

Encode - Decode

have some type X defined as a colimit. want to understand $\text{Id}_X(x=y)$

$\text{code}: X \rightarrow X \rightarrow \text{Type}$ using univ. property of X and show(decode) that it is the identity type.

$$\text{code}(x,y) \xrightleftharpoons[\text{encode}]{\text{decode}} \text{Id}_X(x,y)$$

We find the encoding /decoding using $\text{Id-elim}/\text{path induct}$

"every equality with free end pts might as well be refl ".

$$\begin{array}{l} A \longrightarrow \sum_{x:A} \sum_{y:A} (x=y) \\ a \longmapsto (a, a, \text{refl}) \end{array} \left. \vphantom{\begin{array}{l} A \\ a \end{array}} \right\} \text{this is a homotopy equivalence (shrink paths to starting point).}$$

for any $a:A$, $\sum_{x:A} (a=x)$ is contractible by the same argument.

given $P: \prod_{x:A} \prod_{y:A} \prod_{p:x=y} \text{Type}$ (this is a type)

to define $f:P$, $f: \prod_{x:A} \prod_{y:A} \prod_{p:x=y} P(x,y,p)$

encode: $\prod_{x:X} \prod_{y:X} \prod_{p:x=y} \text{code}(x,y)$

enough to define $\text{encode}(\text{refl}): \text{code}(x,x)$

encode(x, x, refl) by case analysis (since X is a colimit)

ex. ~~encode inl~~ $X = A + B$

code $A+B \rightarrow A+B \rightarrow \text{Type}$

code(inl(a), inl(a')) = $(a =_A a')$

code(inl(a), inr(b)) = ϕ

code(inr(b), inl(a)) = ϕ

code(inr(b), inr(b')) = $(b =_B b')$

encode(x, x, refl) by case analysis:

encode(inl(a), inl(a), refl) = refl_a : code(inl(a), inl(a))

encode(inr(b), inr(b), refl) = refl_b : code(inr(b), inr(b))

decode: $\prod_{x:X} \prod_{y:X} \prod_{p:\text{code}(x,y)} (x=y)$

decode(inl(a), inl(a'), p) = $a \cdot p_{\text{inl}}(p) : \text{inl}(a) = \text{inl}(a')$

where $a \cdot p_f$: paths in domain of f \rightarrow paths in target of f

push path forward. $f: X \rightarrow Y$ some fcn.

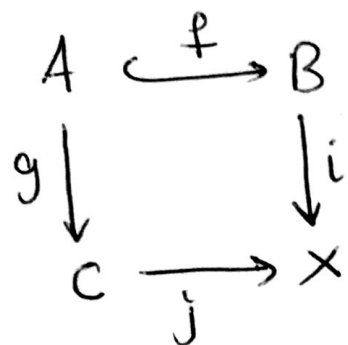
decode(inl(a), inr(b), p) = \checkmark (there are no p's.)

$\frac{}{b} \parallel \frac{}{a} = \checkmark \parallel \frac{}{} \parallel$

decode(inr(b), inr(b')) = $a \cdot p_{\text{inr}}(p)$

want $\prod_{x:X} \prod_{y:X} \prod_{p:x=y} \text{decode}(\text{encode}(p)) = p$
 \uparrow homotopy.

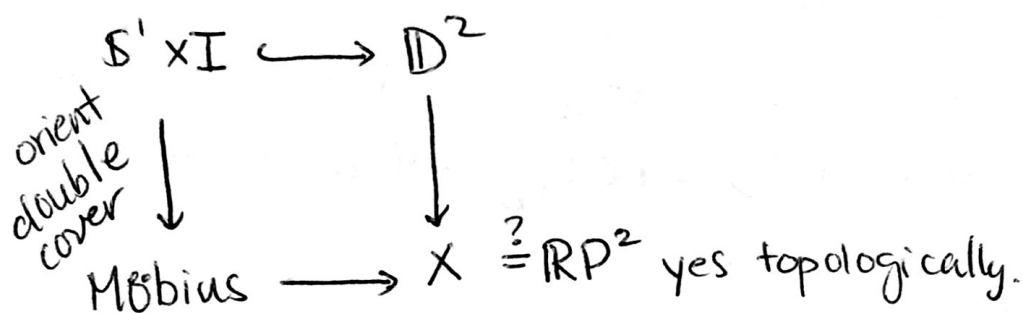
suffices: assume p is refl $y \equiv x$. for refl $\text{decode}(\text{encode}(\text{refl}))$
 $= \text{refl}$ on the nose.



f injective, A, B, C sets. 1) is X a set?

2) is j injective?

3) more generally, what are Ids of a pushout of an embedding.



but as a groupoid, is this a set?

[*wrong correction appears later

$$\text{code}(\text{inl}(b), \text{inl}(b')) = \sum_{a, a': A} (b =_B f(a)) \times (b' =_B f(a')) \times (g(a) = g(a'))$$

$$\text{code}(\text{inr}(c), \text{inr}(c')) = (c =_C c') \quad \text{b/c } f \text{ is injective}$$

$$\text{code}(\text{inl}(b), \text{inr}(c)) = \sum_{a: A} (b = f(a)) \times (c = g(a))$$

$$\text{code}(\text{inr}(c), \text{inl}(b)) = \sum_{a: A} (c = g(a)) \times (b = f(a))$$

$$\text{ap}_{\text{code}}(\text{inr}(c), \text{glue}(a)) = \prod_{c: C} \prod_{a: A} \text{code}(\text{inr}(c), \text{inl}(f(a))) = \text{code}(\text{inr}(c), \text{inl}(g(a)))$$

this is
b/c f is
an injection.

$$= \prod_{c: C} \prod_{a: A} \left(\sum_{a': A} (c = g(a)) \times (f(a) = f(a')) \right)$$

$$\sum_{a': A} (c = g(a)) \times (f(a) = f(a'))$$

\Rightarrow given $\gamma: (c =_c g(a'))$ $\rho: (f(a) =_B f(a'))$

$a =_A a'$ in A b/c f is an injection, so $g(a) =_c g(a')$
 so we get a path $\gamma: (c =_c g(a))$ (path composition)

\Leftarrow given $\gamma: (c =_c g(a))$ let $a' = a$ in A .

then we get $\rho: f(a) =_B f(a')$ by choosing
 $\rho = \text{refl}_{f(a)}$.

~~$$\phi(a', \gamma, \rho) = g \circ a\rho$$~~

$$\phi(a', \gamma, \rho) = \gamma \circ \overline{a\rho_g(a\rho_f^{-1}(p))}$$

$$\Psi(\gamma) = (a, \gamma, \text{refl}_{f(a)})$$

Alternatively: $\sum_{a': A} (c =_c g(a')) \times (f(a) =_B f(a'))$
 $\simeq \sum_{a': A} (c =_c g(a')) \times (a = a')$ since f is injective.
 $\simeq \sum_{(a', r)} (c =_c g(a'))$
 $\simeq (a', r) \cdot \left[\sum_{a': A} (a = a') \right]$
 $\simeq (c =_c g(a))$ \nwarrow contractible.

by symmetry can define
 $\text{ap}_{\text{code}}(\text{glue}(a), \text{inr}(c))$

Now define

$$\text{ap}_{\text{code}}(\text{glue}(a), \text{inl}(b)) = \prod_{b: B} \prod_{a': A} \text{code}(\text{inl}(f(a)), \text{inl}(b))$$

||

$$\text{code}(\text{inr}(g(a)), \text{inl}(b))$$

$$= \prod_{a_1: A} \prod_{b: B} \left(\sum_{\substack{a_2: A \\ a_3: A}} (b =_B f(a_2)) \times (b =_{f(a_1)} f(a_3)) \times (g(a_2) = g(a_3)) \right)$$

is need to prove this equivalence

$$\sum_{a_4: A} (\cancel{g(a_4)}) (g(a_1) = g(a_4)) \times (b = f(a_4))$$

proof of equivalence:

$$\sum_{\substack{a_2: A \\ a_3: A}} (b =_B f(a_2)) \times (f(a_1) =_B f(a_3)) \times (g(a_2) = g(a_3))$$

$$\cong \sum_{\substack{a_2: A \\ a_3: A}} (b =_B f(a_2)) \times (a_1 =_A a_3) \times (g(a_2) = g(a_3))$$

since f is inject.

$$\cong \sum_{\substack{a_2: A \\ \cancel{a_3: A}}} (b =_B f(a_2)) \times (g(a_2) = g(a_3))$$

$$r: \sum_{a_3: A} (a_1 =_A a_3)$$

$$\cong \sum_{a_2: A} (b =_B f(a_2)) \times (g(a_2) = g(a_1))$$

$$\cong \sum_{a_4: A} (b = f(a_4)) \times (g(a_1) = g(a_4))$$

□

Now define $\text{ap}_{\text{code}}(\text{glue}(a), \text{glue}(a'))$

~~$\text{ap}_{\text{code}}(\text{glue}(a), \text{glue}(a')) = \text{code}(\text{inl}(f(a)), \text{inl}(f(a')))$~~

we have shown each of these equalities.

~~ap_{code}~~

$$\text{code}(\text{inl}(f(a)), \text{inl}(f(a'))) \stackrel{x}{=} \text{code}(\text{inl}(f(a)), \text{inr}(g(a')))$$

w \parallel

\parallel y

$$\text{code}(\text{inr}(g(a)), \text{inl}(f(a'))) \stackrel{z}{=} \text{code}(\text{inr}(g(a)), \text{inr}(g(a')))$$

$\text{ap}_{\text{code}}(\text{glue}(a), \text{glue}(a'))$ shows that this square of equalities commute.

$$x: \text{ap}_{\text{code}}(\text{inl}(b), \text{glue}(a)): \prod_{a:A} \prod_{b:B} \text{code}(\text{inl}(b), \text{inl}(f(a))) = \text{code}(\text{inl}(b), \text{inr}(g(a)))$$

$$y: \text{ap}_{\text{code}}(\text{glue}(a), \text{inr}(c)): \prod_{a:A} \prod_{c:C} \text{code}(\text{inl}(f(a)), \text{inr}(c)) = \text{code}(\text{inr}(g(a)), \text{inr}(c))$$

$$w: \text{ap}_{\text{code}}(\text{glue}(a), \text{inl}(b)): \prod_{a:A} \prod_{b:B} \text{code}(\text{inl}(f(a)), \text{inl}(b)) = \text{code}(\text{inr}(g(a)), \text{inl}(b))$$

$$z: \text{ap}_{\text{code}}(\text{inr}(c), \text{glue}(a)): \prod_{a:A} \prod_{c:C} \text{code}(\text{inr}(c), \text{inl}(f(a))) = \text{code}(\text{inr}(c), \text{inr}(g(a)))$$