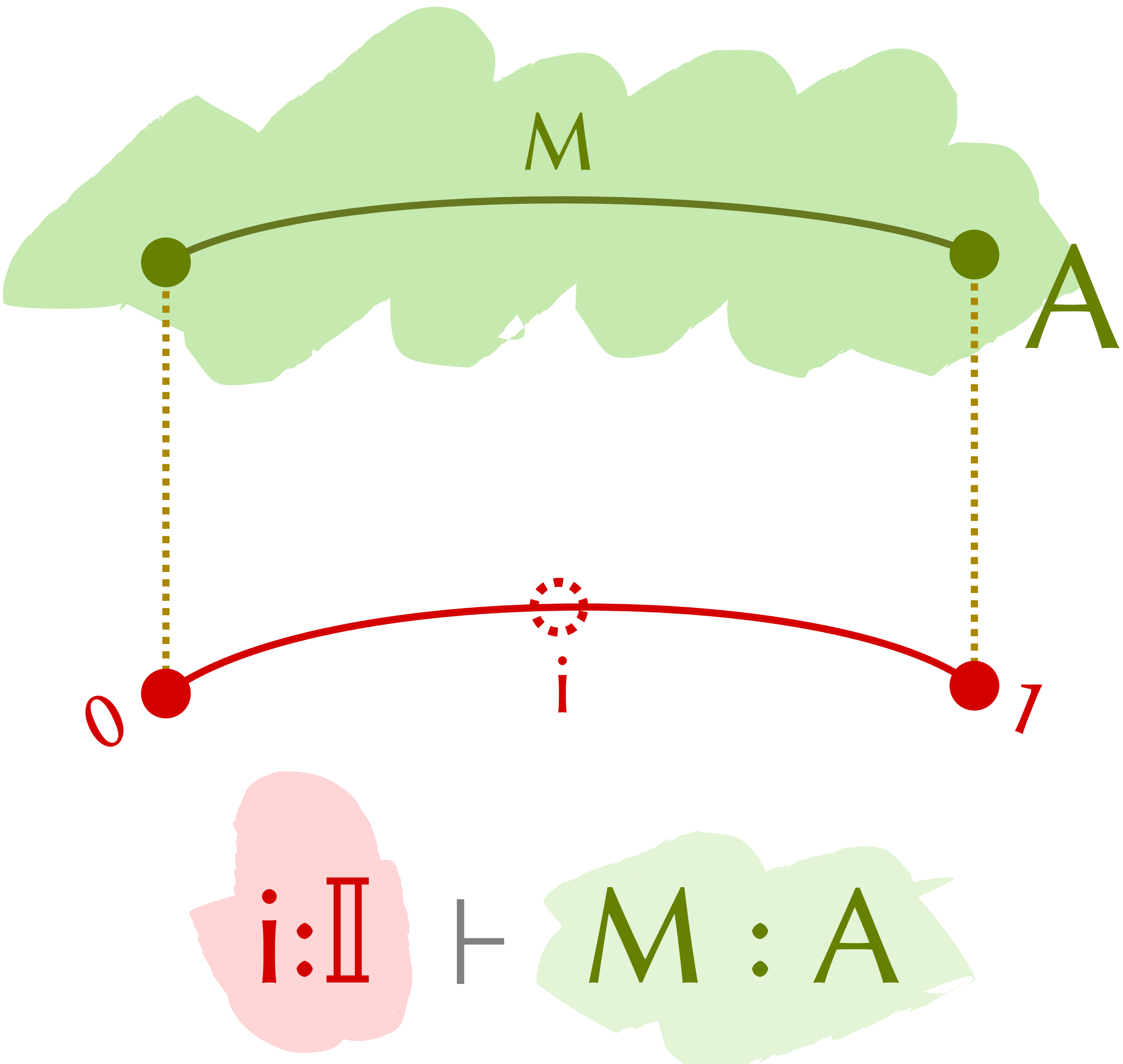
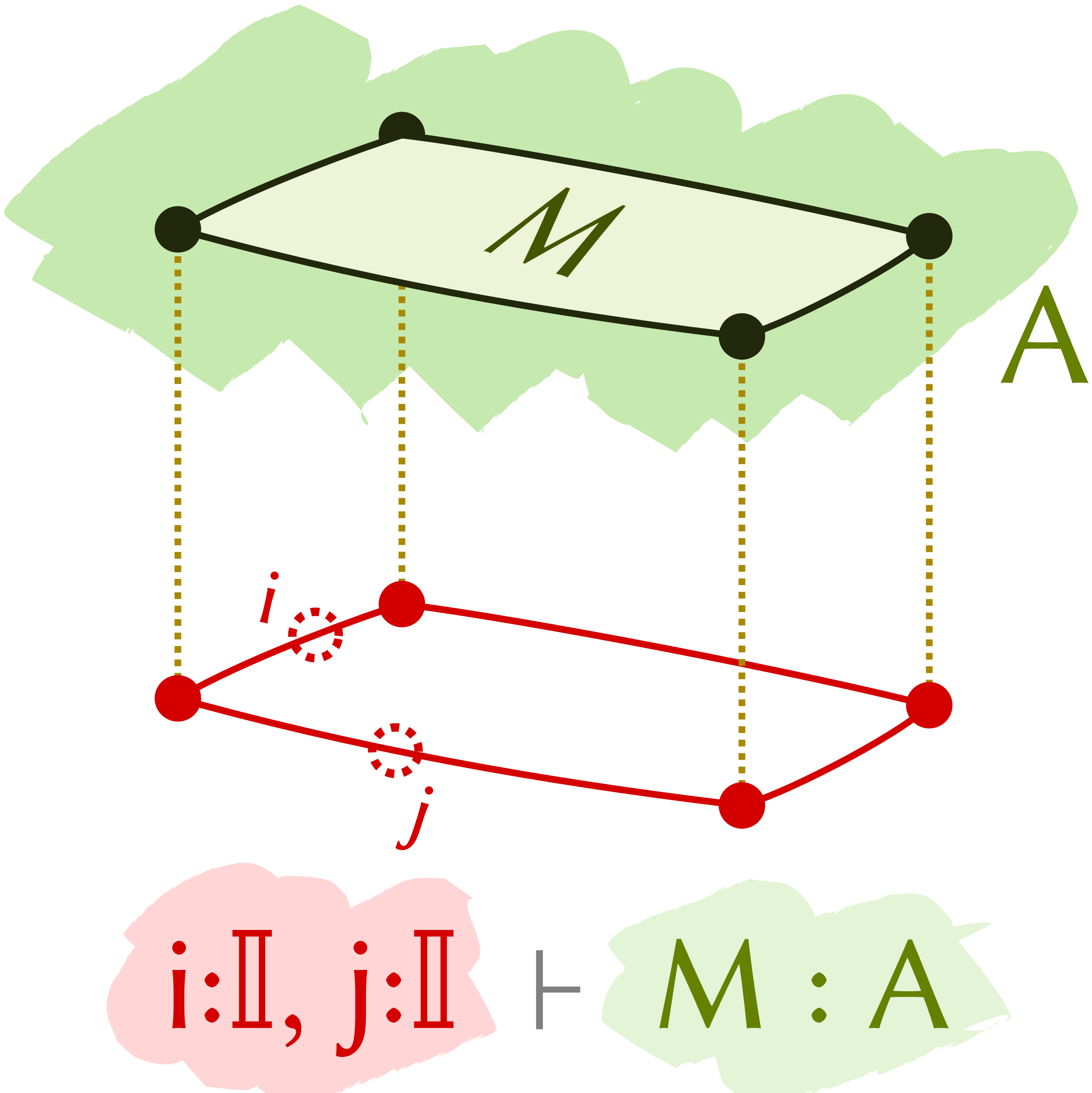
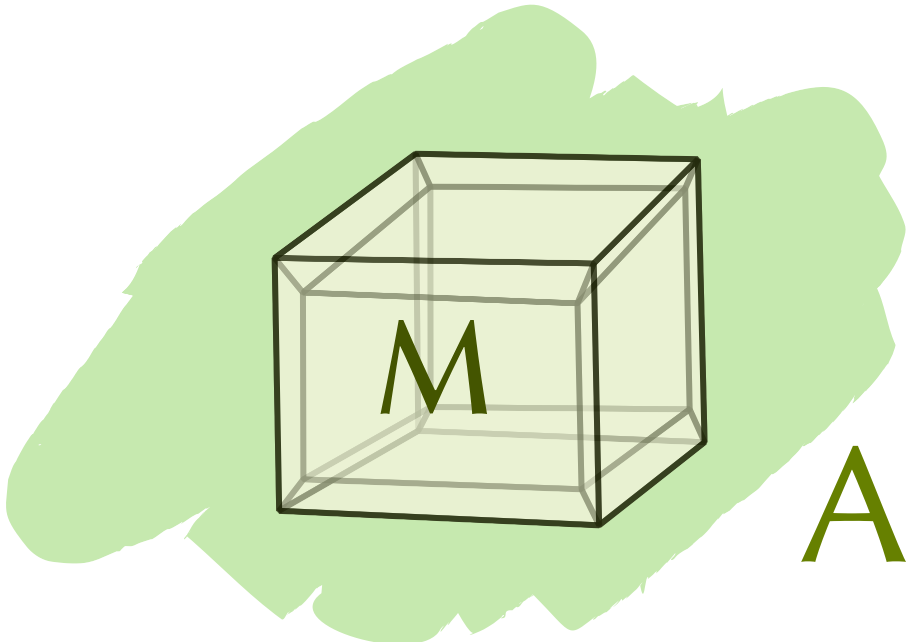
PART I



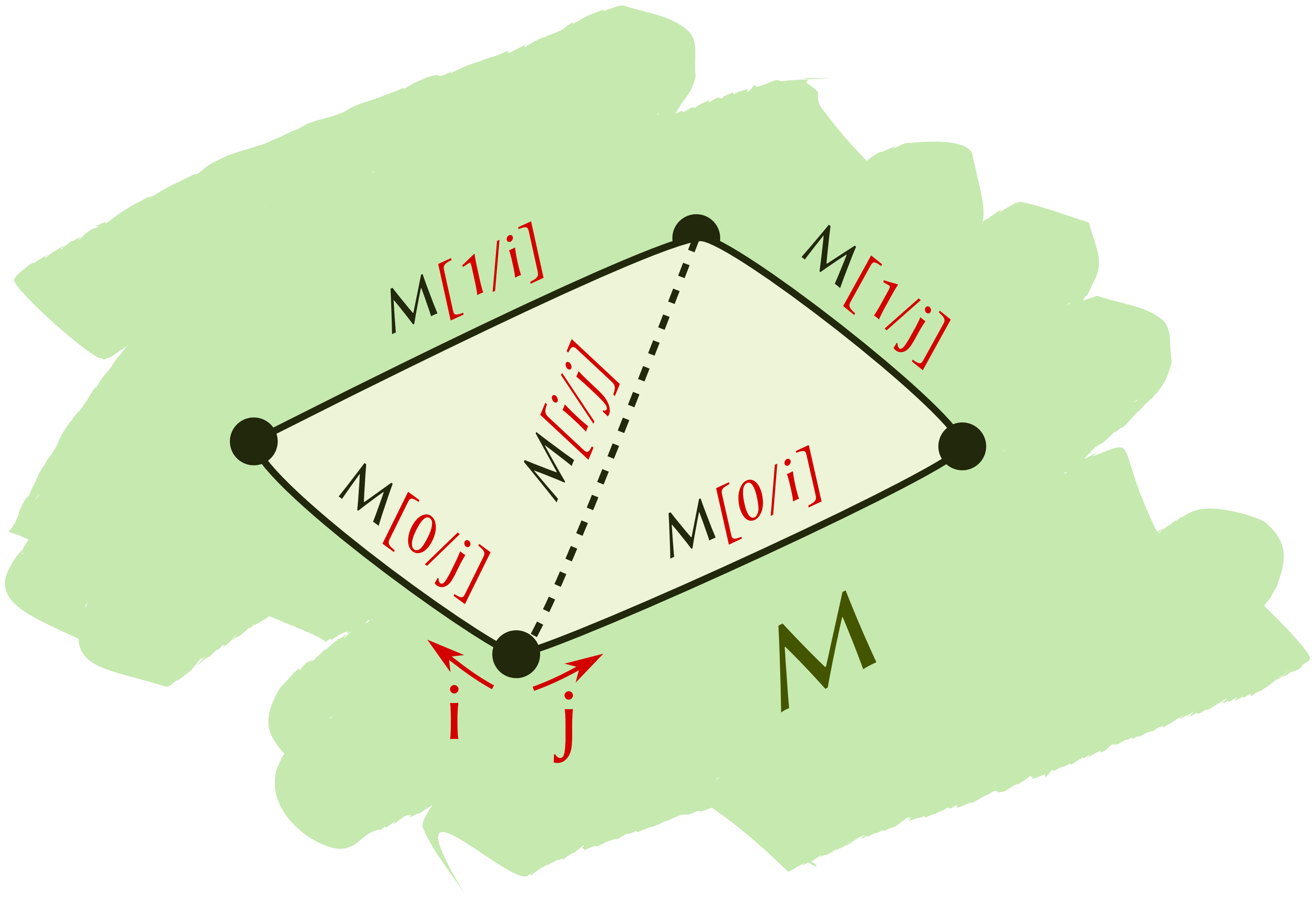
M
A
i
0
1
i:𝕀 ⊢ M : A


M
A
i
j
i:𝕀, j:𝕀 ⊢ M : A



n-cube

$$i_1 : \mathbb{I},\ i_2 : \mathbb{I},\ \ldots,\ i_n : \mathbb{I} \vdash M : A$$

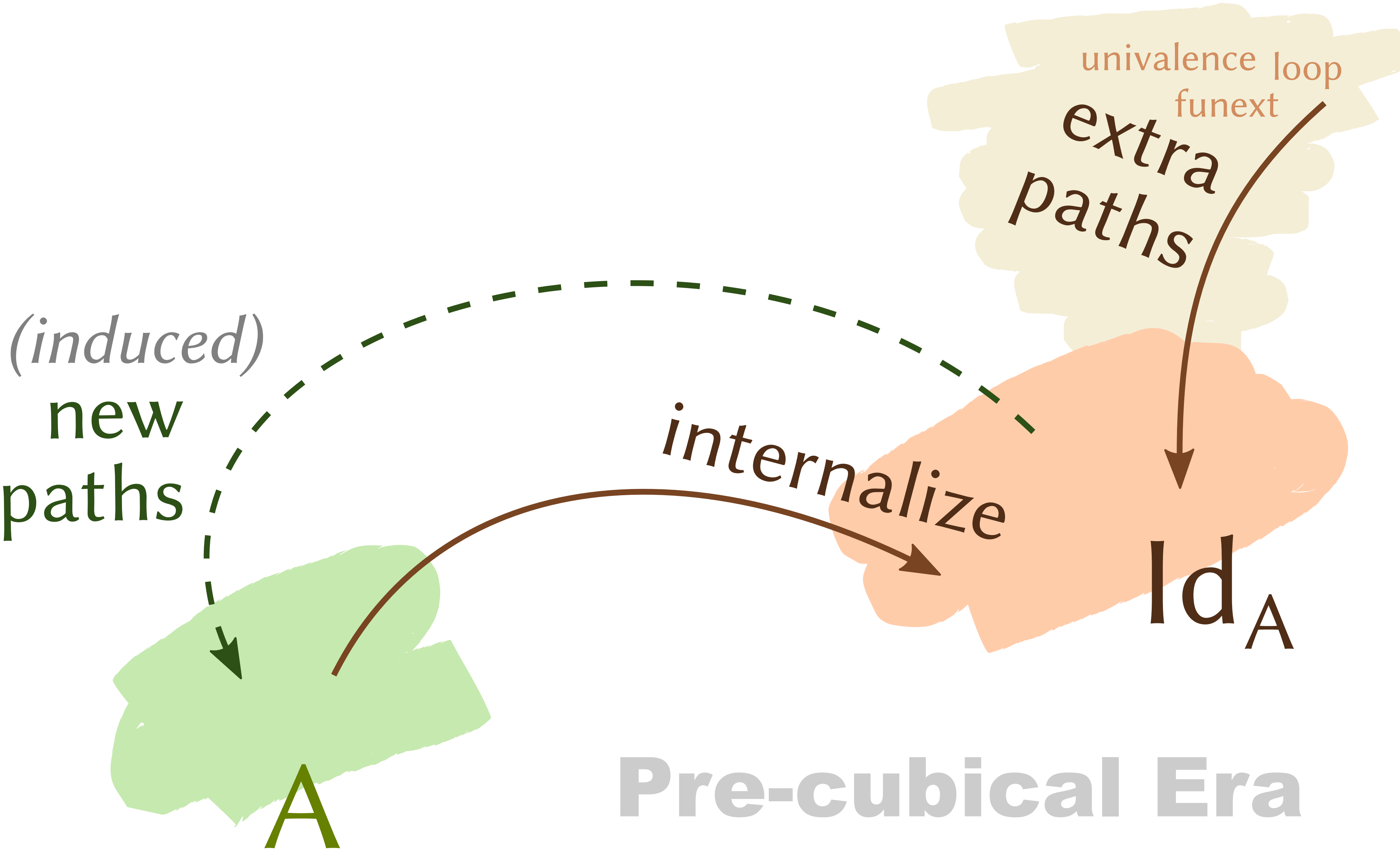

M[1/i]
M[1/j]
M[i/j]
M[0/j]
M[0/i]
i
j
M

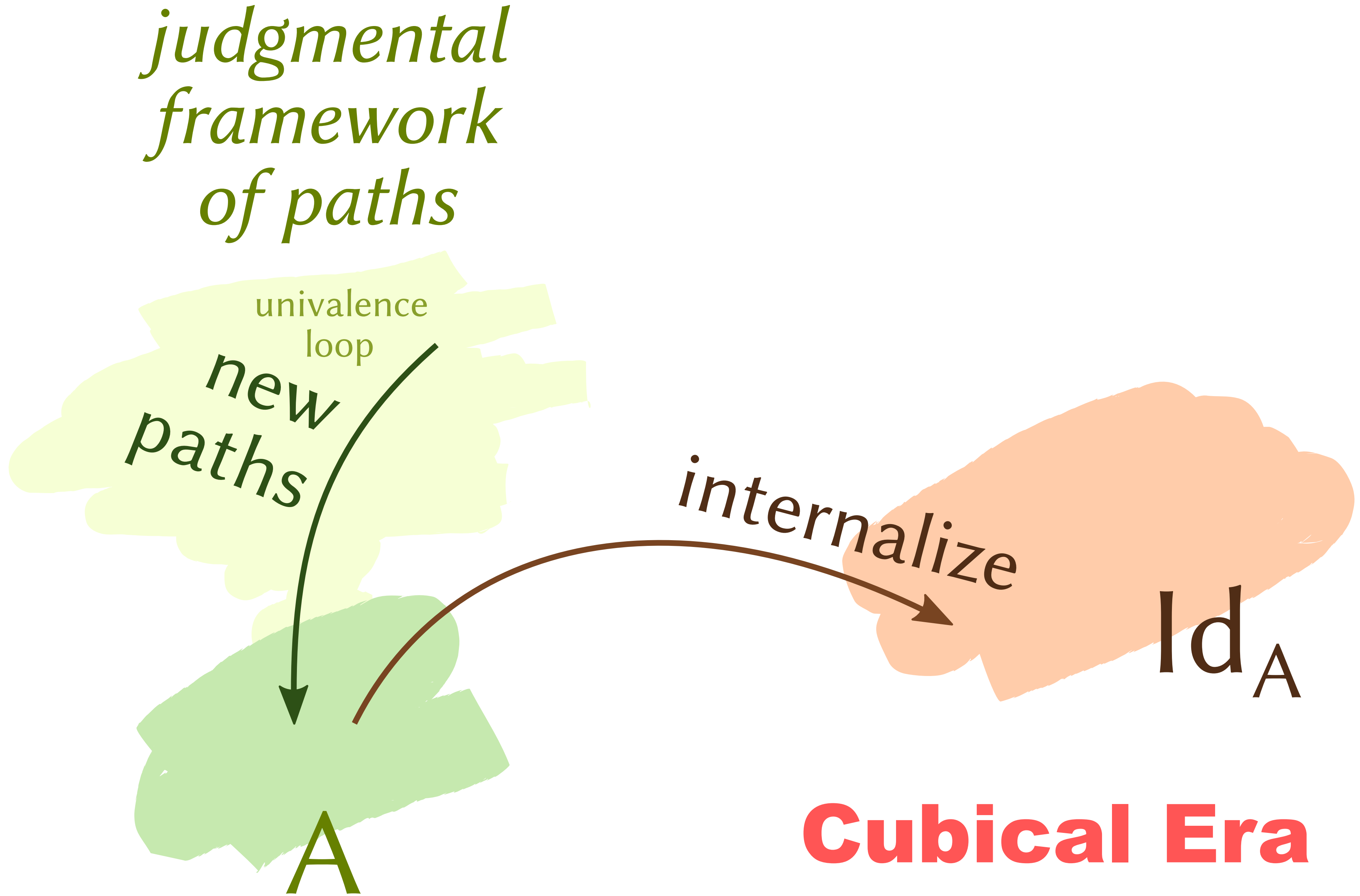
$$\frac{i : \mathbb{I} \in \Gamma}{\Gamma \vdash i : \mathbb{I}} \qquad \frac{}{0 : \mathbb{I}} \qquad \frac{}{1 : \mathbb{I}}$$

$\mathbb{I}$ is not a type!
It is an alian animal

$$\frac{r : \mathbb{I} \quad s : \mathbb{I}}{r \wedge s : \mathbb{I}} \qquad \frac{r : \mathbb{I} \quad s : \mathbb{I}}{r \vee s : \mathbb{I}} \qquad \frac{r : \mathbb{I}}{\sim r : \mathbb{I}}$$

optional


univalence
loop
funext
extra paths
(induced)
new paths
internalize
Id_A
A
Pre-cubical Era


judgmental framework of paths
univalence loop
new paths
A
internalize
Id_A
Cubical Era

$$\frac{}{S1 : U}$$

$$\frac{}{\mathrm{base} : S1}$$

$$\frac{}{\mathrm{loop} : \mathrm{Id}_{S1}(\mathrm{base}; \mathrm{base})}$$

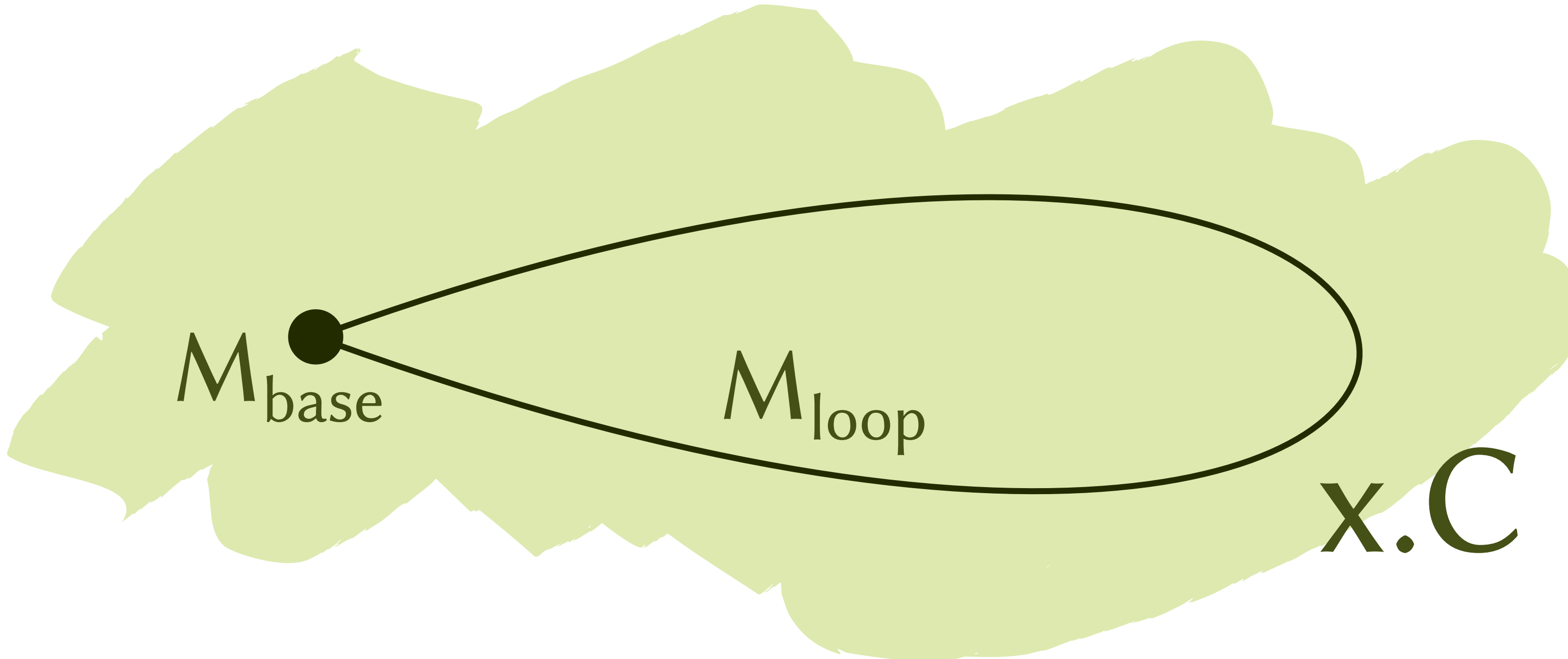
$$\frac{r : \mathbb{I}}{\mathrm{loop}_r : S1}$$

$$\frac{}{\mathrm{loop}_0 \equiv \mathrm{base} : S1}$$

$$\frac{}{\mathrm{loop}_1 \equiv \mathrm{base} : S1}$$

$$\frac{x{:}\,S1 \vdash C : U \qquad M_{base} : C[base/x] \qquad N{:}\,S1 \qquad M_{loop} : M_{base} =^{x.C}_{loop} M_{base}}{\mathrm{elim}_{S1}[x.C](M_{base}; M_{loop}; N) : C[N/x]}$$

$$\frac{x{:}\,S1 \vdash C : U \qquad M_{base} : C[base/x] \qquad N{:}\,S1 \qquad i{:}\mathbb{I} \vdash M_{loop} : C[loop_i/x] \qquad M_{loop}[0/i] \equiv M_{base} : C[base/x] \qquad M_{loop}[1/i] \equiv M_{base} : C[base/x]}{\mathrm{elim}_{S1}[x.C](M_{base}; i.M_{loop}; N) : C[N/x]}$$



$$M_{base} : C[base/x] \qquad base : S1$$

$$i{:}\mathbb{I} \vdash M_{loop} : C[loop_i/x] \qquad loop_r : S1$$

$$M_{loop}[0/i] \equiv M_{base} : C[base/x] \qquad loop_0 \equiv base : S1$$

$$M_{loop}[1/i] \equiv M_{base} : C[base/x] \qquad loop_1 \equiv base : S1$$

$$elim_{S1}[x.C](M_{base}; i.M_{loop}; N) : C[N/x]$$

$(\ldots)$

$$\frac{(\ldots)}{\mathrm{elim}_{S1}[x.C](M_{base};\ j.M_{loop};\ \mathrm{base}) \equiv M_{base} : C[\mathrm{base}/x]}$$

$(\ldots)$

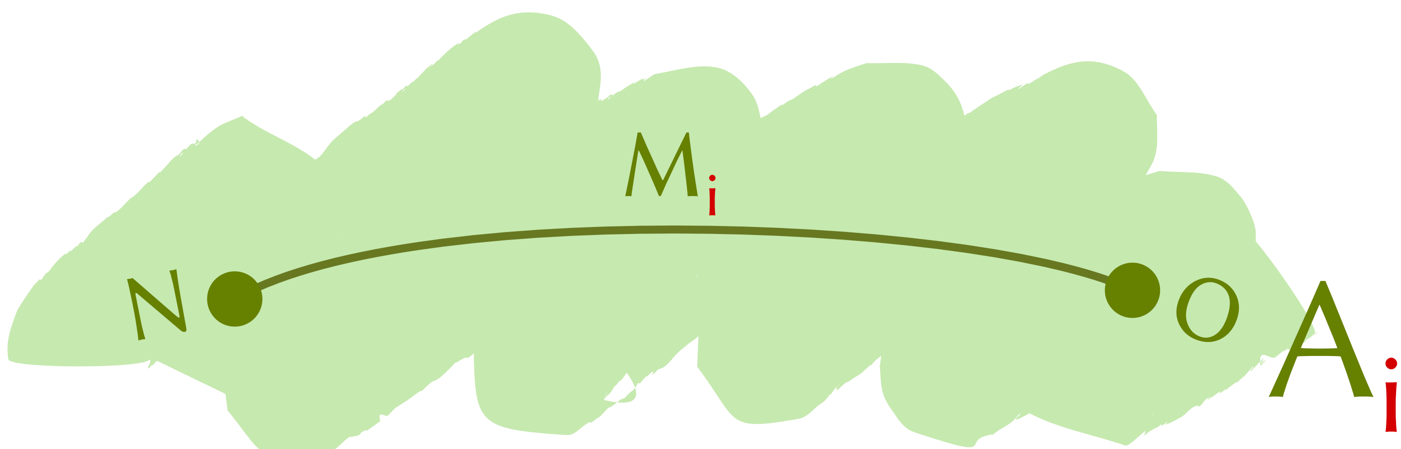
$$\frac{(\ldots)}{\mathrm{elim}_{S1}[x.C](M_{base};\ j.M_{loop};\ \mathrm{loop}_i) \equiv M_{loop}[i/j] : C[\mathrm{loop}_i/x]}$$

1. diagonal substitution is crucial
2. we need to improve the framework to enable path concatenation

   e.g., "loop · loop"

Cubical **Pre-cubical**

Path types "$\Pi_{i:\mathbb{I}}$ A" with endpoints



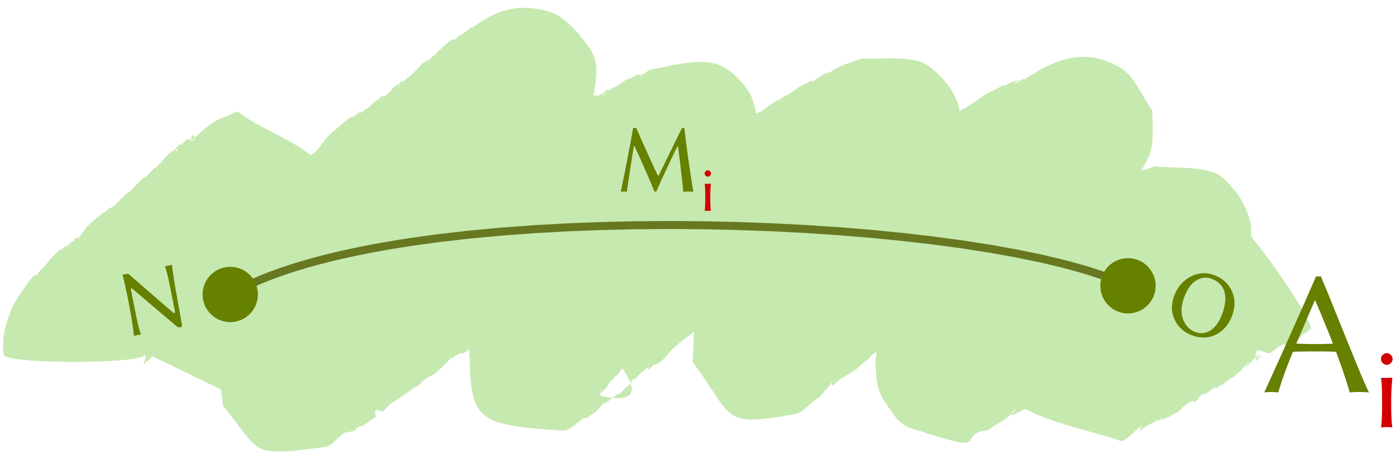
$$\frac{i{:}\mathbb{I} \vdash A : U \qquad N : A[0/i] \qquad O : A[1/i]}{\mathrm{Path}_{i.A}(N; O) : U}$$

$$\frac{i{:}\mathbb{I} \vdash M : A \qquad M[0/i] \equiv N : A[0/i] \qquad M[1/i] \equiv O : A[1/i]}{\lambda i.M : \mathrm{Path}_{i.A}(N; O)}$$

$$\frac{P : \mathrm{Path}_{i.A}(N; O) \qquad r : \mathbb{I}}{P@r : A[r/i]}$$

$$\frac{P : \mathrm{Path}_{i.A}(N; O)}{P@0 \equiv N : A[0/i] \qquad P@1 \equiv O : A[1/i]}$$

Path types "$\Pi_{i:\mathbb{I}}$ A" with endpoints



$$\frac{i{:}\mathbb{I} \vdash M : A \qquad r : \mathbb{I}}{(\lambda i.M)@r \equiv M[r/i] : A[r/i]}$$

$$\frac{P : \mathrm{Path}_{i.A}(N; O)}{P \equiv \lambda i.P@i : \mathrm{Path}_{i.A}(N; O)}$$

Path types

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

Identification types

freely generated by **refl**

these two will become equivalent when we fix the framework (next week)

Where is function extensionality?

YES, WE CAN!

(NOT VIA UNIVALENCE)

Function extensionality

$$P : \Pi_{x:A}\mathrm{Path}_{\_.B(x)}(F(x); G(x))$$

$$x{:}A \vdash P(x) : \mathrm{Path}_{\_.B(x)}(F(x); G(x))$$

$$x{:}A, i{:}\mathbb{I} \vdash P(x)@i : B(x)$$

$$i{:}\mathbb{I}, x{:}A \vdash P(x)@i : B(x)$$

$$i{:}\mathbb{I} \vdash \lambda x.P(x)@i : B(x)$$

$$\lambda i.\lambda x.P(x)@i : \mathrm{Path}_{\_.\Pi(x:A)B(x)}(F; G)$$

(you need to check the boundaries F and G)

$$x{:}A,\ i{:}\mathbb{I} \vdash P(x)@i : B(x)$$

$$i{:}\mathbb{I},\ x{:}A \vdash P(x)@i : B(x)$$

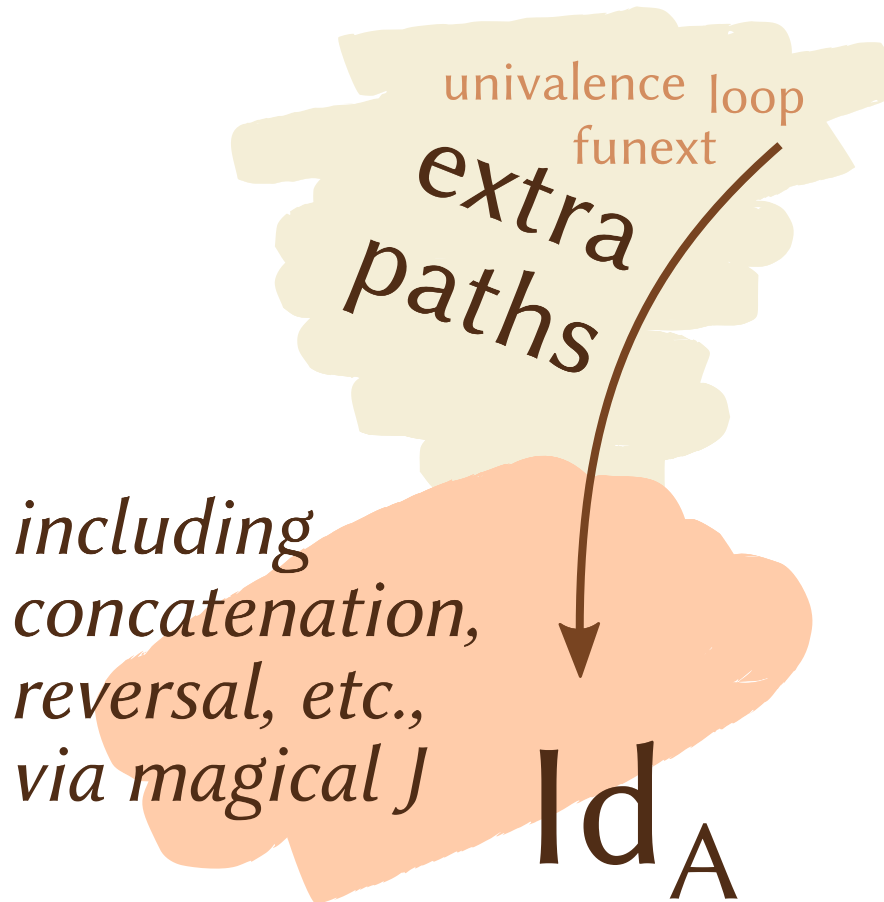
1. Both paths and functions internalize hypothetical judgments
2. You can exchange hypotheses
3. Paths and functions thus "commute"
4. Therefore, function extensionality!

Fix the framework (next week)

1. What are the types? (form)
2. What are the constructors? (intro)
3. How to consume an element? (elim)
4. What if a constructor is consumed? (β)
5. Uniqueness principle? (η)
6. How to compose stuff? (Kan operators)

every type is responsible for its path concatenation

Cubical



Pre-cubical