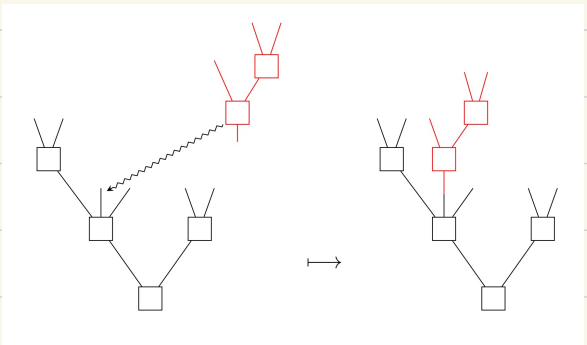
- The composition is by **grafting on the leaves**

$$\langle i \rangle\ \{i \leftarrow t\} = t$$

$$(b\{\dots x \leftarrow P_x \dots\})\ \{\dots x,y \leftarrow t_{x,y}, \dots\}$$

(term builder; composition builder)

$$b\{\dots x \leftarrow P_x\ \{\dots, y \leftarrow t_{x,y}, \dots\}, \dots\}$$



- The unit is the base case of trees: $\eta\ P^*\ i = \langle i \rangle$

Note that our trees have

- edges decorated by elements of I
- nodes ——— of B

The plus (or plie) construction

The plie (BAEZ-DOLAN 98) or plus (KOCK-JOYAL-BATANIN-MARCARI 2010) construction is what allows us to define trees of trees of ... or types

The iterated plus construction gives opetopes.

The iterated (plus + pullback) construction gives opetopic sets (FAS)

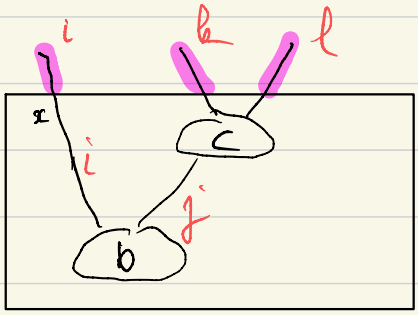
The star construction starts from a polynomial P over I and builds a polynomial monad P^* over I

The plus construction starts from a polynomial monad M over I with underlying polynomial $|M|$ $I \leftarrow E \rightarrow B \rightarrow I$ and builds a new polynomial monad over B. The operations become sorts!

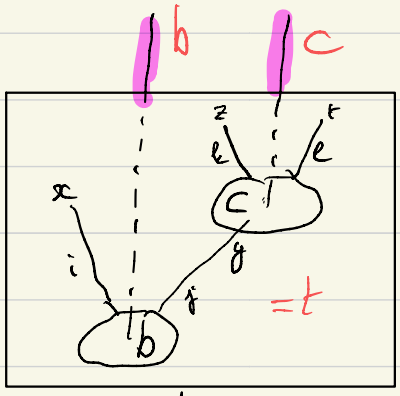
The two constructions have the "same" operations, i.e. trees of operations (of P, $|M|$) but arities are different

$\mathrm{Pos}\, P^* t$ = leaves of t

$\mathrm{Pos}\, P^+ t$ = nodes of t



VS



The output sort is the overall composition of the tree obtained by the monad structure on M.

$b \circ \{x \leftarrow \eta^M i,\ y \leftarrow c\} = [\![t]\!]_M$

The implementation in FAS does this "on the fly" and is forced to do it!

by integrating it in the definition of Tree (next page)

Important invariant: the leaves of t are in bijection with the inputs of $[\![t]\!]_M$.

The plus construction formally

- $\mathsf{Idx}\,(\mathsf{Slice}\,M) = \sum_{(i:\mathsf{Idx}\,M)} \mathsf{Cns}\,M\,i$ ← This is B !

- ```
  data Tree : Idx (Slice M) → U where
    lf  : (i : Idx M) → Tree (i , η M i)
    nd  : {i : Idx M} (c : Cns M i)
        → (δ : (p : Pos M c) → Cns M (Typ M c p))
        → (ϵ : (p : Pos M c) → Tree (Typ M c p , δ p))
        → Tree (i , μ M c δ)
  ```

  ← accumulator; ← recursive definition of tree; overall composition of the tree in M

Cns (Slice M) = Tree

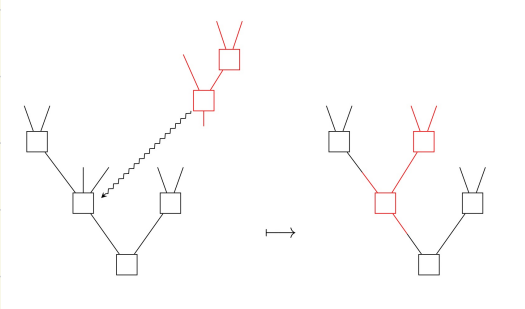
- $$\begin{aligned}
  &\mathsf{Pos}\,(\mathsf{Slice}\,M)\,(\mathsf{lf}\ i) &&= \bot \\
  &\mathsf{Pos}\,(\mathsf{Slice}\,M)\,(\mathsf{nd}\ c\,\delta\,\epsilon) &&= \\
  &\qquad \top \sqcup \sum_{(p:\mathsf{Pos}\,M\,c)} \mathsf{Pos}\,(\mathsf{Slice}\,M)\,(\epsilon\,p)
  \end{aligned}$$

  ← recursive definition of nodes (t)

- $$\begin{aligned}
  &\mathsf{Typ}\,(\mathsf{Slice}\,M)\,(\mathsf{lf}\ i)\,() \\
  &\mathsf{Typ}\,(\mathsf{Slice}\,M)\,(\mathsf{nd}\ \{i\}\,c\,\delta\,\epsilon)\,(\mathsf{inl}\,\mathsf{tt}) &&= (i\,,c) \\
  &\mathsf{Typ}\,(\mathsf{Slice}\,M)\,(\mathsf{nd}\ \{i\}\,c\,\delta\,\epsilon)\,(\mathsf{inl}\,(p\,,q)) &&= \\
  &\qquad \mathsf{Typ}\,(\mathsf{Slice}\,M)\,(\epsilon\,p)\,q
  \end{aligned}$$

  ← retrieves the decoration of a node of t
  ```

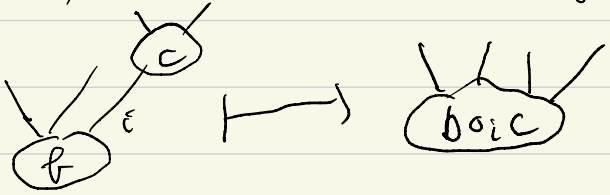
The monad structure of Slice M (alias M^+)



Trees are substituted into nodes.
The definition calls the monad structure on $|M|^*$

cf. second-order substitution !

The following sketch of a definition, based on a variant of μ called partial composition



highlights its trickyness

- $c\{\cdots, x \leftarrow p_x, \cdots\} \circ_{\mathrm{in}(i,t)} t = t\{\cdots, x' \leftarrow p_x, \cdots\}$

where the $\cdots$ are unchanged and the mapping $x \mapsto x'$ matches the leaves of t with the input positions of $C = \|t\|_M$ (cf. ● 6 p.12)

- $c\{\cdots, x \leftarrow p_x, \cdots\} \circ_{\mathrm{in}(x,y)} t = c\{\cdots, x \leftarrow p_x \circ_y t, \cdots\}$
- no case for $\langle i \rangle \circ \_$ (par absence de combattants)

As for the unit: $\eta^{M^+} b = b\{\cdots, x \leftarrow \langle t_x \rangle, \cdots\}$

The monad structure of Slice M, formally

$$
\begin{aligned}
&\gamma : (M : \mathbb{M})\,\{i : \mathsf{Idx}\,M\}\,(c : \mathsf{Cns}\,M\,i)\\
&\quad \to (\sigma : \mathsf{Cns}\,(\mathsf{Slice}\,M)\,(i, c))\\
&\quad \to (\phi : (p : \mathsf{Pos}\,M\,c) \to \mathsf{Cns}\,M\,(\mathsf{Typ}\,M\,c\,p))\\
&\quad \to (\psi : (p : \mathsf{Pos}\,M\,c) \to \mathsf{Cns}\,(\mathsf{Slice}\,M)\,(\mathsf{Typ}\,M\,c\,p, \phi\,p))\\
&\quad \to \mathsf{Cns}\,(\mathsf{Slice}\,M)(i, \mu\,M\,c\,\phi)
\end{aligned}
$$

$$
\begin{aligned}
\gamma\,M\,(\mathsf{If}\ i)\,\delta\,\epsilon &= \epsilon\,(\eta\text{-}\mathsf{pos}\,M\,i)\\
\gamma\,M\,(\mathsf{nd}\ c\,\delta\,\epsilon)\,\phi\,\psi &= \mathsf{nd}\ c\,\delta'\,\epsilon'
\end{aligned}
$$

where we define

$$
\begin{aligned}
\phi'\,p\,q &= \phi(\mu\text{-}\mathsf{pos}\,M\,c\,\delta\,p\,q)\\
\psi'\,p\,q &= \psi(\mu\text{-}\mathsf{pos}\,M\,c\,\delta\,p\,q)\\
\delta'\,p &= \mu\,M\,(\delta\,p)\,(\phi' p)\\
\epsilon'\,p &= \gamma\,M\,(\epsilon\,p)\,(\psi' p)
\end{aligned}
$$

This is the monad structure of $|M|^*$...

$$
\begin{aligned}
\mu\,(\mathsf{Slice}\,M)\,(\mathsf{If}\ i)\,\phi &= \mathsf{If}\ i\\
\mu\,(\mathsf{Slice}\,M)\,(\mathsf{nd}\ c\,\delta\,\epsilon)\,\phi &= \gamma\,M\,w\,\delta\,\psi
\end{aligned}
$$

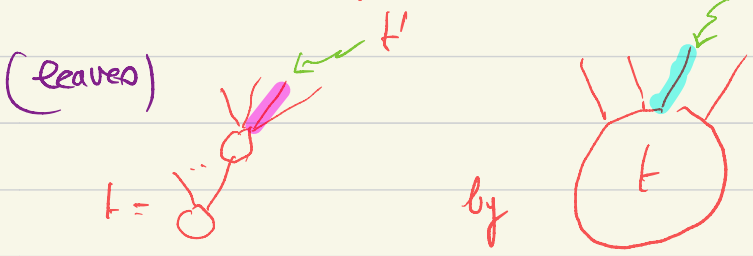
where we put

$$
\begin{aligned}
w &= \phi\,(\mathsf{inl}\,\mathsf{tt})\\
\phi'\,p\,q &= \phi\,(\mathsf{inr}\,(p, q))\\
\psi\,p &= \mu\,(\mathsf{Slice}\,M)\,(\epsilon\,p)\,(\phi'\,p)
\end{aligned}
$$

This code cleverly "encodes" the book-keeping $x \mapsto x'$ of previous page by FORCING $x = x'$

... presented in a way that is aware of M!

The trick is to replace



(see p. 17 for a generalization)

$t\,\{\cdots, x' \in P_{x'}, \cdots\}$ ⇝ $t\,[\cdots, x \in P_x, \cdots]$

{leaf of t} {input of [[t]]}

A formal syntax for (Slice M)

Here is a "type-theoretic" reformulation of Slice M where

- we adopt the partial style for M (Slice M)
- we adopt the "$\{ \dots \}$ to $[\dots]$ trick" of p. 15
- Structure of polynomial:

$$\vdash \langle i \rangle : \eta M_c$$

$$\frac{\cdots \Gamma_x \vdash t_x : b_x \cdots \quad (x \in E_b)}{\{\bullet : b\}\, t \underset{x \in E_b}{\leq} \Gamma_x \vdash b\{\cdots, x \leftarrow t_x, \cdots\} : b \circ_M \{\cdots x \leftarrow b_x \cdots\}}$$

- Partial composition

$$c\{\cdots, x \leftarrow A_x, \cdots\} \circ_{\mathrm{inr}(x,y)} t = c\{\cdots, x \leftarrow A_x \circ_y t, \cdots\}$$

$$c\{\cdots, x \leftarrow A_x, \cdots\} \circ_{\mathrm{inl}(*)} t = t\,[\cdots, x \leftarrow A_x, \cdots]$$

where

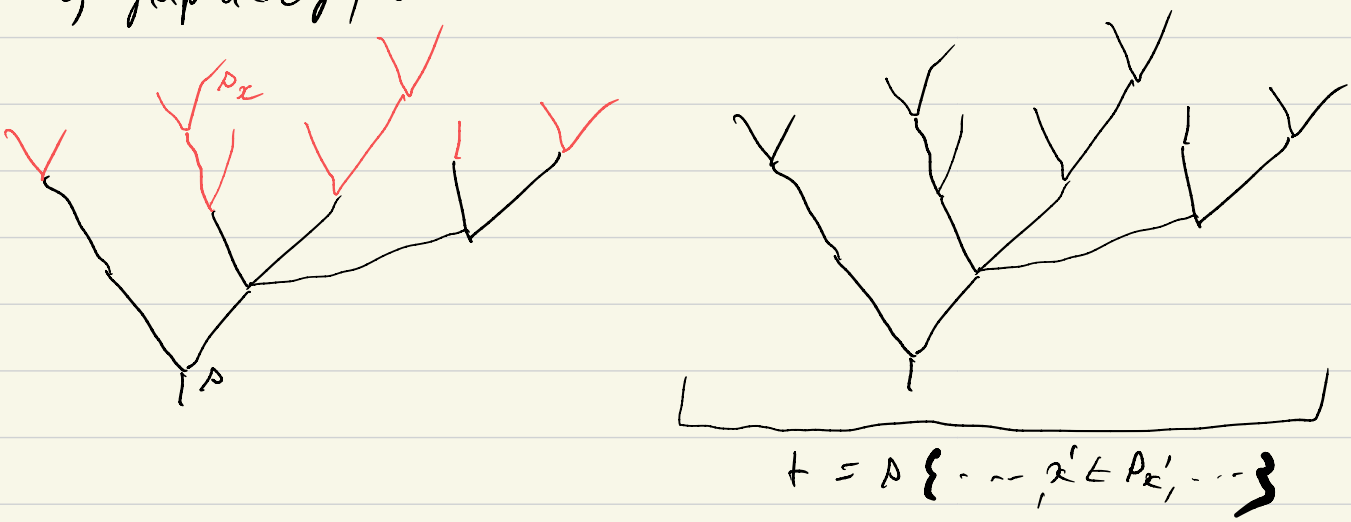
$$\langle i \rangle [\underbrace{\cdots x \leftarrow A_{x_1} \cdots}_{\text{singleton}}] = A_{(\eta\text{-pos } M\ i)}$$

$$(b\{\cdots, y \leftarrow t_{y_1}^{b_y} \cdots\})^{b \circ \{\cdots y \leftarrow b_y \cdots\}} [\cdots, z \leftarrow A_z, \cdots] =$$

$$b\{\cdots y \leftarrow t_y [\cdots, z \leftarrow A_{(\mu\text{-pos } M\ b\ (\lambda y.\, b_y)\ y\ z)}]\}$$

Behind the scene...

What makes the trick work is the fact that the "overall composition" on M
$t \mapsto [\![t]\!]$ has the structure of a $(-)^*$-algebra, which means, graphically, for



that $[\![H]\!] = [\![o]\!] \circ_M \{\cdots, x \leftarrow [\![A_x]\!], \cdots\}$

i.e., we have the following heterogeneous commutation

$$[\![P\{\cdots, x' \leftarrow P_{x'}, \cdots\}]\!] = [\![P]\!] \circ_M \{\cdots, x \leftarrow [\![A_x]\!], \cdots\}$$

By "pretending" that the leaves of P **are** the inputs of $[\![P]\!]$, FAS manage to be able to consider **only** the instance of the $x \mapsto x'$ bijection σ_6, namely that provided by σ_5 (p. 10).

Opetopes

- We start from $B^0 = \{\blacklozenge\}$
  (unique opetope = shape for a point $\bullet$)
- We then define $\mathcal{O}^1$ as the following identity monad:

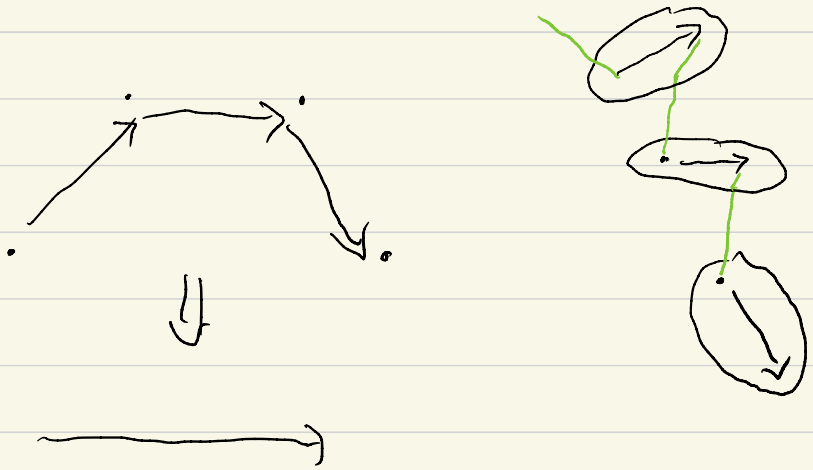
$$\{\blacklozenge\} \leftarrow \{*\} \rightarrow \{\blacksquare\} \rightarrow \{\blacklozenge\}$$

$$\{\blacksquare\} = B^1 \quad \text{(unique 1-opetope } \bullet \rightarrow \bullet\text{)}$$

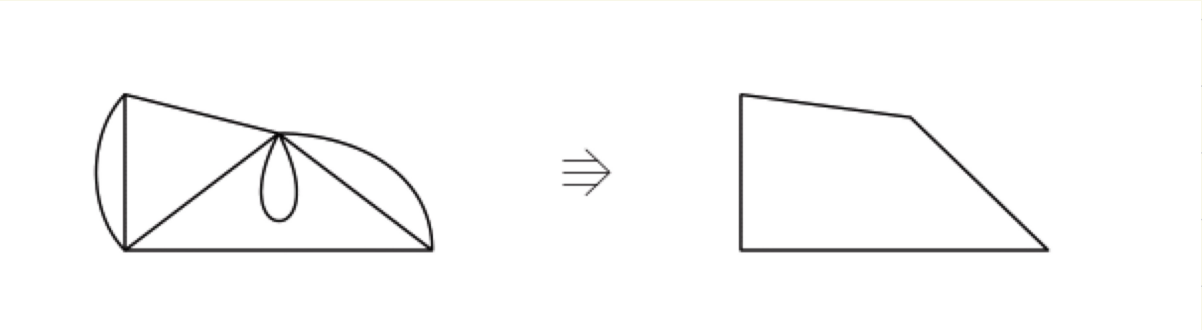
(which deserves its name since $[\mathcal{O}^1]$ is the identity functor in Set)

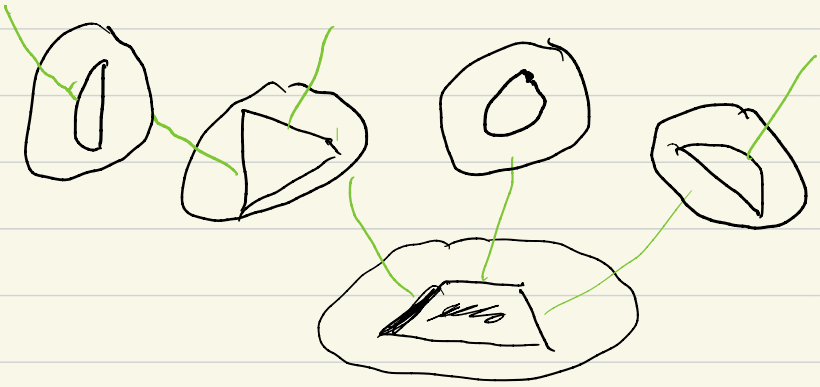
- Induction $\mathcal{O}^n = (\mathcal{O}^{n-1})^+$

2-opetopes are trees of 1-opetopes (they are linear)

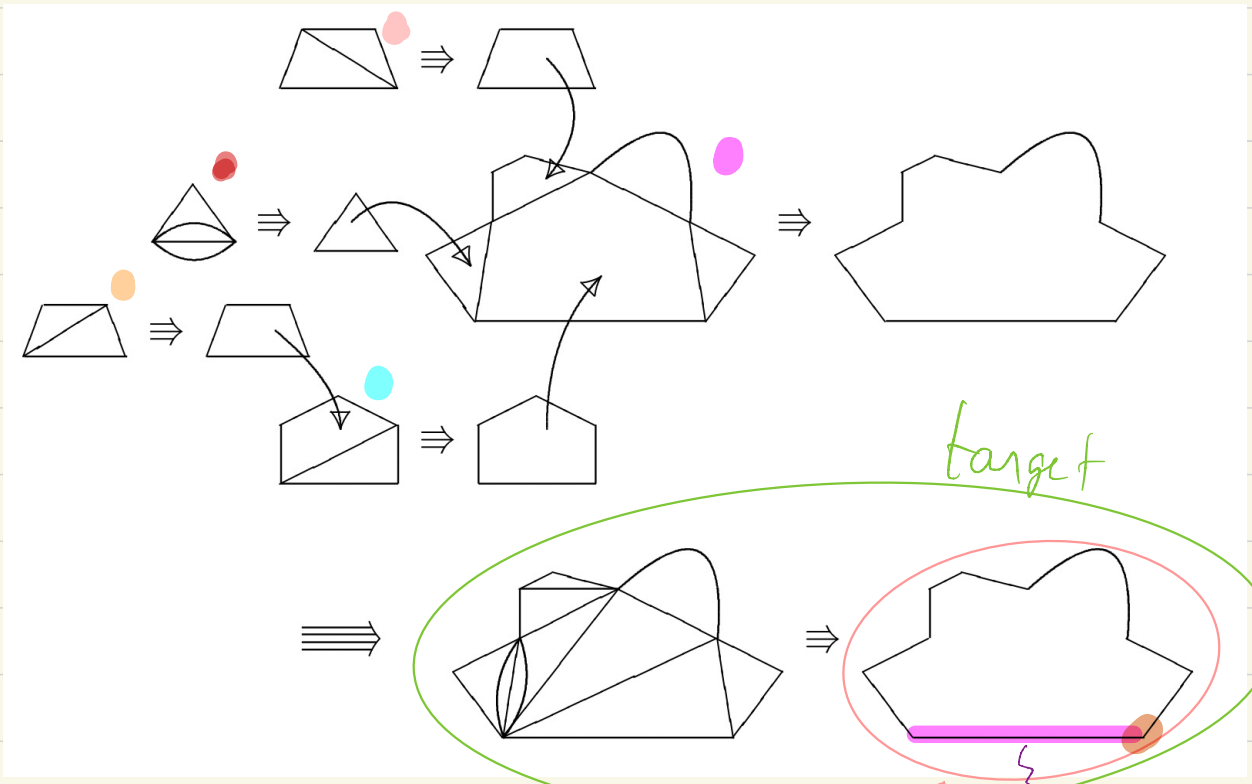


A 3-operype

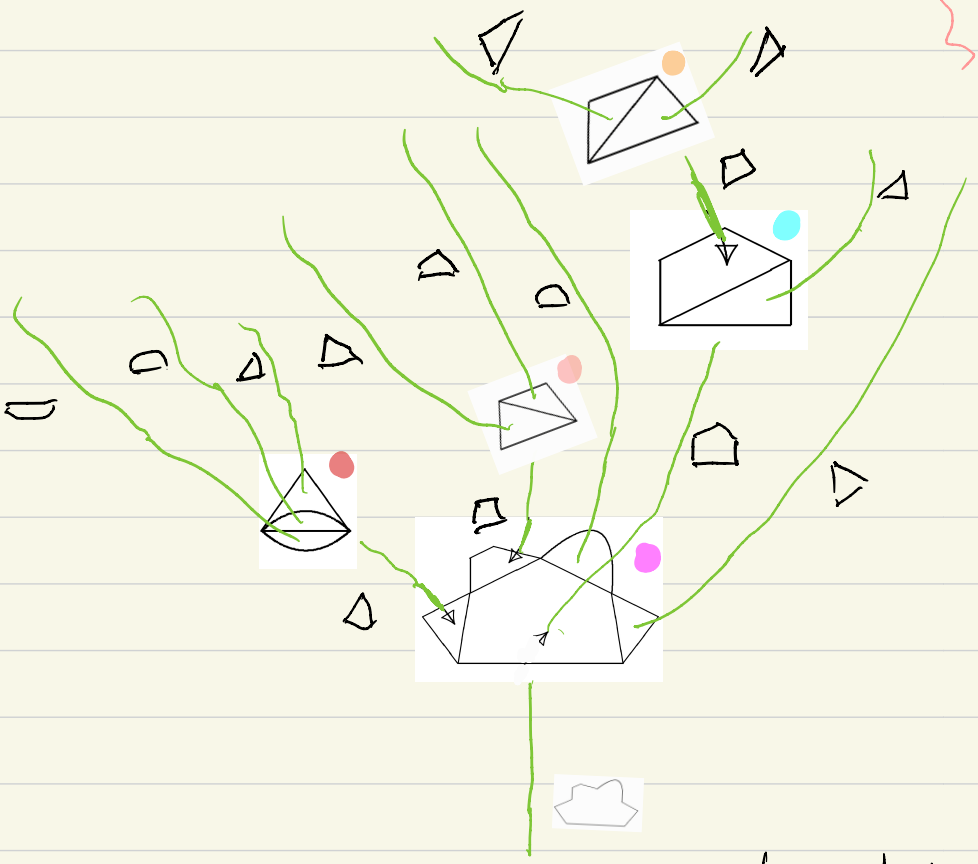




A 4-opetope

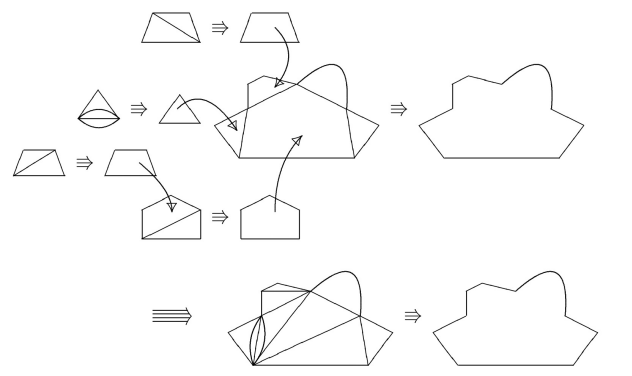


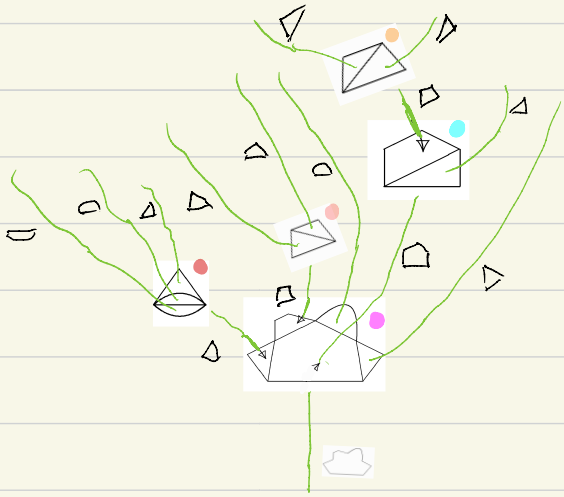
● = target^4

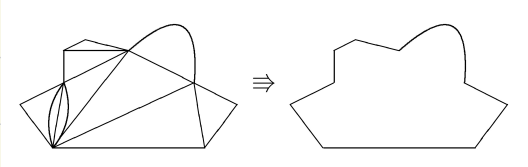


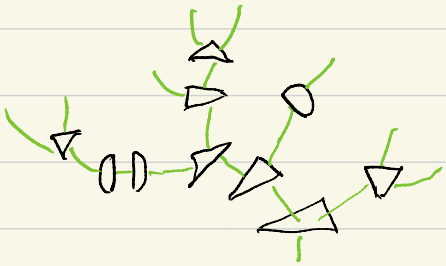
(node, decorated by 3-opetopes, edges decorated by 2-opetopes

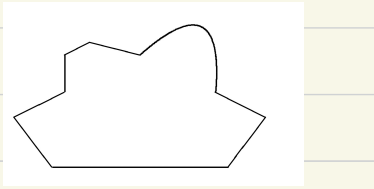
A cascade of targets

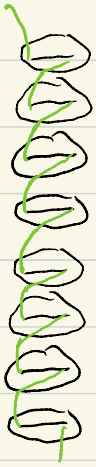












Constellations (KJBM)

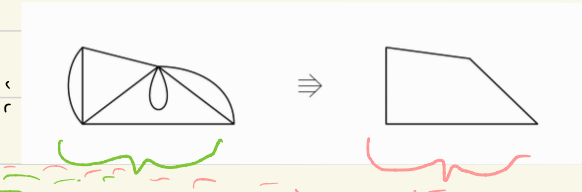
Idea: distribute the information over all trees in the cascade by providing additional information between them

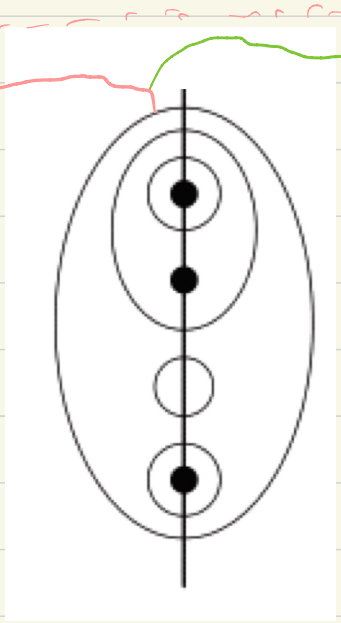
I.e. replace a single tree whose nodes are decorated by $(n-2)$-opetopes and whose edges are decorated by $(n-1)$-opetopes

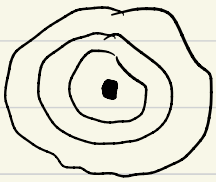
for each iterated target

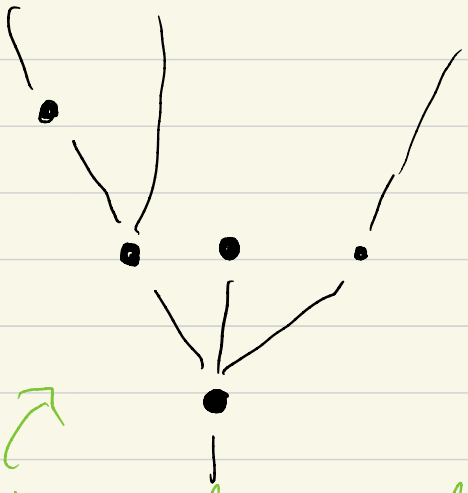
by a sequence of trees with circles

For our example of 3-opetopes:

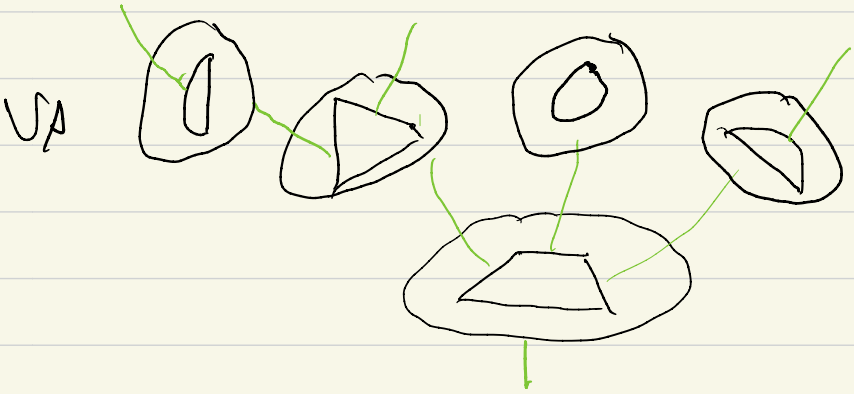






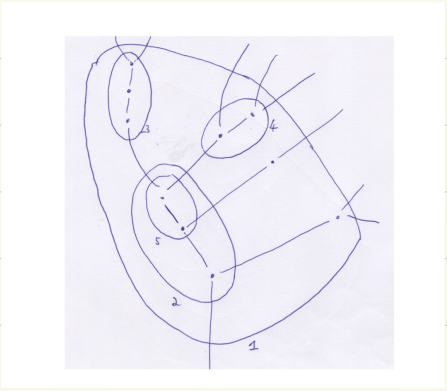


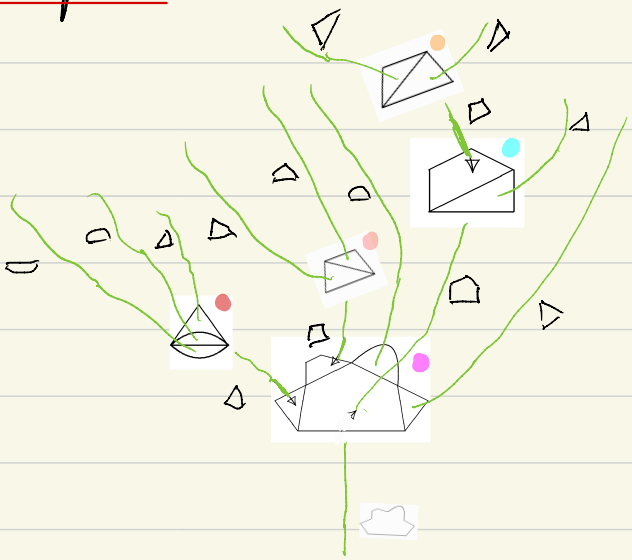
ready for being circled to describe a 4-opetope with target



etc.

The constellation of our 4-opetope

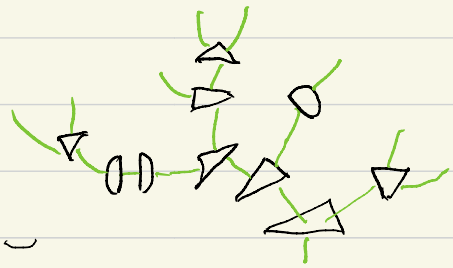


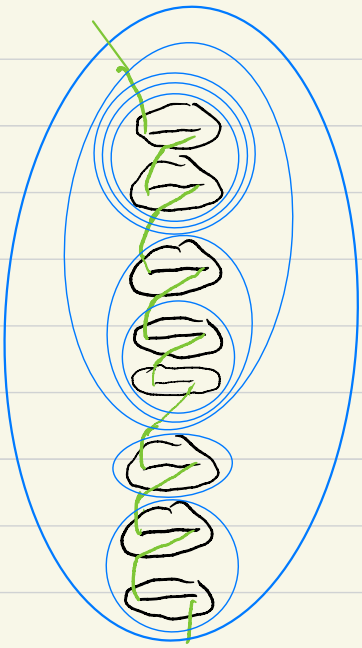


node ⇝ leaver

circle ⇝ nodes

circle added

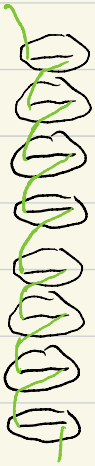


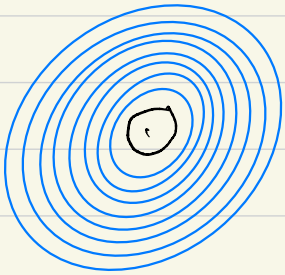


node ⇝ leaver

circle ⇝ nodes

circle added





The pull-back construction and opetopic types

$$\mathsf{Pb} : (M : \mathbb{M})\,(X : \mathsf{Idx}\,M \to \mathcal{U}) \to \mathbb{M}$$

$$\eta\ (\mathsf{Pb}\,M\,X)\ (i\,,x) = (\eta\ M\,i\,,\ \eta\text{-}\mathsf{dec}\,M\,X\,x)$$

$$\begin{aligned}
\mathsf{Idx}\,(\mathsf{Pb}\,M\,X) &= \textstyle\sum_{(i:\mathsf{Idx}\,M)} X\ i \\
\mathsf{Cns}\,(\mathsf{Pb}\,M\,X)\ (i,x) &= \\
&\quad \textstyle\sum_{(c:\mathsf{Cns}\,M\,i)} \prod_{(p:\mathsf{Pos}\,M\,c)} X\,(\mathsf{Typ}\,M\,c\,p) \\
\mathsf{Pos}\,(\mathsf{Pb}\,M\,X)\ (c,\nu) &= \mathsf{Pos}\,M\,c \\
\mathsf{Typ}\,(\mathsf{Pb}\,M\,X)\ (c,\nu)\ p &= (\mathsf{Typ}\,M\,c\ p\,,\ \nu\ p)
\end{aligned}$$

$$\mu\ (\mathsf{Pb}\,M\,X)\ (c\,,\nu)\ \delta = (\mu\ M\,c\ \delta'\,,\nu')$$

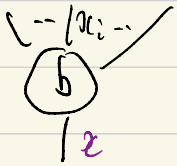
where

$$\begin{aligned}
\delta'\ p &= \mathsf{fst}\ (\delta\,p) \\
\nu'\ p &= \mathsf{snd}\ (\delta\,(\mu\text{-}\mathsf{pos}\text{-}\mathsf{fst}\,p))\,(\delta\,(\mu\text{-}\mathsf{pos}\text{-}\mathsf{snd}\,p))
\end{aligned}$$

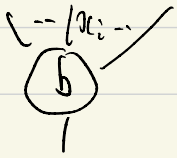
NB: Pb M is similar to, but different from [M]

≀ ≀

operations of $\mathcal{B}$ whose inputs and output are decorated in X



operations of $\mathcal{B}$ whose inputs are decorated in X



(coinductive)

```
record OpetopicType (M : 𝕄) : 𝒰₁ where
  𝒞 : Idx M → 𝒰
  ℛ : OpetopicType (Slice (Pb M 𝒞))
```

↳ (iterated slice + p.b.) construction

(whose opetopes are obtained by iterated slice only)

(inductive)

$$\mathsf{Opetopes}\,(M) = \left(\sum_{i\,:\,\mathsf{Idx}\,M} \mathsf{Cns}\ M\,i\right) + \mathsf{Opetopes}\,(\mathsf{Slice}\ M)$$

$$\mathsf{Opetopes} = \mathsf{Opetopes}\ \mathsf{Id}$$

where $\mathsf{Idx}\,\mathsf{Id} = \top$ $\quad \mathsf{Cns}\,\mathsf{Id}\,\mathsf{tt} = \top$ $\quad \mathsf{Pos}\,\mathsf{Id}\,\mathsf{tt} = \top$ $\quad \mathsf{Typ}\,\mathsf{Id}\,\mathsf{tt}\,\mathsf{tt} = \mathsf{tt}$

An alternative construction of opetopic types / sets is as presheaves over the category of opetopes (to be proved!)

Bibliography

[5] N. Gambino and J. Kock, Polynomial functors and polynomial monads, Mathematical Proceedings of the Cambridge Philosophical Society, 154 (1) 2013, pp. 153-192.

[6] J. Kock, Polynomial functors and trees, Int. Math. Res. Notices 2011 (2011), 609-673.

[7] J. Kock, A. Joyal, M. Batanin, and J.-F. Mascari. Polynomial functors and opetopes, Adv. Math. 224 (2010), 2690-2737.

Types Are Internal ∞-Groupoids (Extended Version)

Eric Finster
Cambridge University
Department of Computer Science
ericfinster@gmail.com

Antoine Allioux
Inria & IRIF, Université de Paris
France
antoine.allioux@irif.fr

Matthieu Sozeau
Inria & LS2N, Université de Nantes
France
matthieu.sozeau@inria.fr

https://arxiv.org/abs/2105.00024