

Strictly Associative Group Theory using Univalence

Alex Rice¹

University of Cambridge

HoTT/UF 2023



UNIVERSITY OF
CAMBRIDGE


```
InvUniqueLeft :  $\forall \{\ell\} (\mathcal{G} : \text{Group } \ell) \rightarrow \text{Type } \ell$   
InvUniqueLeft  $\mathcal{G} = \forall g\ h \rightarrow h \cdot g \equiv 1g \rightarrow h \equiv \text{inv } g$   
  where  
  open GroupStr ( $\mathcal{G} . \text{snd}$ )
```

Motivation

$\text{InvUniqueLeft} : \forall \{\ell\} (\mathcal{G} : \text{Group } \ell) \rightarrow \text{Type } \ell$
 $\text{InvUniqueLeft } \mathcal{G} = \forall g \ h \rightarrow h \cdot g \equiv 1g \rightarrow h \equiv \text{inv } g$

where

open GroupStr ($\mathcal{G} \text{.snd}$)

$\text{inv-unique-left} : \forall \{\ell\} (\mathcal{G} : \text{Group } \ell) \rightarrow \text{InvUniqueLeft } \mathcal{G}$

$\text{inv-unique-left } \mathcal{G} \ g \ h \ p =$

$h \equiv \langle \text{sym } (\cdot \text{IdR } h) \rangle$

$h \cdot 1g \equiv \langle \text{cong } (h \cdot _) (\text{sym } (\cdot \text{InvR } g)) \rangle$

$h \cdot (g \cdot \text{inv } g) \equiv \langle \cdot \text{Assoc } h \ g \ (\text{inv } g) \rangle$

$(h \cdot g) \cdot \text{inv } g \equiv \langle \text{cong } (_ \cdot \text{inv } g) \ p \rangle$

$1g \cdot \text{inv } g \equiv \langle \cdot \text{IdL } (\text{inv } g) \rangle$

$\text{inv } g \quad \square$

where

open GroupStr ($\mathcal{G} \text{.snd}$)

Motivation

```
InvUniqueLeft :  $\forall \{\ell\} (\mathcal{G} : \text{Group } \ell) \rightarrow \text{Type } \ell$   
InvUniqueLeft  $\mathcal{G} = \forall g\ h \rightarrow h \cdot g \equiv 1g \rightarrow h \equiv \text{inv } g$   
  where  
  open GroupStr ( $\mathcal{G} \text{ .snd}$ )
```

```
inv-unique-left-strict :  $\forall \{\ell\} (\mathcal{G} : \text{Group } \ell) \rightarrow \text{InvUniqueLeft } \mathcal{G}$   
inv-unique-left-strict  $\mathcal{G} = \text{strictify InvUniqueLeft}$   
   $\lambda g\ h\ p \rightarrow$   
     $h \cdot 1g \equiv \langle \text{cong } (h \cdot \_) (\text{sym } (\cdot \text{InvR } g)) \rangle$   
     $h \cdot g \cdot \text{inv } g \equiv \langle \text{cong } (\_ \cdot \text{inv } g) p \rangle$   
     $1g \cdot \text{inv } g \quad \square$   
  where  
    open GroupStr (RSymGroup  $\mathcal{G} \text{ .snd}$ )  
    open import Groups.Reasoning  $\mathcal{G}$  using (strictify)
```

- Given a group \mathcal{G} , we create a new group `RSymGroup` \mathcal{G} .

Theorem (Cayley's Theorem)

Every group is isomorphic to a subgroup of a symmetric group.

- In `RSymGroup` \mathcal{G} , various rules hold by reflexivity.
- We show that `RSymGroup` \mathcal{G} is isomorphic to \mathcal{G} .
- By univalence and the structure identity principle, `RSymGroup` \mathcal{G} is equal to \mathcal{G} .
- The `strictify` function transports a proof from `RSymGroup` \mathcal{G} back to \mathcal{G} .

In the strictified group the following equations hold definitionally:

- $a(bc) = (ab)c,$
- $a1 = a = 1a,$
- $a^{-1^{-1}} = a,$
- and $(fg)^{-1} = g^{-1} \cdot f^{-1}.$

Theorem (Cayley's Theorem)

Every group is isomorphic to a subgroup of a symmetric group.

Functions compose strictly

Theorem (Cayley's Theorem)

Every group is isomorphic to a subgroup of a symmetric group.

$$\begin{aligned} _ \circ _ &: (f : B \rightarrow C) \rightarrow (g : A \rightarrow B) \rightarrow (A \rightarrow C) \\ (f \circ g) \ x &= f \ (g \ x) \end{aligned}$$

$$\begin{aligned} \text{comp-assoc} &: (f : C \rightarrow D) \\ &\rightarrow (g : B \rightarrow C) \\ &\rightarrow (h : A \rightarrow B) \\ &\rightarrow f \circ (g \circ h) \equiv (f \circ g) \circ h \end{aligned}$$

$$\text{comp-assoc } f \ g \ h = \text{refl}$$

Do invertible functions compose strictly?

```
record Inverse (A : Type) (B : Type) : Type where
  field
    ↑ : A → B
    ↓ : B → A
    ε : ∀ x → ↓ (↑ x) ≡ x
    η : ∀ y → ↑ (↓ y) ≡ y
```

Strict invertible functions

```
record Inverse (A : Type) (B : Type) : Type where
  constructor [_,_,_,_]
  field
    ↑ : A → B
    ↓ : B → A
    ε : ∀ b {x} → x ≡ ↓ b → ↑ x ≡ b
    η : ∀ a {y} → y ≡ ↑ a → ↓ y ≡ a
```

```
_o_ : Inverse B C → Inverse A B → Inverse A C
_o_ [ f , g , p , q ] [ f' , g' , p' , q' ] =
  [ (λ x → f (f' x)) ,
    (λ y → g' (g y)) ,
    (λ b r → p b (p' (g b) r)) ,
    (λ a r → q' a (q (f' a) r)) ]
```

Strict invertible functions

$\text{assoc} : (f : \text{Inverse } C \ D)$
 $\rightarrow (g : \text{Inverse } B \ C)$
 $\rightarrow (h : \text{Inverse } A \ B)$
 $\rightarrow f \circ (g \circ h) \equiv (f \circ g) \circ h$

$\text{assoc } f \ g \ h = \text{refl}$

$\text{id-inv} : \text{Inverse } A \ A$

$\text{id-inv} = \llbracket (\lambda x \rightarrow x) , (\lambda x \rightarrow x) ,$
 $(\lambda b \ r \rightarrow r) , (\lambda a \ r \rightarrow r) \rrbracket$

$\text{id-unit-left} : (f : \text{Inverse } A \ B)$
 $\rightarrow \text{id-inv} \circ f \equiv f$

$\text{id-unit-left } f = \text{refl}$

$\text{id-unit-right} : (f : \text{Inverse } A \ B)$
 $\rightarrow f \circ \text{id-inv} \equiv f$

$\text{id-unit-right } f = \text{refl}$

Strict invertible functions

`inv-inv` : `Inverse A B` \rightarrow `Inverse B A`

`inv-inv` $\llbracket f, g, \varepsilon, \eta \rrbracket = \llbracket g, f, \eta, \varepsilon \rrbracket$

`inv-involution` : $(f : \text{Inverse } A \ B)$
 $\rightarrow \text{inv-inv } (\text{inv-inv } f) \equiv f$

`inv-involution` $f = \text{refl}$

`inv-comp` : $(f : \text{Inverse } B \ C)$
 $\rightarrow (g : \text{Inverse } A \ B)$
 $\rightarrow \text{inv-inv } (f \circ g) \equiv \text{inv-inv } g \circ \text{inv-inv } f$

`inv-comp` $f \ g = \text{refl}$

Representable functions

The map $\iota : g \mapsto g \cdot _$ includes the group \mathcal{G} in the symmetric group. We now want to restrict the symmetric group to those functions that are in the image of ι .

Proposition

*A function $f : \mathcal{G} \rightarrow \mathcal{G}$ is in the image of ι if and only if for all $g, h \in \mathcal{G}$,
 $f(g \cdot h) = f(g) \cdot h$.*

Representable functions

The map $\iota : g \mapsto g \cdot _$ includes the group \mathcal{G} in the symmetric group. We now want to restrict the symmetric group to those functions that are in the image of ι .

Proposition

A function $f : \mathcal{G} \rightarrow \mathcal{G}$ is in the image of ι if and only if for all $g, h \in \mathcal{G}$, $f(g \cdot h) = f(g) \cdot h$.

`Representable : Inverse $\langle \mathcal{G} \rangle \langle \mathcal{G} \rangle \rightarrow \text{Type}$`

`Representable f = $\forall x g h \rightarrow x \equiv g \cdot h \rightarrow \uparrow f x \equiv \uparrow f g \cdot h$`

`Repr : Type`

`Repr = $\Sigma [f \in \text{Inverse } \langle \mathcal{G} \rangle \langle \mathcal{G} \rangle] \text{Representable } f$`

Representable symmetric group

- Let $\text{RSymGroup } \mathcal{G}$ be the subgroup of the symmetric group on \mathcal{G} consisting of those functions that are representable.

Representable symmetric group

- Let $\text{RSymGroup } \mathcal{G}$ be the subgroup of the symmetric group on \mathcal{G} consisting of those functions that are representable.
- This subgroup still has strict composition.

Representable symmetric group

- Let $\text{RSymGroup } \mathcal{G}$ be the subgroup of the symmetric group on \mathcal{G} consisting of those functions that are representable.
- This subgroup still has strict composition.
- The inclusion ι is an isomorphism from \mathcal{G} to the representable symmetric group.

Representable symmetric group

- Let $\mathbf{RSymGroup} \mathcal{G}$ be the subgroup of the symmetric group on \mathcal{G} consisting of those functions that are representable.
- This subgroup still has strict composition.
- The inclusion ι is an isomorphism from \mathcal{G} to the representable symmetric group.
- By univalence we get an equality:

$$\iota \equiv \mathcal{G} : \mathcal{G} \equiv \mathbf{RSymGroup} \mathcal{G}$$

Representable symmetric group

- Let `RSymGroup` \mathcal{G} be the subgroup of the symmetric group on \mathcal{G} consisting of those functions that are representable.
- This subgroup still has strict composition.
- The inclusion ι is an isomorphism from \mathcal{G} to the representable symmetric group.
- By univalence we get an equality:

$$\iota \equiv \mathcal{G} : \mathcal{G} \equiv \text{RSymGroup } \mathcal{G}$$

- This lets us define:

```
strictify : (G : Group ℓ-zero)
           → (P : Group ℓ-zero → Type)
           → P (RSymGroup G)
           → P G
strictify G P p = transport (sym (cong P (ι ≡ G))) p
```

Further thoughts

Further thoughts

Does this all work with categories instead of groups?

- For each group \mathcal{G} we can generate an isomorphic group `RSymGroup` \mathcal{G} .
- This group has nice definitional properties
- Univalence allows us to generate an equality between the two groups.
- This allows us to prove theorems about an arbitrary group by instead proving them on the strictified group.
- <https://alexarice.github.io/posts/sgtuf/Strict-Group-Theory-UF.html>