Cubical Agda Explore

Week 6, Spring 2023

Chenchao Ding, Feb 19

Path

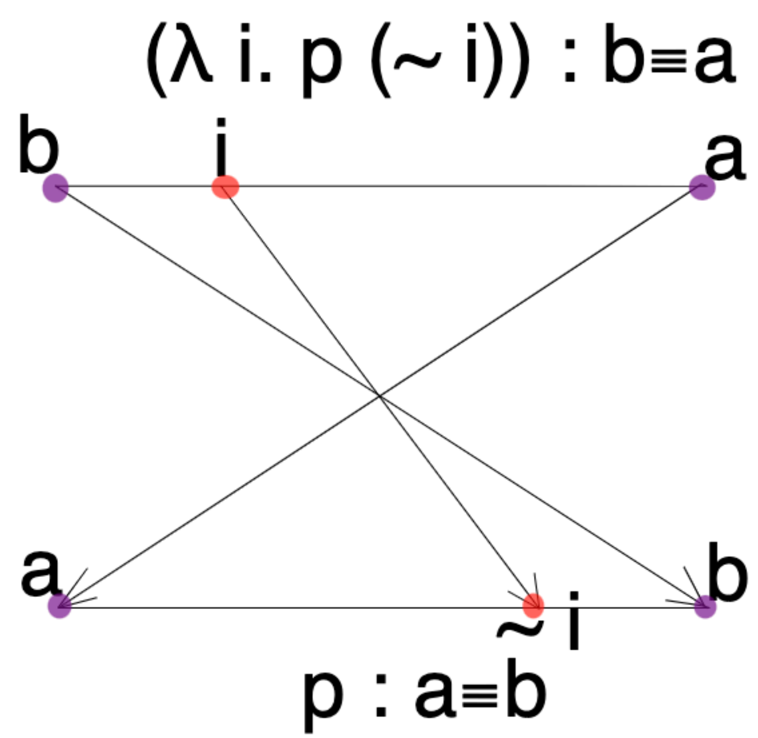
Reflexivity

```
refl' : (a : A) → Path A a a
refl' a = λ _ → a
```

Path

Symmetricity (~)

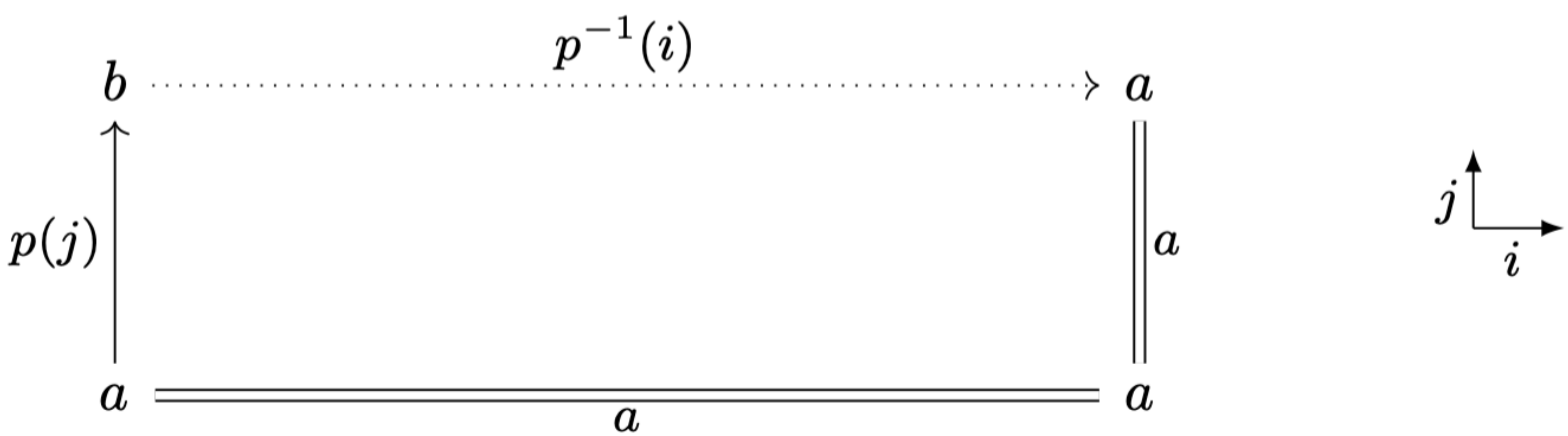
```
!'_ : ∀ {ℓ} {A : Type ℓ} {a b : A} → a ≡ b → b ≡ a
!'_ {ℓ}{A}{a}{b} p = λ i → p (~ i)
```



Path

Symmetricity (hcomp)

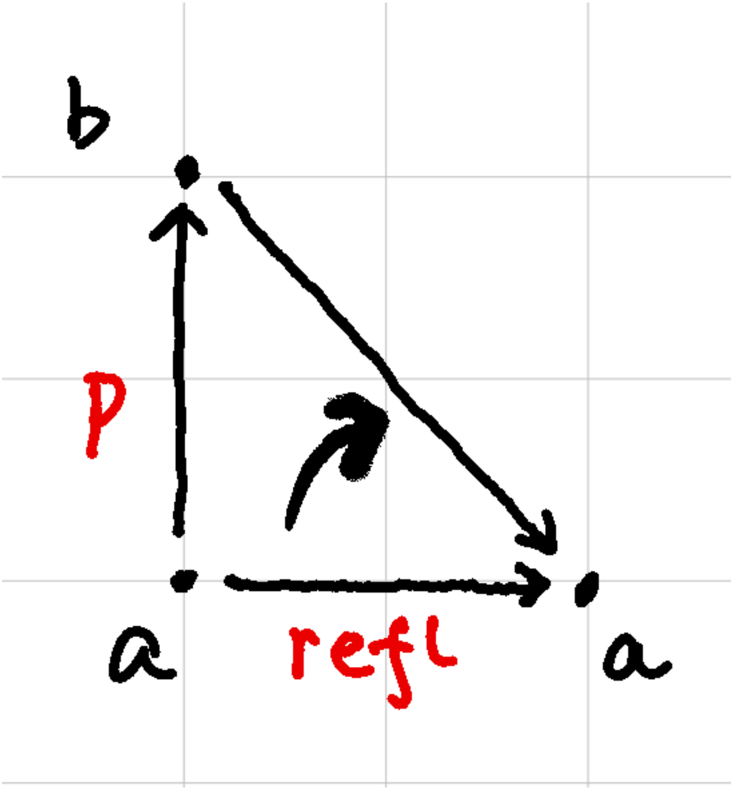
```
!_ : ∀ {ℓ} {A : Type ℓ} {a b : A} → a ≡ b → b ≡ a
!_ {ℓ}{A}{a}{b} p i = hcomp walls a
  where
    walls : ∀ (j : I) → Partial (~ i ∨ i) A
    walls j (i = i0) = p j
    walls j (i = i1) = a
```



Path

Symmetricity (coercion)

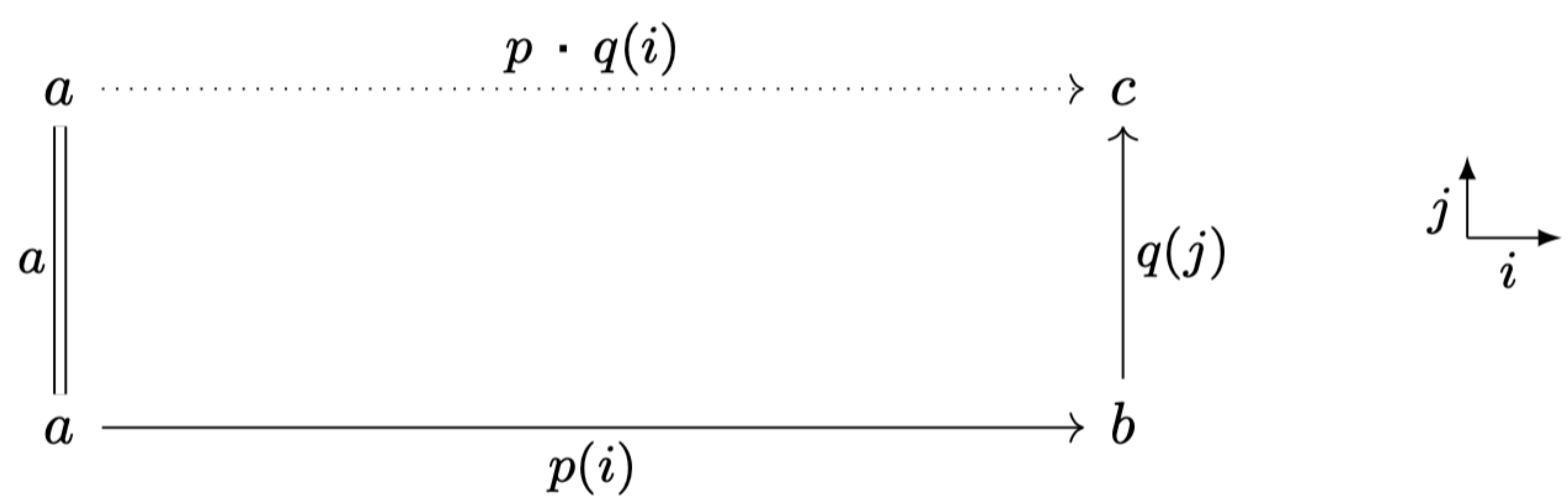
```
coe0→1 : ∀ {ℓ} (A : I → Type ℓ) → A i0 → A i1
coe0→1 A a = transp (λ i → A i) i0 a

!''_ : ∀ {ℓ} {A : Type ℓ} {a b : A} → a ≡ b → b ≡ a
!''_ {ℓ}{A}{a}{b} p = coe0→1 (λ i → p i ≡ a) refl
```



Path

Transitivity

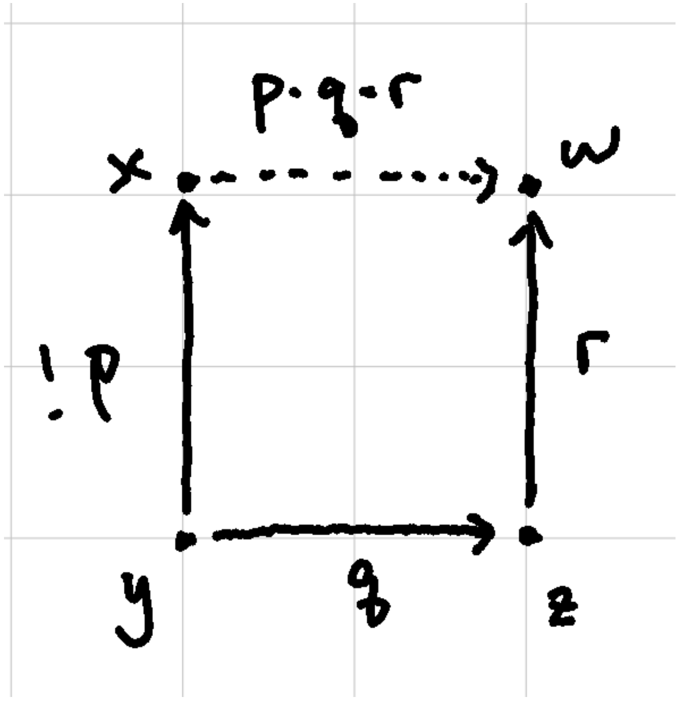
```
compPath : ∀ {ℓ} {A : Type ℓ} {a b c : A} → a ≡ b → b ≡ c → a ≡ c
compPath {ℓ}{A}{a}{b}{c} p q i = hcomp walls (p i)
  where
    walls : ∀ (j : I) → Partial (~ i ∨ i) A
    walls j (i = i0) = a
    walls j (i = i1) = q j
```



Path

Transitivity

```
_••_••_ : ∀ {ℓ} {A : Type ℓ} {x y z w : A} → x ≡ y → y ≡ z → z ≡ w → x ≡ w
_••_••_ {ℓ}{A}{x}{y}{z}{w} p q r i = hcomp walls (q i)
  where
    walls : ∀ (j : I) → Partial (~ i ∨ i) A
    walls j (i = i0) = (! p) j  -- or p (~ j)
    walls j (i = i1) = r j
```



Path

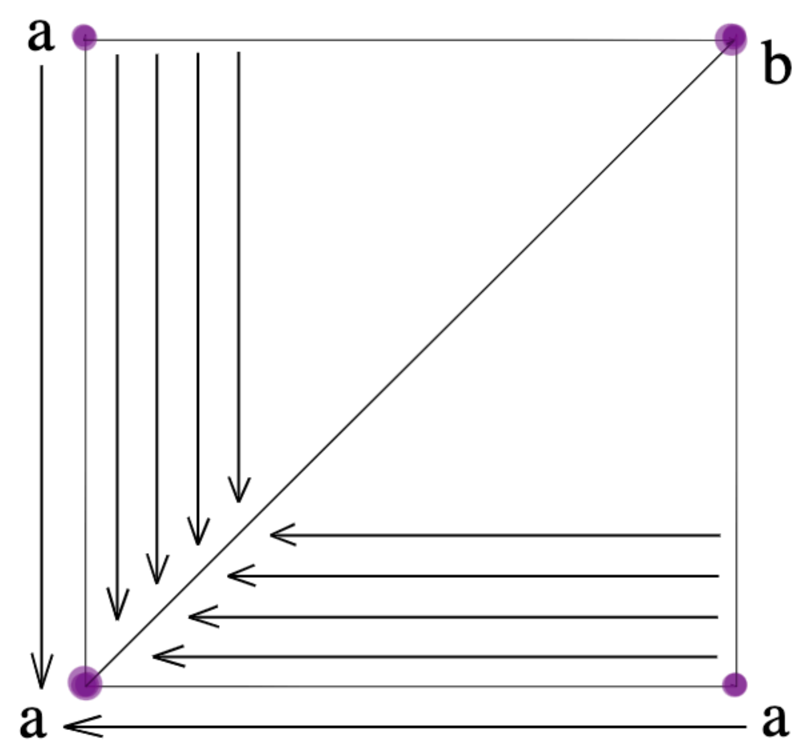
Transitivity

```
compPath1 : ∀ {ℓ} {A : Type ℓ} {a b c : A} → a ≡ b → b ≡ c → a ≡ c
compPath1 {ℓ}{A}{a}{b}{c} p q = refl •• p •• q

compPath2 : ∀ {ℓ} {A : Type ℓ} {a b c : A} → a ≡ b → b ≡ c → a ≡ c
compPath2 {ℓ}{A}{a}{b}{c} p q = p •• refl •• q

compPath3 : ∀ {ℓ} {A : Type ℓ} {a b c : A} → a ≡ b → b ≡ c → a ≡ c
compPath3 {ℓ}{A}{a}{b}{c} p q = p •• q •• refl
```

Weak connections

Meet

```
-- Square [left] [right] [bottom] [top]
∧-conn' : ∀ {ℓ} {A : Type ℓ} {a b : A} (p : a ≡ b) → Square refl p refl p
∧-conn' {ℓ}{A}{a}{b} p i j = p (i ∧ j)
```

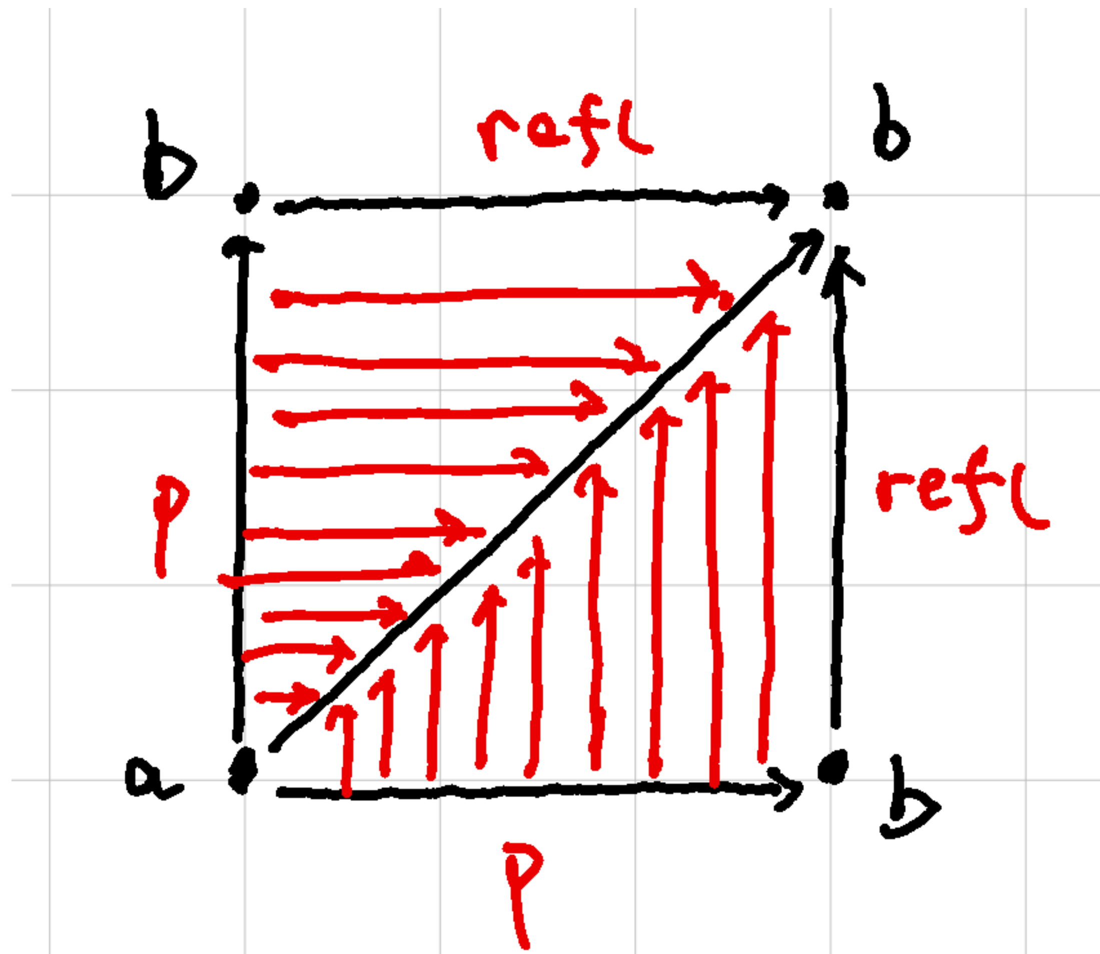


Another diagonal is not equal to `refl`

Weak connections

Join

```
v-conn' : ∀ {ℓ} {A : Type ℓ} {a b : A} (p : a ≡ b) → Square p refl p refl
v-conn' {ℓ}{A}{a}{b} p i j = p (i ∨ j)
```



The groupoid laws

`hcomp` and `hfill`

$$
\begin{tikzcd}
\bullet \arrow[r, "M"] \arrow[d, "N_0"'] & \bullet \arrow[d, "N_1"] \\
\bullet \arrow[r, dashed, "K"'] & \bullet
\end{tikzcd}
\qquad
\begin{tikzcd}
\bullet \arrow[r, "M"] \arrow[d, "N_0"'] \arrow[dr, phantom, "F"] & \bullet \arrow[d, "N_1"] \\
\bullet \arrow[r, dashed, "K"'] & \bullet
\end{tikzcd}
$$

$$y : \mathbb{I} \gg K \coloneqq \mathsf{hcom}_A^{0 \to 1}(M; y \equiv 0 \hookrightarrow x.N_0, y \equiv 1 \hookrightarrow x.N_1) \in A$$

$$x : \mathbb{I}, y : \mathbb{I} \gg F \coloneqq \mathsf{hcom}_A^{0 \to x}(M; y \equiv 0 \hookrightarrow x.N_0, y \equiv 1 \hookrightarrow x.N_1) \in A$$

The groupoid laws

Right unit

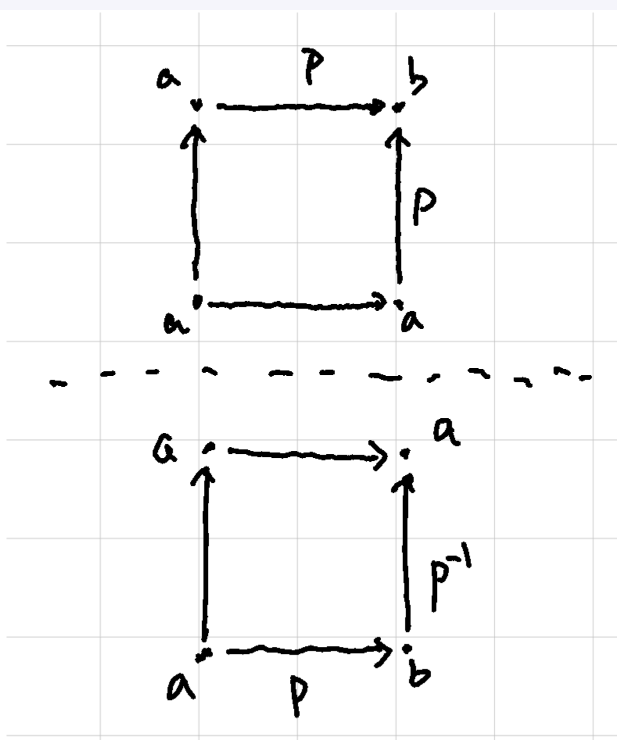
```
•-filler : ∀ {ℓ} {A : Type ℓ} {a b c : A} (p : a ≡ b) (q : b ≡ c)
         → Square refl q p (p • q)
•-filler {ℓ}{A}{a}{b}{c} p q i j = hfill walls (inS (p i)) j
  where
    walls : ∀ (j : I) → Partial (~ i ∨ i) A
    walls j (i = i0) = a
    walls j (i = i1) = q j
```

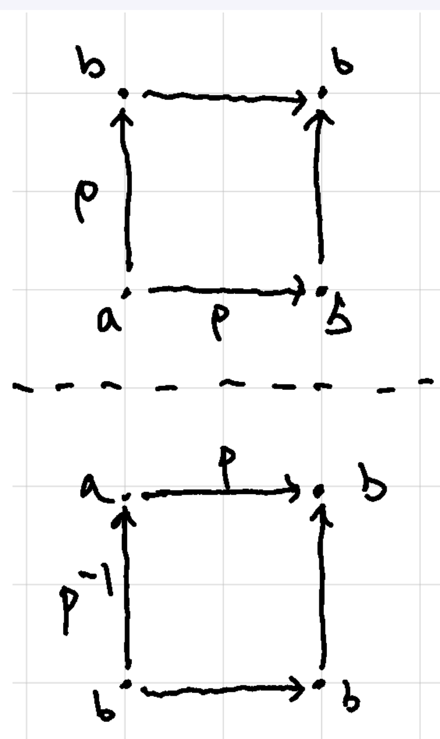
```
ru : ∀ {ℓ} {A : Type ℓ} {a b : A} {p : a ≡ b}
   → p ≡ p • refl
ru {ℓ}{A}{a}{b}{p} j i = •-filler p refl i j
```

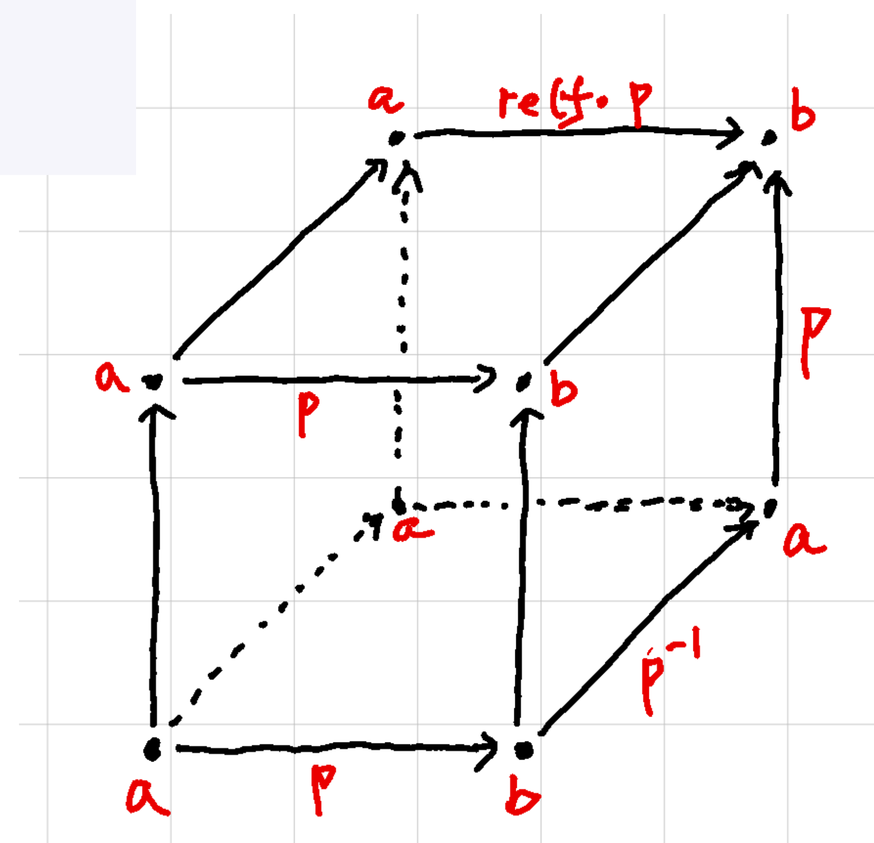
The groupoid laws

Left unit

```
lu : ∀ {ℓ} {A : Type ℓ} {a b : A} {p : a ≡ b}
   → p ≡ refl • p
lu {ℓ}{A}{a}{b}{p} j i = hcomp walls (p (~ j ∧ i))
  where
    walls : ∀ (k : I) → Partial (~ i ∨ i ∨ ~ j) A
    walls k (i = i0) = a
    walls k (i = i1) = p (~ j ∨ k)
    walls k (j = i0) = p i
```



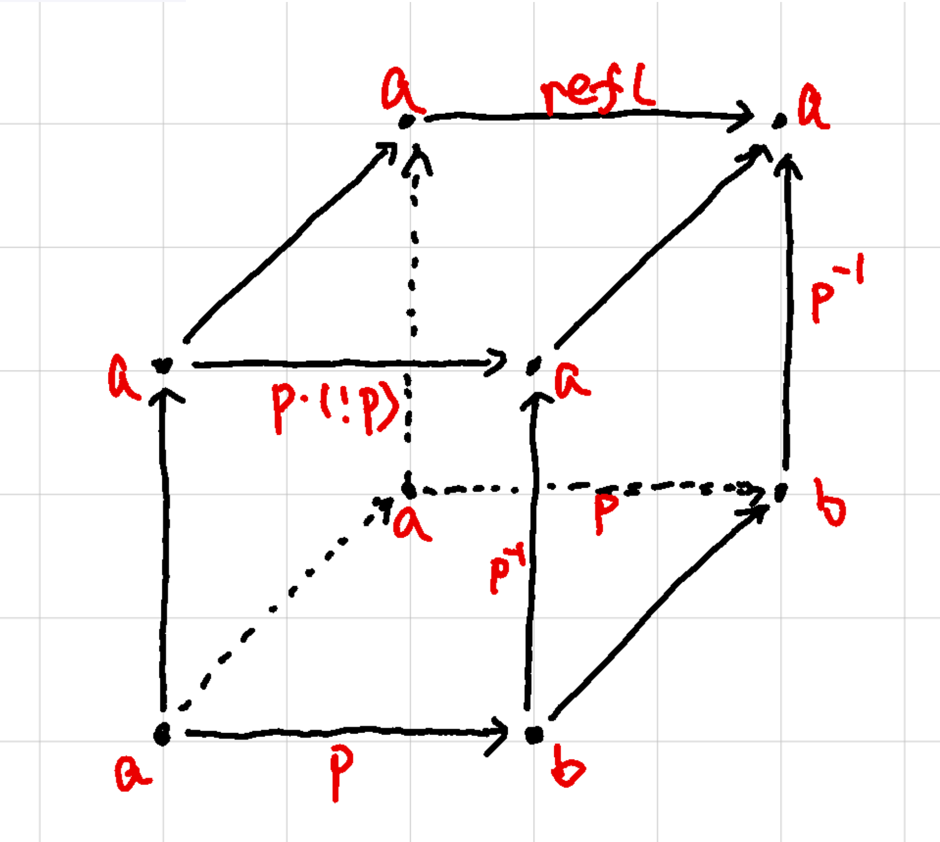




The groupoid laws

Right cancel

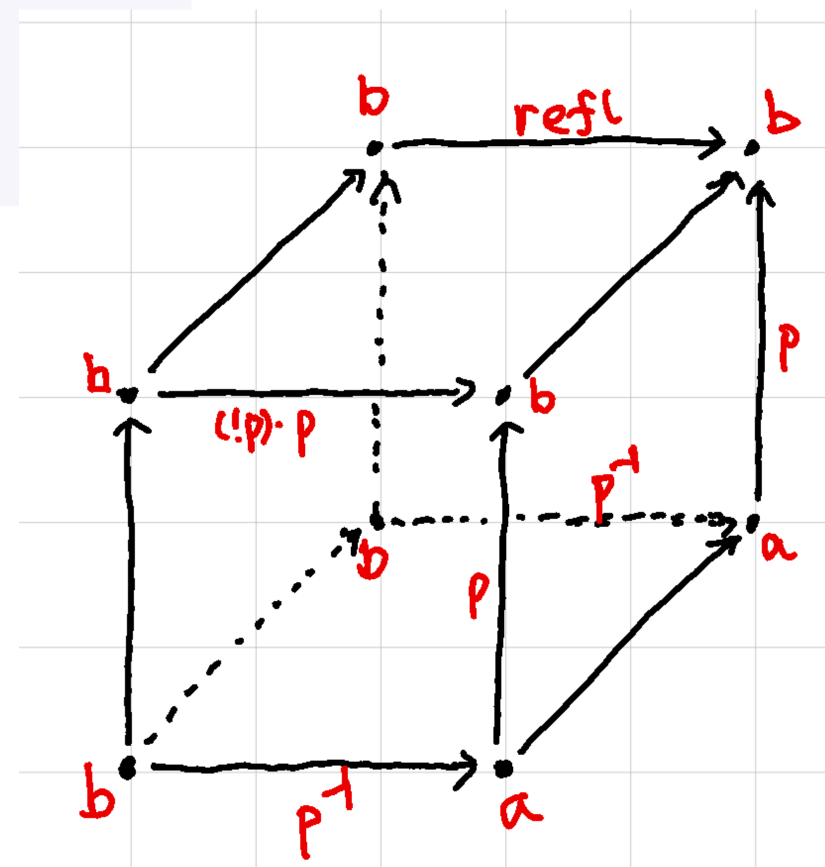
```
rc : ∀ {ℓ} {A : Type ℓ} {a b : A} {p : a ≡ b}
   → p • (sym p) ≡ refl
rc {ℓ}{A}{a}{b}{p} j i = hcomp walls (p i)
  where
    walls : ∀ (k : I) → Partial (~ i ∨ i ∨ j) A
    walls k (i = i0) = a
    walls k (i = i1) = p (~ k)
    walls k (j = i1) = p (i ∧ ~ k)
```



The groupoid laws

Left cancel

```
lc : ∀ {ℓ} {A : Type ℓ} {a b : A} {p : a ≡ b}
   → (sym p) • p ≡ refl
lc {ℓ}{A}{a}{b}{p} j i = hcomp walls (p (~ i))
  where
    walls : ∀ (k : I) → Partial (~ i ∨ i ∨ j) A
    walls k (i = i0) = b
    walls k (i = i1) = p k
    walls k (j = i1) = p (~ i ∨ k)
```



The groupoid laws

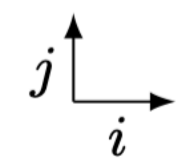
Involution

```
inv : ∀ {ℓ} {A : Type ℓ} {a b : A} {p : a ≡ b}
    → (sym (sym p)) ≡ p
inv {ℓ}{A}{a}{b}{p} = refl
```

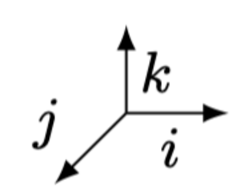
The groupoid laws

Associativity

$$\begin{tikzcd}
a \arrow[r, "(p \cdot (q \cdot r))(i)"] \arrow[d, equal] \arrow[dr, phantom, "\alpha"] & d \\
a \arrow[r, "p(i)"'] & b \arrow[u, "(q \cdot r)(j)"']
\end{tikzcd}
\qquad
\begin{tikzcd}
a \arrow[r, "((p \cdot q) \cdot r)(i)"] \arrow[d, equal] \arrow[dr, phantom, "\beta"] & d \\
a \arrow[r, "p(i)"'] & b \arrow[u, "(q \cdot r)(j)"']
\end{tikzcd}$$

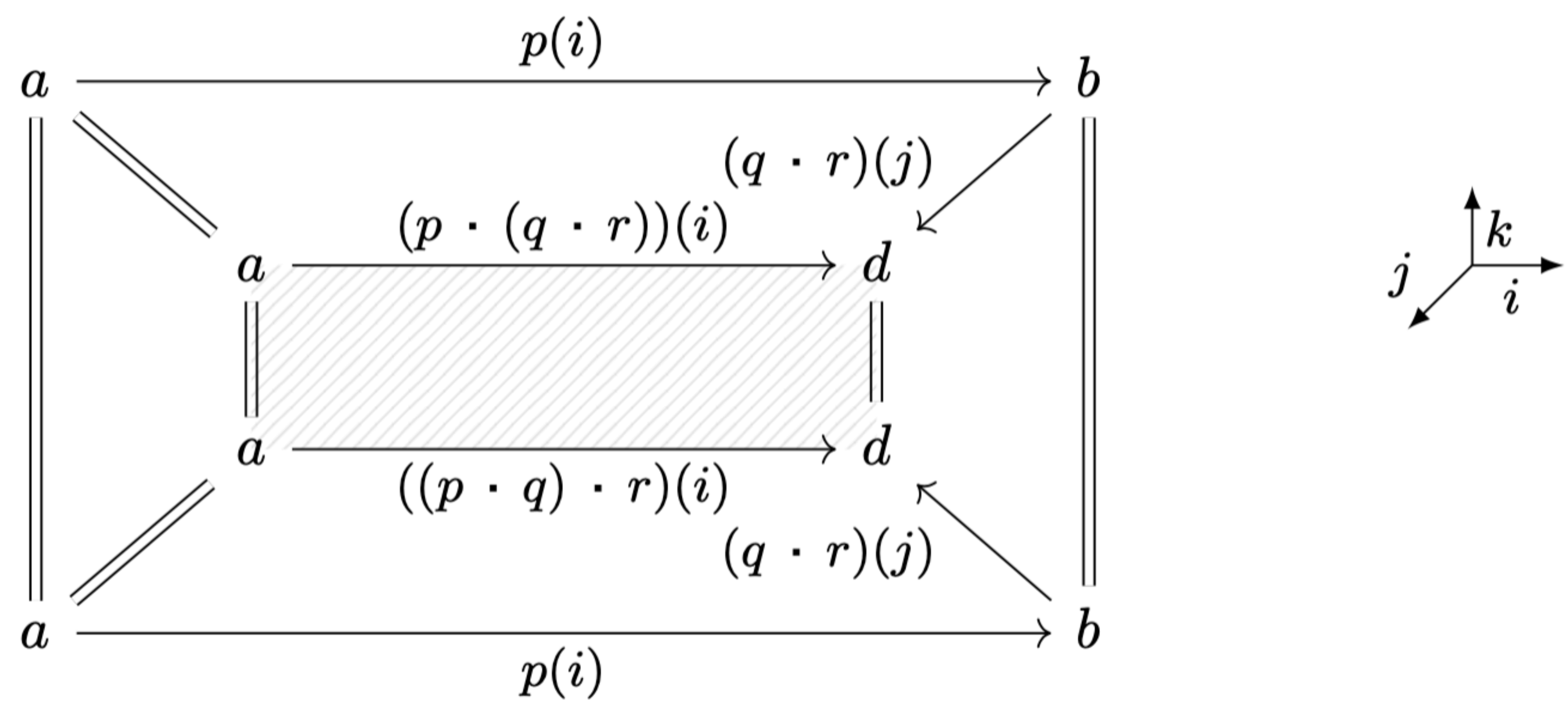


$$\begin{tikzcd}
a \arrow[rrr, "(p \cdot q)(i)"] \arrow[ddd, equal] \arrow[dr, equal] & & & c \arrow[dl, "r(j)"'] \\
& a \arrow[r, "((p \cdot q) \cdot r)(i)"] \arrow[d, equal] & d & \\
& a \arrow[r, "p(i)"'] & b \arrow[u, "(q \cdot r)(k)"] \arrow[dr, equal] & \\
a \arrow[rrr, "p(i)"'] \arrow[ur, equal] & & & b \arrow[uuu, "q(j)"']
\end{tikzcd}$$



The groupoid laws

Associativity



The groupoid laws…

For path of types