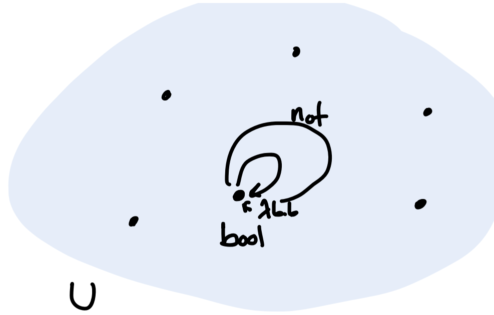
Homotopy levels



We want to say things like "U is not a set".

- A set is something whose =-types don't have structure.

Def. A type T's <u>h-level is 0</u> if

$$\text{hlevel } 0\, T := \sum_{t:T} \prod_{s:T} s =_T t.$$

A type T's <u>h-level</u> is $\mathrm{s}n$ if

$$\text{hlevel } \mathrm{s}n\, T := \prod_{s,t:T} \text{hlevel } n\; s=t.$$

We've defined a function $\text{hlevel} : \mathbb{N} \to \text{Type} \to \text{Type}$.

<u>h-level 0</u>. AKA contractible, isContr

Ex. $\mathbb{1}$ is contractible.

Most boring.

<u>h-level 1</u>. AKA propositions, isProp

Fact. Equivalent to $\prod_{x,y:P} x=y$.

Ex. $\emptyset, \mathbb{1}$ are propositions

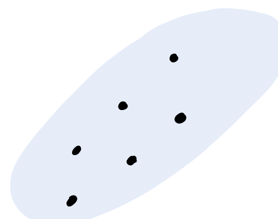
__Ex.__ In fact, any contractible type is a proposition.

__Ex.__ If a proposition is inhabited, it is contractible.

→ So roughly, a proposition is = to $\emptyset$ or $\mathbb{1}$.

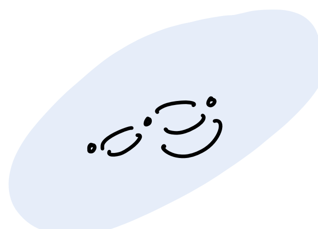
So these behave like logical propositions where Σ behaves like $\exists$, etc.

__h-level 2.__ AKA sets, is Set



__Fact.__ bool, $\mathbb{N}$ are propositions

__h-level 3.__ AKA groupoids



__Fact.__ Type has h-level at least 3.

__Ex.__ If a type T has h-level n, then it has h-level $n+1$.

__Equivalences__

Sometimes we want types to be propositions (no structure). Sometimes we're interested in structure.

Given $f : A \to B$, want a __proposition__ isEquiv(f).

The type $\sum_{g:B\to A} fg = 1 \times gf = 1$ is not a proposition.

→ would ask for adjoint equivalence, or equivalently:

Def. A function $f : A \to B$ is an equivalence if:

$$\mathrm{isEquiv}(f) := \prod_{b:B} \mathrm{isContr}\Big(\sum_{a:A} fa = b\Big).$$

$1 =$ fiber

Write

$$A \simeq B := \sum_{f : A \to B} \mathrm{isEquiv}(f).$$

Fact. For every type A, $A \simeq A$, so we can define

$$\mathrm{idtoequiv} : A = B \to A \simeq B.$$

Def. The univalence axiom asserts

$$\mathrm{ua} : \mathrm{isEquiv}(\mathrm{idtoequiv}).$$

Univalence for logic and sets

Def. $\mathrm{Prop} := \sum_{P:\mathrm{Type}} \mathrm{isProp}(P)$ (Since isProp is a proposition $\mathrm{isProp} \subseteq \mathrm{Type}$)

Fact. The univalence axiom implies

$$(P =_{\mathrm{Prop}} Q) \simeq (P \leftrightarrow Q).$$

Def. $\mathrm{Set} := \sum_{S:\mathrm{Type}} \mathrm{isSet}(S)$

Fact. The univalence axiom implies

$$(P =_{Set} Q) \simeq (P \cong Q).$$

Def. $\mathrm{Grp} := \sum_{G:Set} \sum_{e:G} \sum_{m:G\to G\to G} \sum_{i:G\to G} \prod_{x:G} (m(e,x)=x) \times (m(x,e)=x)$
$\times \prod_{x,y,z:G} ((xy)z = x(yz))$
$\times \prod_{x:G} (m(ix,x) = e) \times (m(x,ix)=e).$

Fact. The univalence axiom implies

$$(G =_{Grp} H) \simeq (G \cong H)$$