M.) To Prove that

$$(\circ\ ((\text{curry map})\ f)\ ((\text{curry map})\ g)) == ((\text{curry map})\ (\circ\ f\ g))$$

1.) Base Case – where each function is applied to '()

Left hand side of Proof

$$((\circ\ ((\text{curry map})\ f)\ ((\text{curry map})\ g))\ '())$$

= { apply - compose law }

$$(((\text{curry map})\ f)\ (((\text{curry map})\ g)\ '()))$$

= { apply - curried law }

$$(((\text{curry map})\ f)\ (\text{map}\ g\ '()))$$

= { map - nil law }

$$(((\text{curry map})\ f)\ '())$$

= { apply - curried law }

$$(\text{map}\ f\ '())$$

= { map - nil law }

= '()

Right hand side of the Proof

$$(((\text{curry map})\ (\circ\ f\ g))\ '())$$

= { apply - curried law }

$$(\text{map}\ (\circ\ f\ g)\ '())$$

= { map - nil law }

= '()

So,

As we use same argument on each side it returns the same thing so the Base case holds on empty list

Scanned with CamScanner

We apply non-empty list ys = (cons x xs)

<u>Left hand side of the Proof</u>

((o ((curry map) f) ((curry map) g)) ys)

= { substituting ys with (cons x xs)}

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

= { apply - compose law }

(((curry map) f) (((curry map) g) (cons x xs)))

= { apply - curried law }

(((curry map) f) (map g (cons x xs)))

= { map - cons law }

(((curry map) f) (cons (g x) (map g xs)))

= { apply - curried law }

(map f (cons (g x) (map g xs)))

= { map - cons law }

(cons (f (g x)) (map f (map g xs)))

= { apply - curried law }

(cons (f (g x)) (map f (((curry map) g) xs)))

= { apply - curried law }

(cons (f (g x)) (((curry map) f) (((curry map) g) xs)))

= { apply - compose law }

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

{

= { By induction hypothesis }

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

= { apply - compose law }

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

= { apply - curried law }

(cons ((o f g) x) (map (o f g) xs))

= { map - cons law }

(map (o f g) (cons x xs))

= { replace (cons x xs) with ys }

(map (o f g) ys)

= { ~~applyd~~ apply - curried law }

(((curry map) (o f g)) ys)

As we notice

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

for any ys we can say that it is true for any list.

Scanned with CamScanner