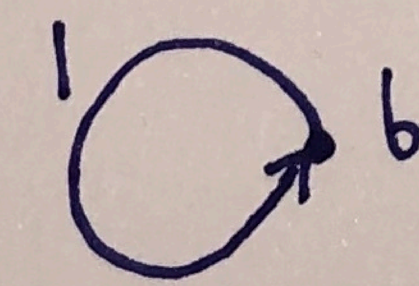
Circle

To do algebra we need: terms $x, y, \dots$
operations $*, \dots$
relations $xy = yx, \dots$

HIT

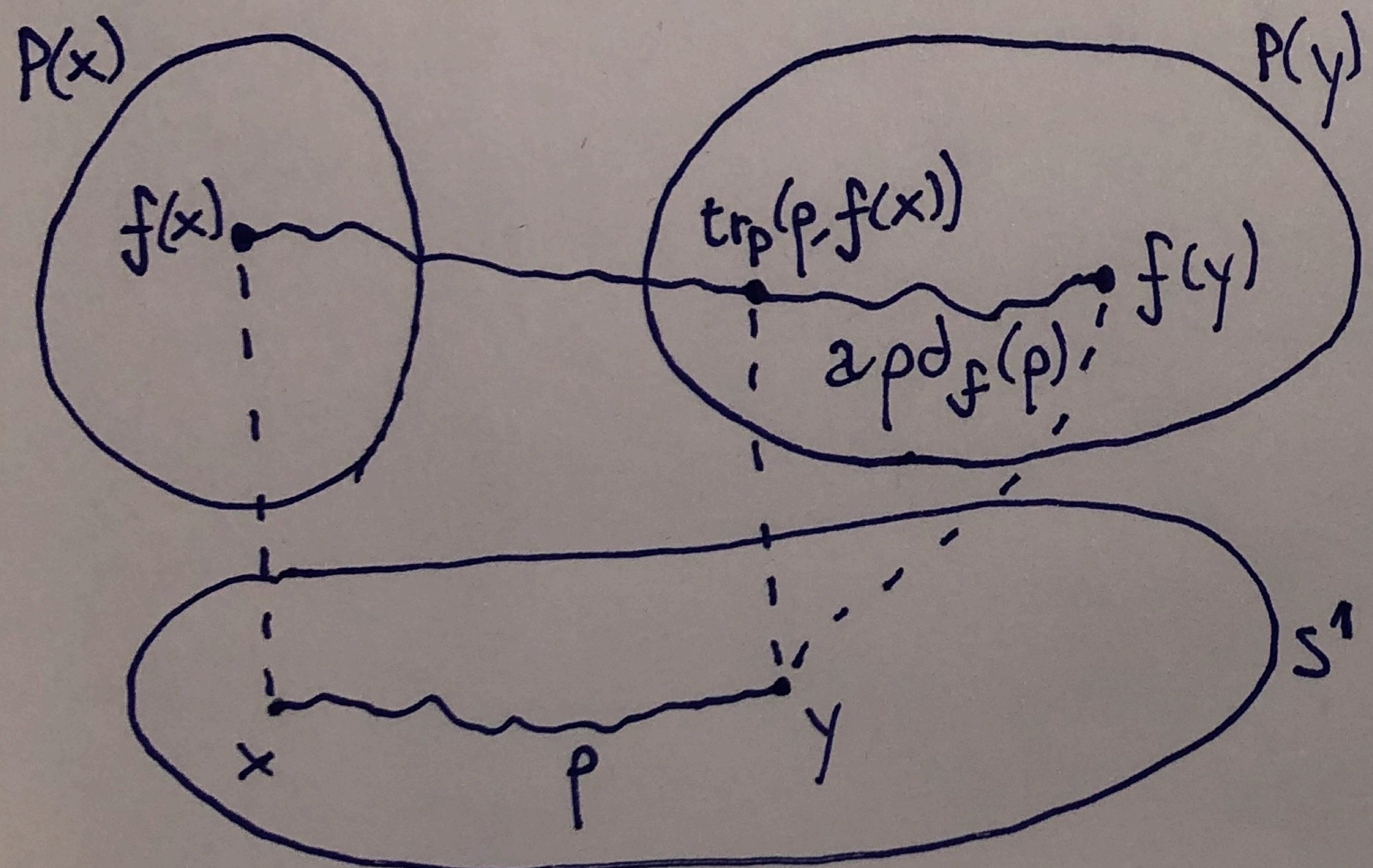
→ We need points and path constructors, i.e. higher inductive types

Ex The circle $b : S^1$ (base)
$l : b = b$ (loop)



Given a dependent function on S^1, $f : \prod_{x:S^1} P(x)$, we get: 1- $f(b) : P(b)$

2- $apd_f(l) : tr_P(l, f(b)) = f(b)$



To give an induction principle on S^1 we need to consider the action on the loop, $apd_f(l)$!

Dependent action on generators
$$dgen : \left(\prod_{x:S^1} P(x)\right) \longrightarrow \left(\sum_{y:P(b)} tr_P(l, y) = y\right)$$
$$f \longmapsto (f(b), apd_f(l))$$

Induction: dgen has a section ind

we get $comp : dgen \cdot ind \sim id$, our computation rule

Rem In $\sum_{y:P(b)} tr_P(l,y)=y$ we have

$((y',p')=(y,p)) \simeq$ type of pairs α, β s.t.

$$\begin{tikzcd}
tr_P(l,y') \arrow[r, equal, "ap_{tr_P(l)}(\alpha)"] \arrow[d, equal, "p'"'] \arrow[dr, phantom, "\beta /\!/"] & tr_P(l,y) \arrow[d, equal, "p"] \\
y' \arrow[r, equal, "\alpha"'] & y
\end{tikzcd}$$

$(y,p) \rightsquigarrow f:\prod_{x:S^1} P(x)$ from __ind__ and $\alpha: f(b)=y$, β from __comp__

$$\begin{tikzcd}
tr_P(l,f(b)) \arrow[r, equal, "ap_{tr_P(l)}(\alpha)"'] \arrow[d, equal, "p'"'] \arrow[dr, phantom, "\beta /\!/"] & tr_P(l,y) \arrow[d, equal, "p"] \\
f(b) \arrow[r, equal, "\alpha"'] & y.
\end{tikzcd}$$

Thm dgen is an equivalence (dependent universal property)

Pf We need $ind \circ dgen \sim id$, i.e. $ind(dgen(f))=f$,

i.e. $g(x)=f(x)$ for $g :\equiv ind(dgen(f))$

Idea: use S^1-induction with $E(x) :\equiv g(x)=f(x)$

$$ind : \Big(\sum_{\alpha:E(b)} tr_E(l,\alpha)=\alpha\Big) \longrightarrow \Big(\prod_{x:S^1} E(x)\Big)$$

Need: $\alpha: g(b)=f(b)$ and $\beta: tr_E(l,\alpha)=\alpha$

[Get a map $(apd_g(p)\cdot r = ap_{tr_E(p)}(q)\cdot apd_f(p)) \to (tr_E(p,q)=r)$
by path induction on $p: x=x'$ with $r: x'=x'$, $q: x=x$

Enough: $\alpha: g(b)=f(b)$ and β

$$\begin{tikzcd}
tr_P(l,g(b)) \arrow[r, equal, "ap_{tr_P(l)}(\alpha)"] \arrow[d, equal, "apd_g(l)"'] \arrow[dr, phantom, "\beta /\!/"] & tr_P(l,f(b)) \arrow[d, equal, "apd_f(l)"] \\
g(b) \arrow[r, equal, "\alpha"'] & f(b)
\end{tikzcd}$$

They are given by $comp(f(b), apd_f(l))$
by the remark

Cor P/S^1, $y : P(b)$, $p : tr_P(l,y) = y$

The type of triples $f : \prod_{x:S^1} P(x)$, $\alpha : f(b) = y$, β is contractible,

$$\begin{tikzcd}
tr_P(l,f(b)) \arrow[r, equal, "ap_{tr_P(l)}(\alpha)"'] \arrow[d, equal, "apd_f(l)"'] \arrow[dr, phantom, "\beta \,/\!/"] & tr_P(l,y) \arrow[d, equal, "p"] \\
f(b) \arrow[r, equal, "\alpha"'] & y
\end{tikzcd}$$

i.e. there's a unique function with unique proofs of the desired equalities.

Thm $gen : (S^1 \to X) \to \left(\sum_{x:X} x = x\right)$ is an equivalence.

$f \mapsto (f(b), ap_f(l))$ (universal property)

Pf Careful with $tr_X(l)$!

Strategy: consider the triangle and construct the bottom map

$$\begin{tikzcd}
 & (S^1 \to X) \arrow[dl, "gen"'] \arrow[dr, "\sim", "dgen"'] & \\
\sum_{x:X} x = x \arrow[rr, "\sim"] & & \sum_{x:X} tr_X(l,x) = x
\end{tikzcd}$$

1- Construct $tr\text{-}X(p,x) : tr_X(p,x) = x$ by path induction

2- Define the map as $(x,\ell) \mapsto (x, tr\text{-}X(l,x) \cdot \ell)$

Check commutativity: consider

$$\begin{tikzcd}
tr_X(p,f(y)) \arrow[rr, equal, "tr\text{-}X(p,f(y))"'] \arrow[dr, equal, "apd_f(p)"'] & & f(y) \arrow[dl, equal, "ap_f(p)"] \\
 & f(y') &
\end{tikzcd}$$

It commutes by path induction. Now substitute b/y, b/y', l/p.

Cor $p : x = x$ in X

$$\sum_{f:S^1 \to X} \; \sum_{\alpha : f(b) = x} \alpha \cdot p = ap_f(l) \cdot \alpha$$

$$\begin{tikzcd}
f(b) \arrow[r, equal, "\alpha"] \arrow[d, equal, "ap_f(l)"'] \arrow[dr, phantom, "\beta \,/\!/"] & x \arrow[d, equal, "p"] \\
f(b) \arrow[r, equal, "\alpha"'] & x
\end{tikzcd}$$

is contractible.

In order to do algebra we still need an operation.

Def We construct $mul : S^1 \to (S^1 \to S^1)$

We want $mul(b) = id$, so consider the universal property wrt

$$ind : \left(\sum_{f : S^1 \to S^1} f = f\right) \to \left(S^1 \to (S^1 \to S^1)\right)$$

$(id, \text{eq-htpy}(H)) \longmapsto$ b-mul : $mul(b) = id$ and l-mul (plus mul)

$$\begin{tikzcd}
mul(b) \arrow[r, equal, "\text{b-mul}"] \arrow[d, equal, "ap_{mul}(l)"'] \arrow[dr, phantom, "\text{l-mul} \mathbin{/\!/}"] & id \arrow[d, equal, "\text{eq-htpy}(H)"] \\
mul(b) \arrow[r, equal, "\text{b-mul}"'] & id
\end{tikzcd}$$

We have to specify $H : id \sim id$

Use the dependent universal property on $E(x) :\equiv x = x$

$$ind : \left(\sum_{p : E(b)} tr_E(l, p) = p\right) \to \left(\prod_{x : S^1} E(x)\right)$$ (this is exactly $id \sim id$)

$(l, \gamma) \longmapsto H : id \sim id$ with $\alpha : H(b) = l$ and β

$$\begin{tikzcd}
tr_E(l, H(b)) \arrow[r, equal, "ap_{tr_E(l)}(\alpha)"] \arrow[d, equal, "apd_H(l)"'] \arrow[dr, phantom, "\beta \mathbin{/\!/}"] & tr_E(l, l) \arrow[d, equal, "\gamma"] \\
H(b) \arrow[r, equal, "\alpha"'] & l
\end{tikzcd}$$

We have to specify γ

Get the map $(p \cdot r = q \cdot p) \to (tr_E(p, q) = r)$ by path induction on $p : b = x$

with $q : b = b$, $r : x = x$

$refl_{l \cdot l} \mapsto \gamma$

Rem $H \neq$ refl-htpy for otherwise $mul(x, y) = y$

We have $mul(b, x) = x$ by the computation rule, while $mul(x, b) = x$ is more involved.

left-unit $:\equiv$ htpy-eq(b-mul)

Thm $\mathrm{mul}(x,b)=x$

Pf We want to use S^1-induction on $P(x) :\equiv \mathrm{mul}(x,b)=x$

$$\left(\sum_{p:\mathrm{mul}(b,b)=b} \mathrm{tr}_P(1,p)=p\right) \longrightarrow \left(\prod_{x:S^1} \mathrm{mul}(x,b)=x\right)$$

We have $\text{left-unit}(b) : \mathrm{mul}(b,b)=b$, so we need $\mathrm{tr}_P(1,\text{left-unit}(b)) = \text{left-unit}(b)$

[Construct $(\text{htpy-eq}(\mathrm{ap}_{\mathrm{mul}}(p))(b)\cdot r = q\cdot p) \longrightarrow (\mathrm{tr}_P(p,q)=r)$ by path induction on $p : b=x$ with $q : \mathrm{mul}(b,b)=b$, $r:\mathrm{mul}(x,b)=x$

Enough:

$$\begin{tikzcd}
\mathrm{mul}(b,b) \arrow[r, equal, "\text{left-unit}(b)"] \arrow[d, equal, "\text{htpy-eq}(\mathrm{ap}_{\mathrm{mul}}(1))(b)"'] \arrow[dr, phantom, "\beta \Nearrow"] & b \arrow[d, equal, "1"] \\
\mathrm{mul}(b,b) \arrow[r, equal, "\text{left-unit}(b)"'] & b
\end{tikzcd}$$

Since $H(b)=1$, we get β from

$$\begin{tikzcd}
\mathrm{mul}(b,b) \arrow[r, equal, "\text{htpy-eq}(b\text{-mul})(b)"] \arrow[d, equal, "\text{htpy-eq}(\mathrm{ap}_{\mathrm{mul}}(1))(b)"'] & b \arrow[d, equal, "H(b)"] \\
\mathrm{mul}(b,b) \arrow[r, equal, "\text{htpy-eq}(b\text{-mul})(b)"'] & b
\end{tikzcd}$$

which commutes by definition of $1\text{-mul} \mapsto \text{htpy-eq}(1\text{-mul})(b)$

What is <u>the fundamental cover of S^1</u>? It's a dependent function $D(\mathbb{Z},\mathrm{succ}) : S^1 \to \mathcal{U}$ induced by the universal property from the pair $(\mathbb{Z}:\mathcal{U}, \text{eq-equiv}(\mathrm{succ}) : \mathbb{Z}=\mathbb{Z})$

$$\begin{tikzcd}
D(\mathbb{Z},\mathrm{succ})(b) \arrow[r, "e", "\sim"'] \arrow[d, "\mathrm{tr}_{D(\mathbb{Z},\mathrm{succ})}(1)"', "\wr"] & \mathbb{Z} \arrow[d, "\mathrm{succ}", "\wr"'] \\
D(\mathbb{Z},\mathrm{succ})(b) \arrow[r, "\sim", "e"'] & \mathbb{Z}
\end{tikzcd}$$

The fiber over the base is $\mathbb{Z}$ and going through the loop once sends a point in the fiber to the next one.

Homotopy Pushouts

How to construct new types?

In algebraic topology we attach cells to spaces by pushout.

Def

- A span $\mathcal{S}$ is a triple (S, f, g)

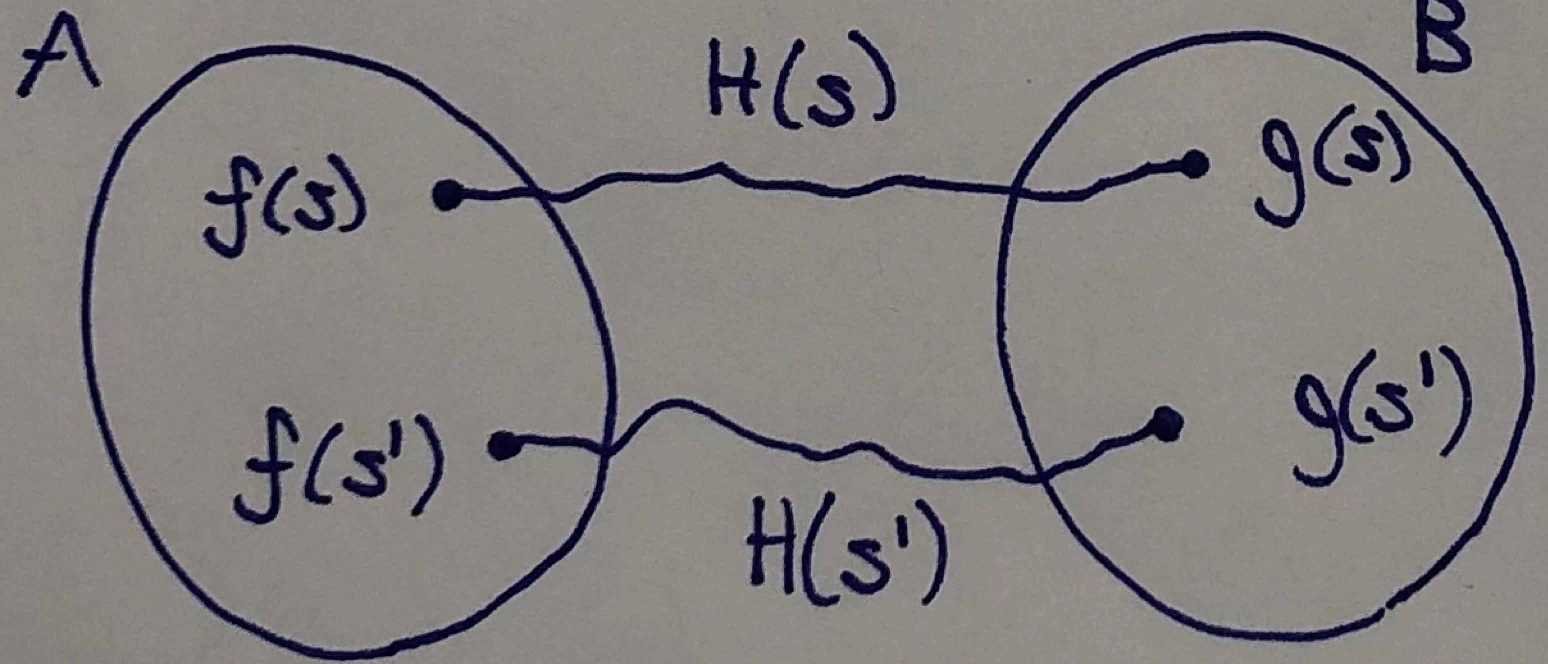
$$\begin{tikzcd} S \arrow[r, "g"] \arrow[d, "f"'] \arrow[dr, phantom, "H \nearrow"] & B \arrow[d, "j"] \\ A \arrow[r, "i"'] & X \end{tikzcd}$$

- A commutative square is a span $\mathcal{S}$ with a triple $(i, j, H) : \sum_{i : A \to X} \sum_{j : B \to X} if \sim jg$ ← $\mathrm{cocones}_{\mathcal{S}}(X)$
- $\mathrm{cocone\text{-}map}_{X'}(i, j, H) : (X \to X') \to \mathrm{cocones}_{\mathcal{S}}(X')$

$$h \mapsto (hi, hj, hH)$$

- A (homotopy) pushout is a commutative square s.t. this map is an equivalence $\forall X'$

A pushout X is a disjoint union of A and B with added paths from $f(s)$ to $g(s)$.



Pushouts may not exist! They do allow to construct new types.

Lemma $(i, j, H), (i', j', H') : \mathrm{cocones}_{\mathcal{S}}(X)$

$$((i, j, H) = (i', j', H')) \simeq \sum_{K : i \sim i'} \sum_{L : j \sim j'} H' Kf \sim Lg H$$

$$\begin{tikzcd} if \arrow[r, "Kf"] \arrow[d, "H"'] \arrow[dr, phantom, "M \nearrow"] & i'f \arrow[d, "H'"] \\ jg \arrow[r, "Lg"'] & j'g \end{tikzcd}$$

Lemma

$$\begin{tikzcd} S \arrow[r, "g"] \arrow[d, "f"'] \arrow[dr, phantom, "H \Nearrow"] & B \arrow[d, "j"] \\ A \arrow[r, "i"'] & X \end{tikzcd} \quad \text{pushout} \qquad \begin{tikzcd} S \arrow[r, "g"] \arrow[d, "f"'] \arrow[dr, phantom, "H' \Nearrow"] & B \arrow[d, "j'"] \\ A \arrow[r, "i'"'] & X' \end{tikzcd}$$

$$\sum_{h:X\to X'} \; \sum_{K:hi\sim i'} \; \sum_{L:hj\sim j'} H'Kf \sim LghH \text{ is contractible}$$

$$\begin{tikzcd} hif \arrow[r, "Kf"] \arrow[d, "hH"'] \arrow[dr, phantom, "M \Nearrow"] & i'f \arrow[d, "H'"] \\ hjg \arrow[r, "Lg"'] & jg \end{tikzcd}$$

(Uniqueness of factorization)

Thm Two commutative squares.
If two of the following hold, so does the third:
(i) (i, j, H) is a pushout on $\mathcal{S}$
(ii) (i', j', H') //
(iii) $h: X \to X'$ is an equivalence

Pf

$$\begin{tikzcd} (X' \to Y) \arrow[rr, "h^*"] \arrow[dr, "\text{cocone-map}(i',j',H)"'] & & (X \to Y) \arrow[dl, "\text{cocone-map}(i,j,H)"] \\ & \text{cocone}_{\mathcal{S}}(Y) & \end{tikzcd}$$

h is an equivalence $\Leftrightarrow$ h^* is $\forall Y$

Cor If the two squares are pushouts, then

$$\circledast \; \sum_{h:X\simeq X'} \; \sum_{K:hi\sim i'} \; \sum_{L:hj\sim j'} H'Kf \sim LghH \text{ is contractible}$$

Cor $\mathcal{S}$ is a span in $\mathcal{U}$ $\Rightarrow$ $\displaystyle\sum_{X:\mathcal{U}} \sum_{c:\text{cocone}_{\mathcal{S}}(X)} \prod_{Y:\mathcal{U}} \text{is-equiv}(\text{cocone-map}_Y(c))$
is a proposition

Pf $\circledast \simeq ((X, c, u) = (X', c', u'))$

Duality

Lemma

$$\begin{tikzcd} \mathrm{cocone}_{\mathcal{S}}(X) \arrow[r, "\pi_2"] \arrow[d, "\pi_1"'] \arrow[dr, phantom, "\pi_3 \Nearrow"] & X^B \arrow[d, "g^*"] \\ X^A \arrow[r, "f^*"'] & X^S \end{tikzcd}$$

$\pi_3 :\equiv \lambda(i,j,H).\ \text{eq-htpy}(H)$ is a pullback

Pf The gap map $\mathrm{cocone}_{\mathcal{S}}(X) \longrightarrow X^A \times_{X^S} X^B$ is induced by eq-htpy on the total spaces.

Prop Given a commutative square, TFAE:

(i) it's a pushout

(ii) the commutative square with $H' :\equiv \lambda h.\ \text{eq-htpy}(hH)$ is a pullback $\forall T$

$$\begin{tikzcd} T^X \arrow[r, "j^*"] \arrow[d, "i^*"'] \arrow[dr, phantom, "H' \Nearrow"] & T^B \arrow[d, "g^*"] \\ T^A \arrow[r, "f^*"'] & T^S \end{tikzcd}$$

Pf

$$\begin{tikzcd} & T^X \arrow[dl, "\text{cocone-map}(i,j,H)"'] \arrow[dr, "\text{gap}(i^*, j^*, \lambda h.\text{eq-htpy}(hH))"] & \\ \mathrm{cocone}_{\mathcal{S}}(T) \arrow[rr, "\sim", "\text{gap}(i,j,\text{eq-htpy}(H))"'] & & T^A \times_{T^S} T^B \end{tikzcd}$$

Def • X type

Its suspension is given by:
- ΣX type
- $N, S : \Sigma X$ poles
- $\text{merid} : X \longrightarrow (N = S)$

s.t.

$$\begin{tikzcd} X \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{merid} \Nearrow"] & 1 \arrow[d, "\text{const}_S"] \\ 1 \arrow[r, "\text{const}_N"'] & \Sigma X \end{tikzcd}$$

is a pushout

• Spheres: $S^0 :\equiv 2$ $\quad S^{n+1} :\equiv \Sigma S^n$

It works because
$$\begin{tikzcd} X^{S^1} \arrow[r] \arrow[d] & X^1 \arrow[d] \\ X^1 \arrow[r] & X^2 \end{tikzcd}$$
is a pullback $\forall X$

$2 \to (b=b)$
$0_2 \mapsto 1$
$0_1 \mapsto \mathrm{refl}_b$

$\Rightarrow S^1 \simeq \Sigma S^0$

Inspired by the link between S^1 and suspensions, we get the following lemma.

Lemma $(\Sigma X \to Y) \to \left(\sum_{y,y':Y} X \to y = y'\right)$ is an equivalence

$f \mapsto (f(N), f(S), f \cdot \mathrm{merid})$

Pf
$$\begin{tikzcd} (\Sigma X \to Y) \arrow[d, "\text{cocone-map}"', "\wr"] \arrow[dr] & \\ \mathrm{cocone}_{\mathcal{S}}(Y) \arrow[r, "\sim"] & \left(\sum_{y,y':Y} X \to y = y'\right) \end{tikzcd}$$
$(i, j, H) \mapsto (i(*), j(*), H)$

Thm If the left square is a pushout, then the right one is iff the rectangle is (pasting)
$$\begin{tikzcd} A \arrow[r, "i"] \arrow[d, "f"'] \arrow[dr, phantom, "\circlearrowleft"] & B \arrow[r, "k"] \arrow[d, "g"] \arrow[dr, phantom, "\circlearrowleft"] & C \arrow[d, "h"] \\ X \arrow[r, "j"'] & Y \arrow[r, "l"'] & Z \end{tikzcd}$$

Def $f : A \to B$
Its cofiber is the pushout
$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d] \arrow[dr, phantom, "H \Nearrow"] & B \arrow[d, "\mathrm{inr}_f"] \\ 1 \arrow[r, "\mathrm{inl}_f"'] & \mathrm{cofib}(f) \end{tikzcd}$$

Lemma $\mathrm{cofiber}(\mathrm{inr}_f) \simeq \Sigma A$

Pf
$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d] & B \arrow[r] \arrow[d, "\mathrm{inr}_f"] & 1 \arrow[d, "\mathrm{inr}_{\mathrm{inr} f}"] \\ 1 \arrow[r, "\mathrm{inl}_f"] & \mathrm{cofib}(f) \arrow[r, "\mathrm{inl}_{\mathrm{inr}_f}"] & \mathrm{cofiber}(\mathrm{inr}_f) \simeq \Sigma A \end{tikzcd}$$