(o ((curry map) f) ((curry map) g)) == ((curry map) (o f g))

Must Prove:

((o ((curry map) f) ((curry map) g)) xs) == (((curry map) (o f g) xs)

LHS: ((o curry map) f) ((curry map) g)) xs) ~~[illegible]~~

{ apply compose law }

(((curry map) f) (curry map) (g xs))

{ apply curried law }

= (map f (((curry map) g) xs))

= (map f (map g xs))

RHS:

((((curry map) (o f g)) xs)

= { apply curry }

(map (o f g) xs)

For base case where xs = '()

LHS: (map f (map g '()))

{ substitute }

(if (null? '())

'()

(cons (g (car '()) (map g (cdr '()))))

= { null? - empty law }

(if #t

'()

(cons (g (car '())) (map g (cdr '())))))

{ #t law }

'()

RHS: (map (o f g) '())

{ substitute }

(if (null? '())

'()

(cons ((o f g) (car '())) (map (o f g) (cdr '())))

= { null empty law }

⟶

if (#t)
  '()
  (cons ((o f g) (car '()))(map (o f g)(cdr '())))
  = if #t law
    '()

LHS and RHS both equal '()

Inductive step, prove for any xs, where xs = (cons y ys)

LHS (map f (map g xs)
  {substitute}
  map f (if null? xs)
    '()
    (cons (g (car xs))(map g (cdr xs)))
  ~~{sub xs = (cons y ys)}~~
  ~~map f (if #f~~
    '()
    (cons (g (car (cons y ys)))(map g (cdr (cons y ys)))
  ={if #f law}
  map f cons (g (car (cons y ys)))(map g (cdr (cons y ys))))
  = {car cons law}
  ~~cons (g y) (map g (cdr (cons y ys)))~~
  = {cdr, cons law}
  map f cons (g y)(map g ys)
  (if (null? (cons (g y) (map g ys)))
    '()
    (cons (f (car (cons (g y) (map g ys)))
      (map f (cdr (cons (g y) (map g ys)))
  = {null cons law}
  (if #f
    '()
    (cons (f (car (cons (g y) (map g ys)))
      (map f (cdr (cons (g y) (map g ys)))

= if #f law
(cons (g (car (cons (g y) (map g ys)))
(map f (cdr (cons (g y) (map g ys)))))
= {car-cons law}
(cons (f (g y))
(map f ((cdr (cons (g y) (map g ys)))))
= {cdr-cons law}
((cons (f (g y))) (map g ys)))
= induction hypothesis
(cons (f (g y)) (map (o f g) ys)

RHS: (map (o f g) xs)
= {substitute}
(if (null? xs)
'()
(cons ((o f g (car xs)) (map (o f g) (cdr xs))))
= null-cons law
(if #f
'()
(cons ((o f g) (car (cons x ys))) (map (o f g) ((cdr (cons x ys)))
= {if #f law}
'()
(cons ((o f g) (car (cons y ys)))) (cons (o f g) (cdr (cons y ys)))
= {car-cons law}
(cons ((o f g) y) (map (o f g) (cdr (cons y ys)))
= {cdr-cons law}
(cons (((o f g) y) (map (o f g) ys))
= apply compose
(cons (f (g y)) (map (o f g) ys))

LHS = RHS thus by induction the
statement is proven