towards efficient cubical type theory

Favonia
U of Minnesota
2018/10/11

Scientific Study

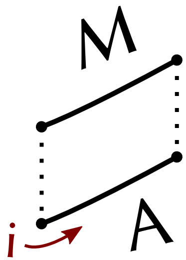
of

efficiency

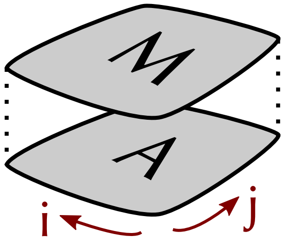
into
the
cubes

⊢ M : A

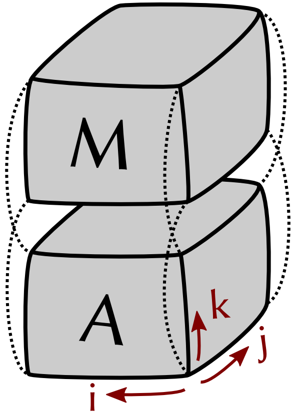
$$i : \mathbb{I} \vdash M : A$$



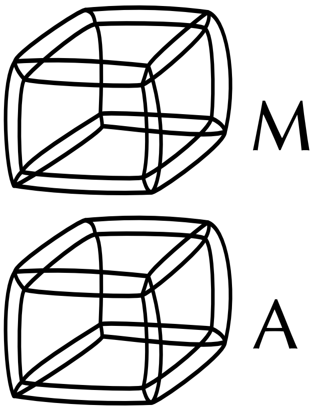
$$\mathbf{i} : \mathbb{I}, \mathbf{j} : \mathbb{I} \vdash M : A$$



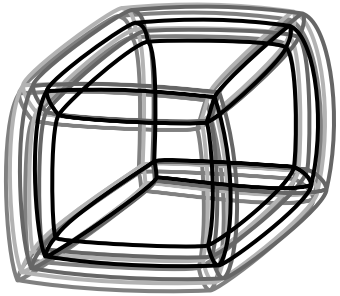
$$i : \mathbb{I},\ j : \mathbb{I},\ k : \mathbb{I} \vdash M : A$$



$$\mathsf{i} : \mathbb{I}, \mathsf{j} : \mathbb{I}, \mathsf{k} : \mathbb{I}, \mathsf{l} : \mathbb{I} \vdash M : A$$

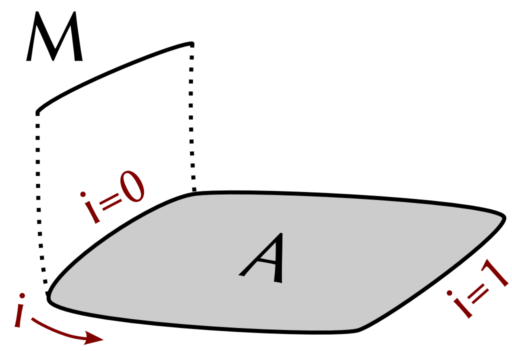


$$i_1 : \mathbb{I}, \ldots, i_n : \mathbb{I} \vdash M : A$$

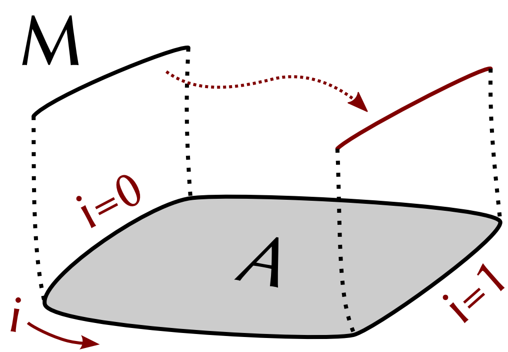


n-cube

Kan filling/ composition structure

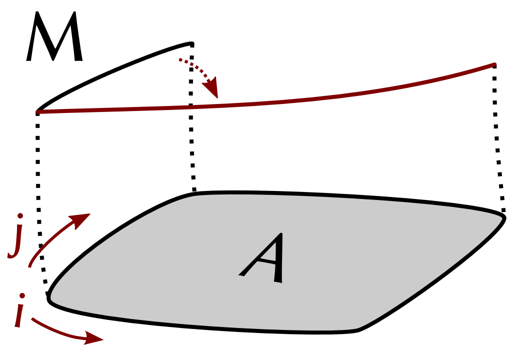


coercion/transport



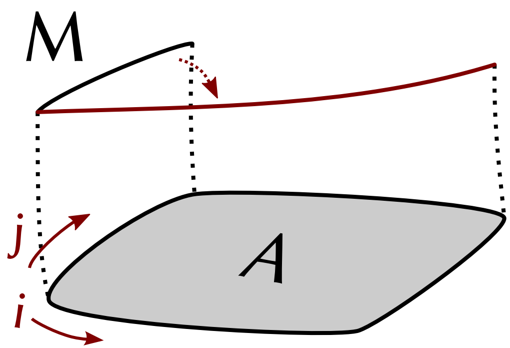
$$\mathrm{coe}^{0\rightsquigarrow 1}[i.A](M) : A[1/i]$$

coercion/transport



$\mathrm{coe}^{0 \rightsquigarrow j}[i.A](M) : A[j/i]$

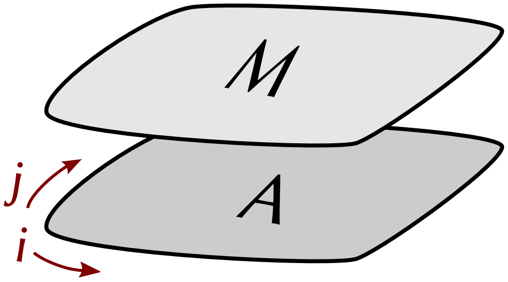
coercion/transport



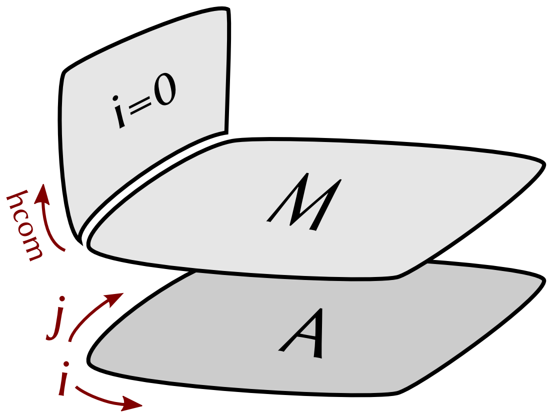
$$\mathsf{coe}^{0 \rightsquigarrow j}[i.A](M) : A[j/i]$$

$r \rightsquigarrow r'$

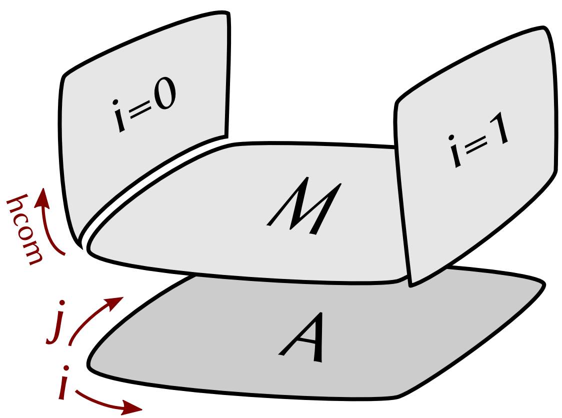
coercion/transport



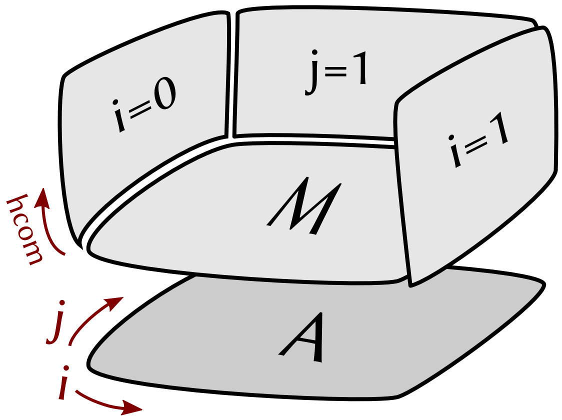
homogeneous composition



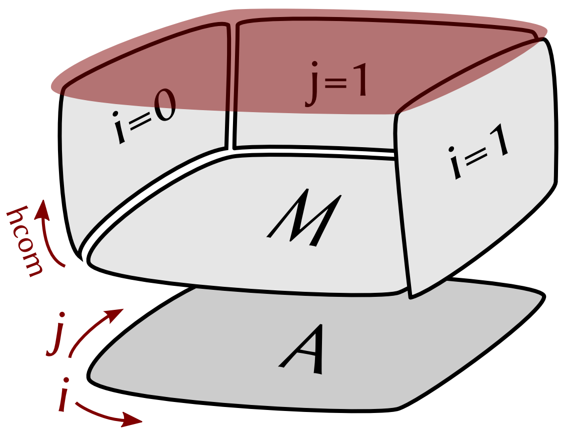
homogeneous composition



homogeneous composition

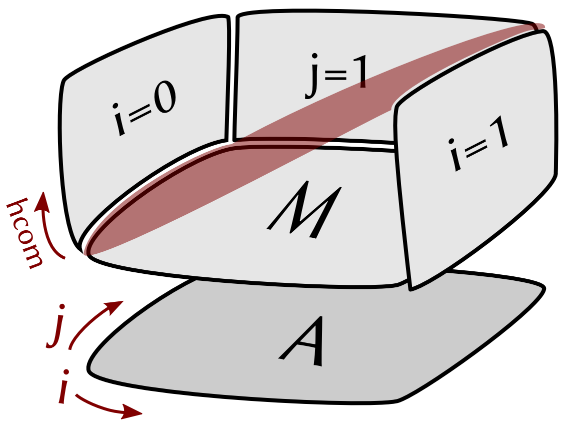


homogeneous composition



$\mathrm{hcom}^{0 \rightsquigarrow 1}[A](M)$
$[i{=}0 \hookrightarrow \ldots, i{=}1 \hookrightarrow \ldots, j{=}1 \hookrightarrow \ldots] : A$

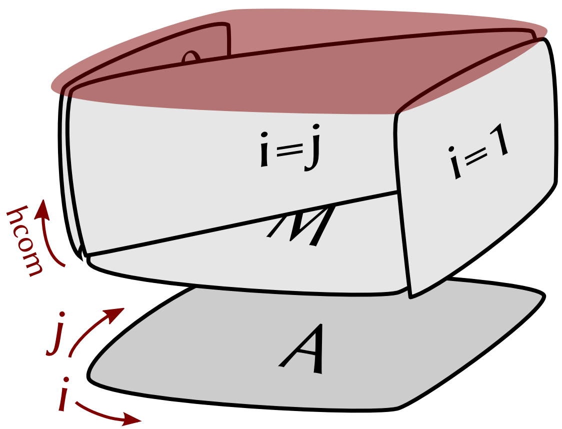
homogeneous composition



$\mathrm{hcom}^{0 \rightsquigarrow i}[A](M)$

$[i{=}0 \hookrightarrow \ldots,\ i{=}1 \hookrightarrow \ldots,\ j{=}1 \hookrightarrow \ldots] : A$

homogeneous composition



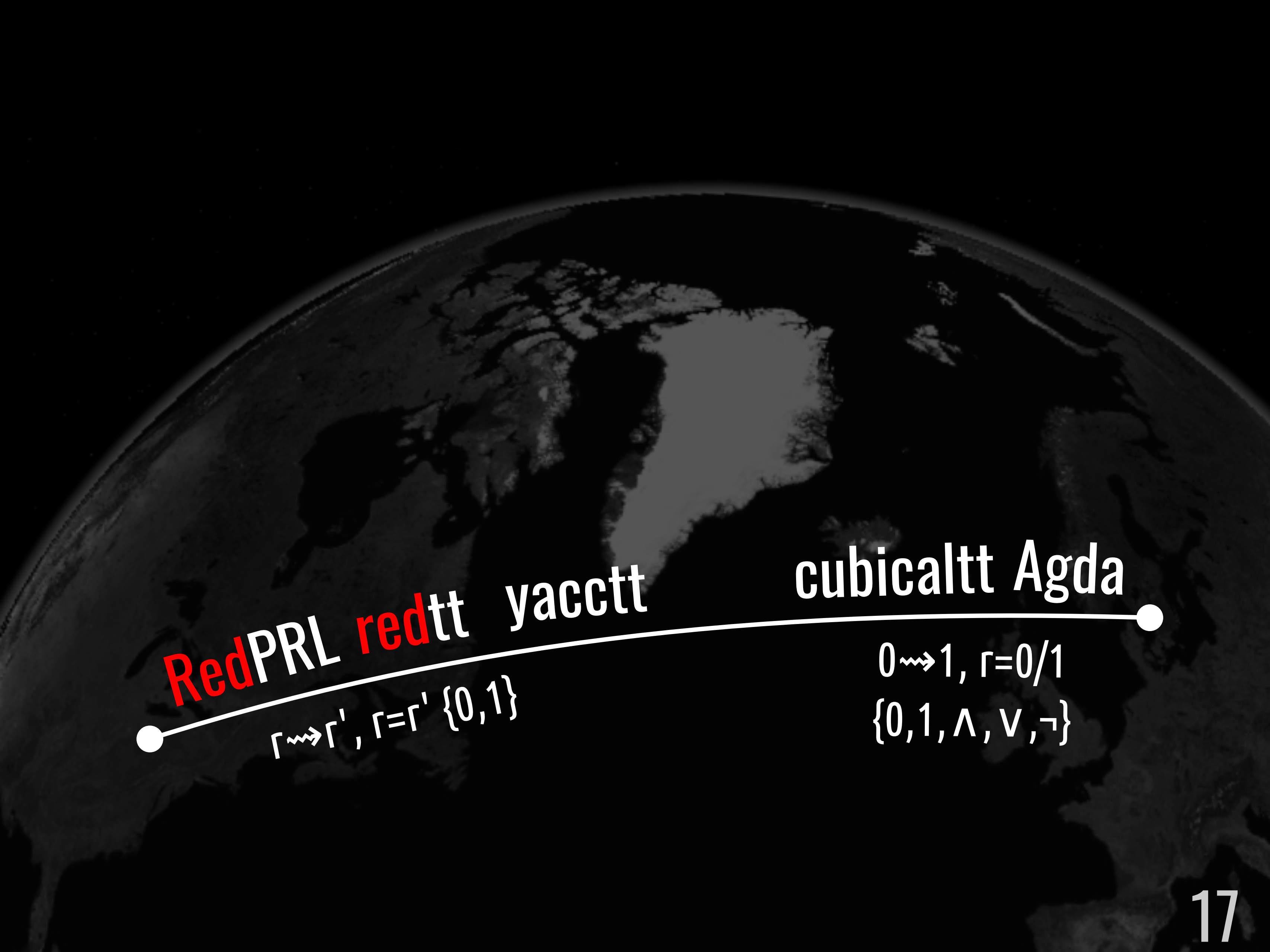
$\text{hcom}^{0 \rightsquigarrow 1}[A](M)$
$[i=0 \hookrightarrow \ldots, i=1 \hookrightarrow \ldots, i=j \hookrightarrow \ldots] : A$

homogeneous composition

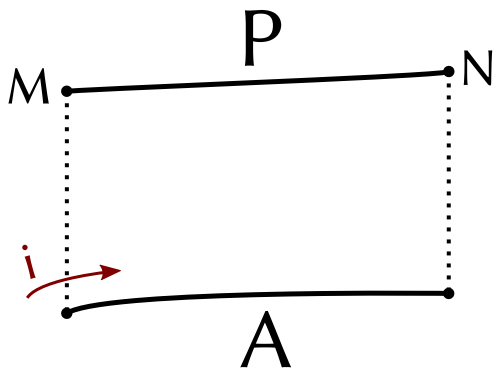
with the power of cubes

univalence and higher indexed inductive types with canonicity⚠

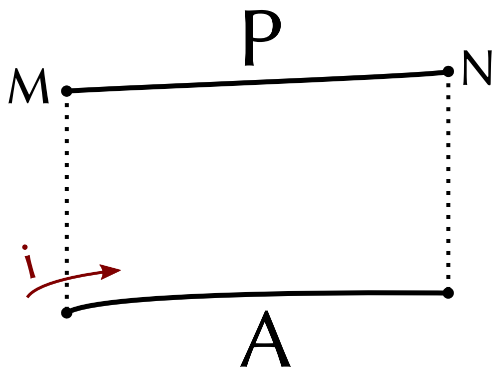
[CCHM, AFH, ABCFHL, CHM, Cavallo & Harper]
see also Coquand's notes


RedPRL redtt yacctt
r⇝r', r=r' {0,1}
cubicaltt Agda
0⇝1, r=0/1
{0,1,∧,∨,¬}

extension types



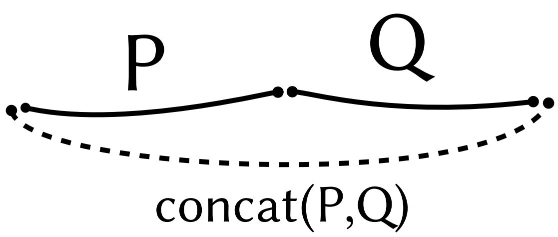
<i>P : Path[i.A](M,N)



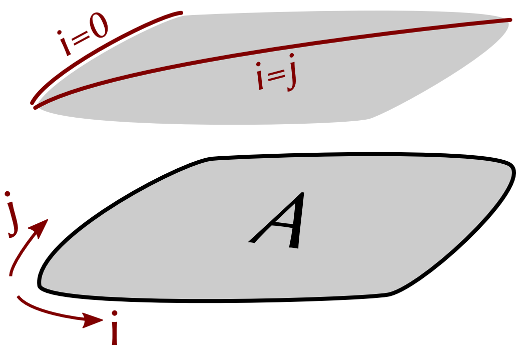
<i>P : [i] A [i=0 ↪ M, i=1 ↪ N]

extension types

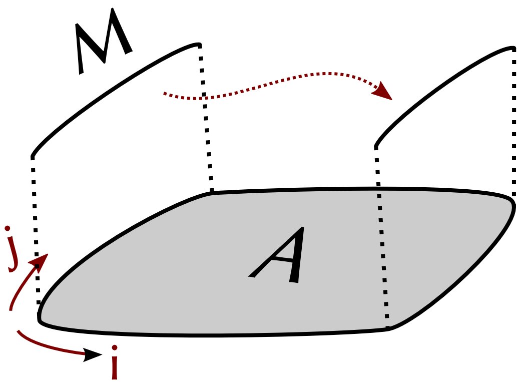
[Shulman & Riehl]



$$(P : [i]\ A\ [\,]) \longrightarrow$$
$$(Q : [i]\ A\ [i{=}0 \hookrightarrow P\ 1]) \longrightarrow$$
$$[i]\ A\ [i{=}0 \hookrightarrow P\ 0,\ i{=}1 \hookrightarrow Q\ 1]$$



[i j] A [i=0 ↪..., i=j ↪...]

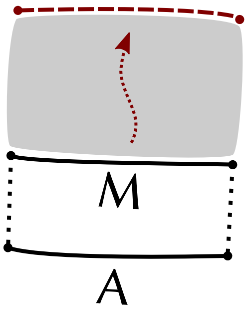


coe[i.[j]A[]](<j>M)

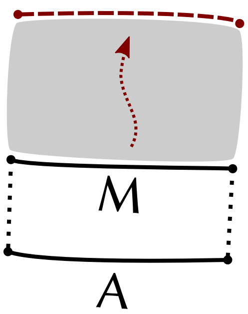
= <j> coe[i.A](M)

fewer fixers, fewer fixes

empty systems



hcom[A](M)[]

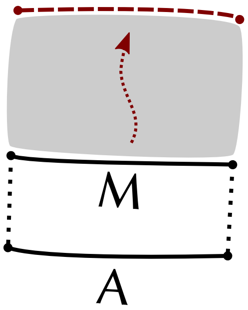


hcom[A](M)[]

= M with regularity

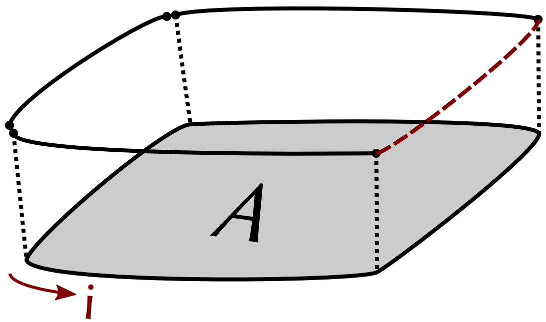
easy to have regularity without univalent Kan universes & HITs

see summary in [Swan] 1808.00920



why do we have empty systems?

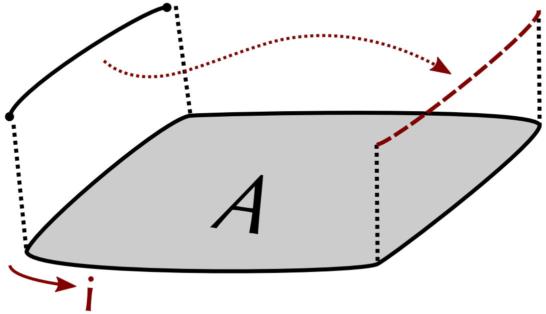
- the lack of coe (in some variants)
- "∀" operator (in some variants)



com[i.A]

⇕

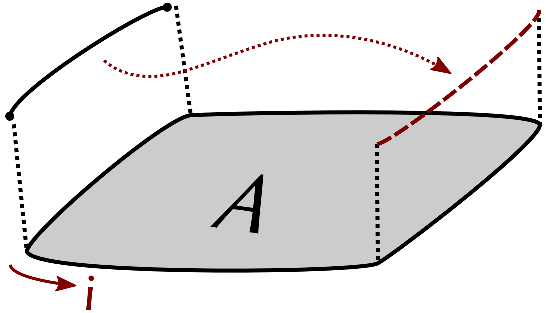
coe[i.A] + hcom[A]



com[i.A](M)[]

coercion without coe

~~coe[i.A] + hcom[A]~~



separating coe and hcom

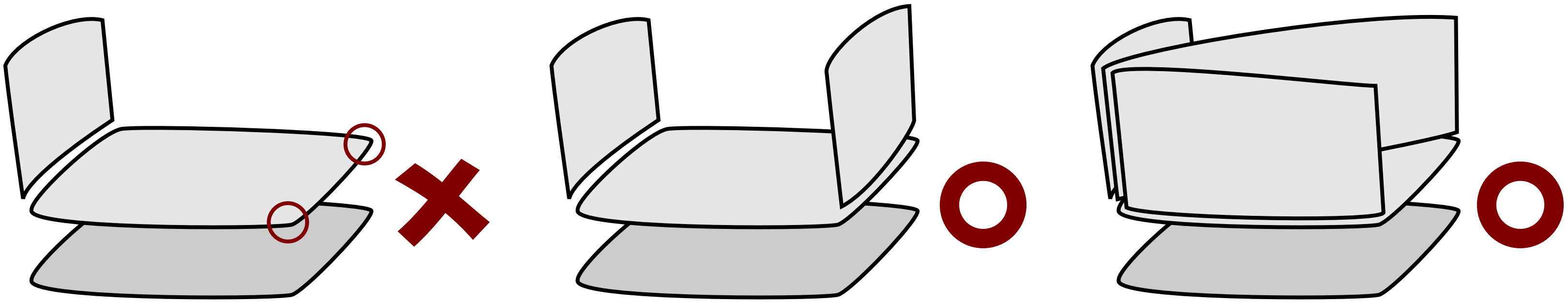
- makes HITs possible and
- kills a major source of empty systems

kill empty systems *completely?*

restrict shapes of hcom to cofibrations that are, equivalently,

- [geometry] covering every point; or
- [syntax] *true* under all closed substitutions; or
- [topos] in $\{ \varphi \in \mathrm{Cof} \mid \neg\neg [\![\varphi]\!] \}$

thanks to Christian Sattler for the topos formulation

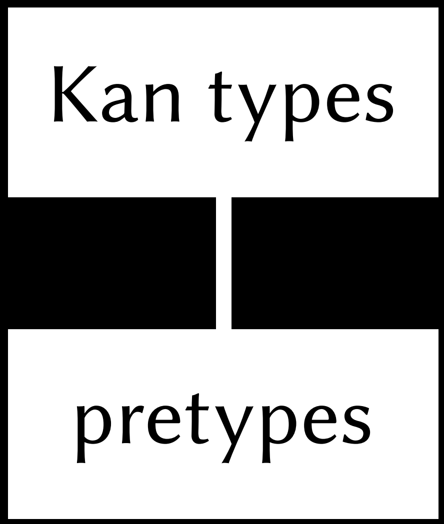


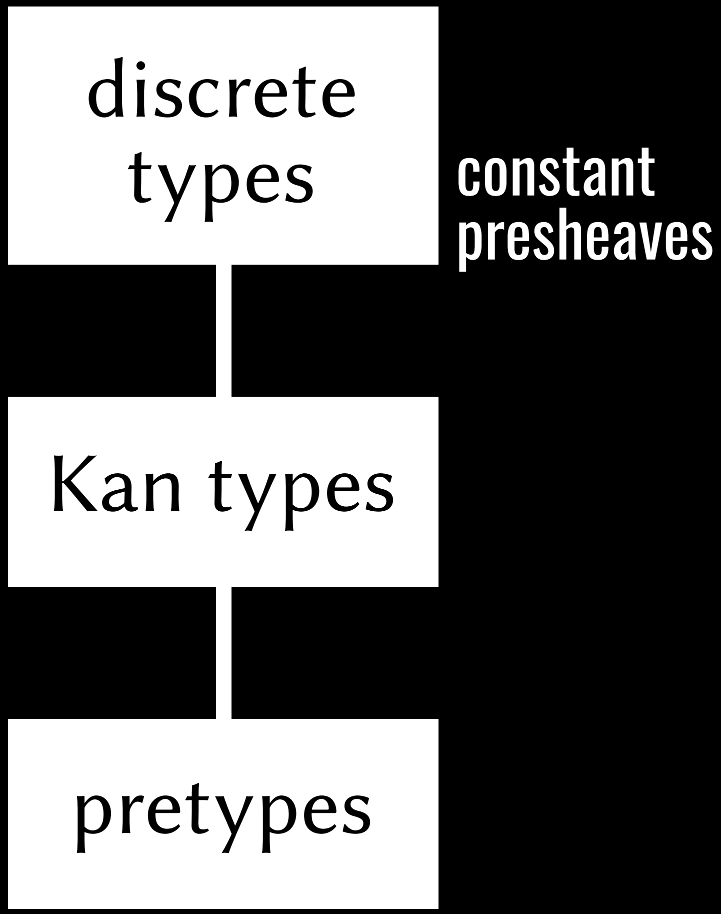
- variants based on cartesian cubes: CHTT [AFH,CH], RedPRL, redtt, ...
- variants based on de morgan cubes: maybe? ask Andrea Vezzosi

difficulty: still need to handle arbitrary cofibrations (due to "∀")
open: generality? is the extra complexity worth it?

kind

semilattices


Kan types
pretypes

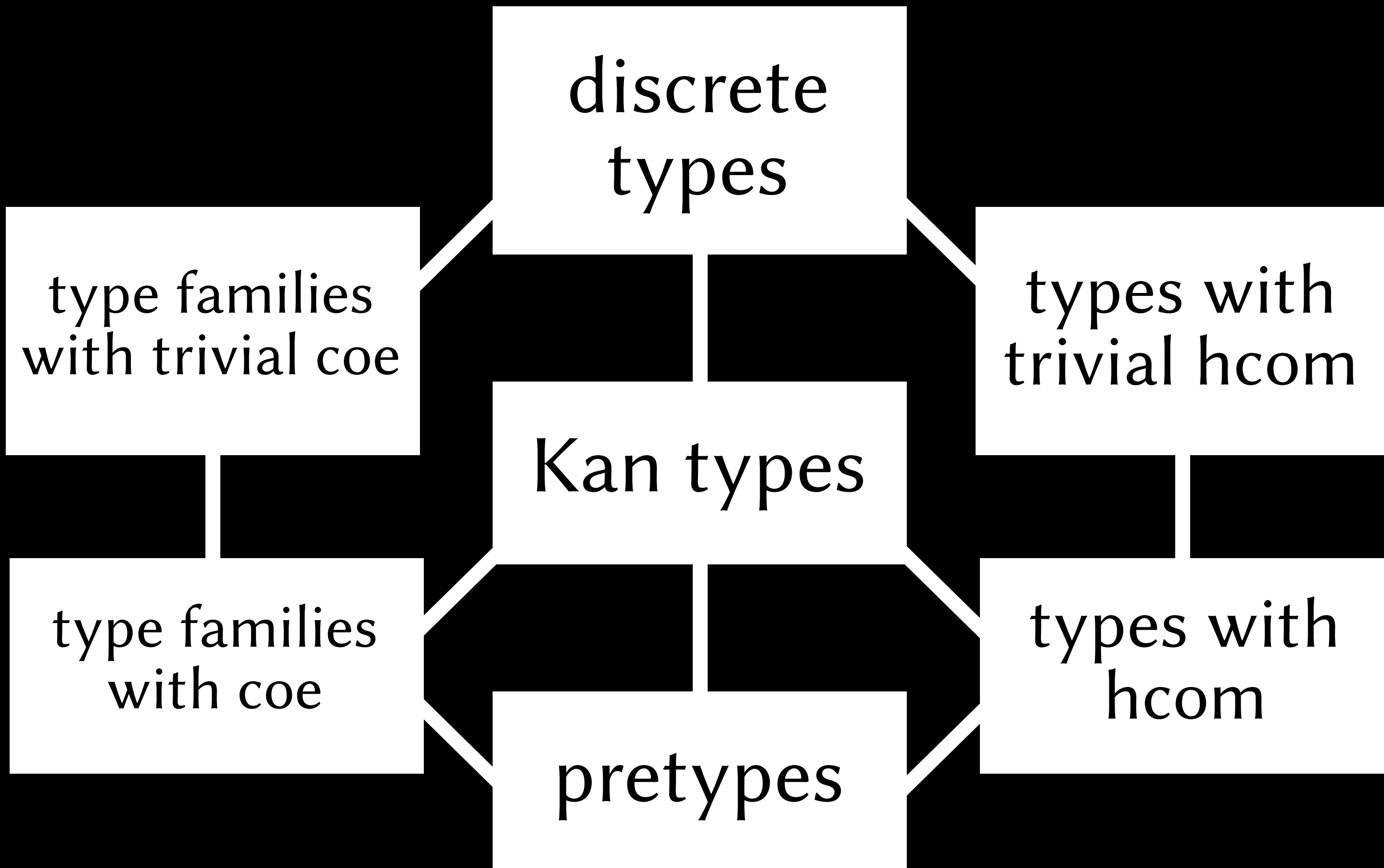

discrete types
constant presheaves
Kan types
pretypes

discrete types

the *entire* “ETT”, including equality types, can be embedded while coexisting with other cubical features

Kan types

pretypes



more can be added; ask Evan Cavallo about trivial coe/hcom

kinds

automatic association of structure or properties with (families of) types (*cf.* the [LOPS] style)

needs a meet semilattice; better if it is Heyting

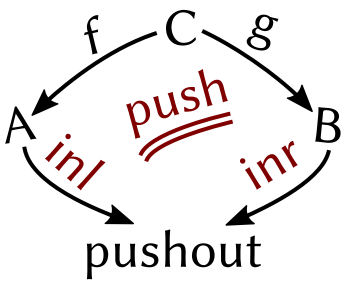
kinds

if A : U_{k_1} , A : U_{k_2} , ..., A : U_{k_n} , then A : U_{k^*}? $\text{meet}_i(k_i) \leq k^*$

what's missing from A : U_k to reach A : U_{k^*}? $k \longrightarrow k^*$

kinds +

higher inductive types




```
data pushout where
| inl (a : A)
| inr (b : B)
| push (i : 𝕀) (c : C) [i=0 ↪ inl (f c), i=1 ↪ inr (g c)]
```

coe(inl(a)) = inl(coe(a))

coe(inr(b)) = inr(coe(b))

$\mathrm{coe}(\mathrm{inl}(a)) = \mathrm{inl}(\mathrm{coe}(a))$

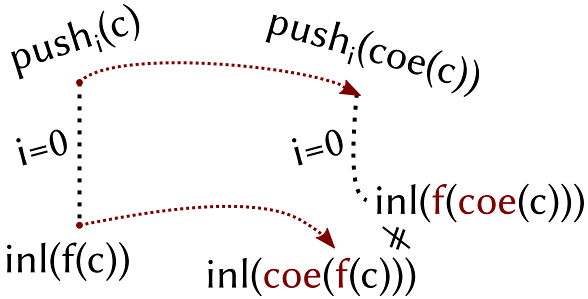
$\mathrm{coe}(\mathrm{inr}(b)) = \mathrm{inr}(\mathrm{coe}(b))$

$\mathrm{coe}(\mathrm{push}_i(c)) \neq \mathrm{push}_i(\mathrm{coe}(c))$

$\mathrm{coe}(\mathrm{inl}(a)) = \mathrm{inl}(\mathrm{coe}(a))$

$\mathrm{coe}(\mathrm{inr}(b)) = \mathrm{inr}(\mathrm{coe}(b))$

$\mathrm{coe}(\mathrm{push}_i(c)) \neq \mathrm{push}_i(\mathrm{coe}(c))$



$\mathrm{coe}(\mathrm{inl}(a)) = \mathrm{inl}(\mathrm{coe}(a))$

$\mathrm{coe}(\mathrm{inr}(b)) = \mathrm{inr}(\mathrm{coe}(b))$

$\mathrm{coe}(\mathrm{push}_i(c)) = \mathrm{hcom}$... *(omitted)*

naive coercion is fine
when f and g are *"clean"* (ex: joins) or
when A and B are *discrete* (ex: suspensions)

ask Evan Cavallo about cleanliness

what's next?

- make great proof *assistants*
- optimize Kan operations of universes
- recover regularity as much as possible
- finish all the meta-theorems