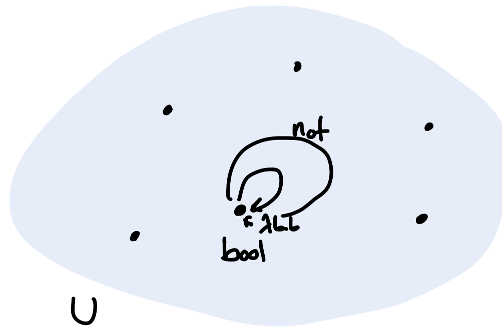
Homotopy levels

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 sn if

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

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

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

<u>Ex</u>. $\mathbb{1}$ is contractible.

Most boring.

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

<u>Fact</u>. Equivalent to $\prod_{x,y:P} x=y$.

<u>Ex</u>. $\emptyset, \mathbb{1}$ are propositions

<u>Ex</u>. 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.

Equivalences

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

Given $f : A \to B$, want a proposition $\mathrm{isEquiv}(f)$.

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

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

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

1 = fiber

Write

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

Ex. Every contractible type is equivalent to $\mathbb{1}$.

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

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

Higher inductive types.

Ex. Given $P, Q : \mathrm{Prop}$

$P + Q$ is not a proposition in general

Def. Given a type T, the propositional truncation $\|T\|_1$ of T is the higher inductive type with constructors

- $|-|_1 : T \to \|T\|_1$
- $\prod_{x,y : \|T\|_1} |x|_1 =_{\|T\|_1} |y|_1$

Def. Define $P \vee Q := \|P + Q\|_1$ for $P, Q : \mathrm{Prop}$.

Ex. Equivalences

Ex. The univalence axiom implies

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

Lem. $(P \leftrightarrow Q)$ is a proposition.

Cor. Prop is a set.

Univalence for logic

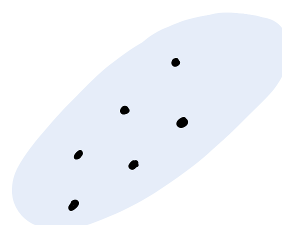
Def. $\mathrm{Prop} := \sum_{P:\mathrm{Type}} \mathrm{isProp}(P)$

Ex. The univalence axiom implies
$$(P =_{\mathrm{Prop}} Q) \simeq (P \leftrightarrow Q).$$

Lem. $(P \leftrightarrow Q)$ is a proposition.

Cor. Prop is a set.

h-level 2. AKA sets, isSet



Fact. bool, $\mathbb{N}$ are sets,

Univalence for sets

Fact. The univalence axiom implies
$$(P =_{\mathrm{Set}} Q) \simeq (P \cong Q).$$

Lem. $P \cong Q$ is a set.

Cor. Set is a groupoid.

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

Q. Why do we ask G to be a set?

Fact. The univalence axiom implies

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

Cor. Grp is a groupoid.

Fact. We have the same univalence principle for any algebraic structure on a set. (Coquand-Danielsson)

Moral: univalence allows us to do mathematics <u>up to the appropriate notion of</u> sameness in a type (in these examples).

→ 'Structure identity principle' (Aczel, Coquand)
→ 'identity of indiscernables' (Leibniz)

Fact. The univalence axiom implies

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

Cor. Grp is a groupoid.

<u>Fact</u>. We have the same univalence principle for any algebraic structure on a set.

Moral: univalence allows us to do mathematics <u>up to the appropriate notion of</u> sameness in a type (in these examples).