Theory

Reasoning about Calculational Proofs

Prove that `(o ((curry map) f) ((curry map) g)) == ((curry map) (o f g))`
This means we need to show that
`((o ((curry map) f) ((curry map) g)) xs) == (((curry map) (o f g)) xs)`
Thus, we will use induction to show that the two functions are equivalent.

Base Case

Assume that the given list to the function is empty. Then xs = ’().

`((o ((curry map) f) ((curry map) g)) xs)`
$=$ {By assumption that xs is empty}
`((o ((curry map) f) ((curry map) g)) ’())`
$=$ {By apply-compose law}
`(((curry map) f) (((curry map) g) ’()))`
$=$ {By apply-curry law}
`(((curry map) f) (map g ’()))`
$=$ {By map-nil law}
`(((curry map) f) ’())`
$=$ {By apply-curry law}
`(map f ’())`
$=$ {By map-nil law}
`’()`
$=$ {By map-nil}
`(map (o f g) ’())`
$=$ {By apply-curry law}
`(((curry map) (o f g)) ()’)`
$=$ {By assumption that xs is empty}
`(((curry map) (o f g)) xs)`

Inductive Case

Let xs be a non-empty list xs = (cons y ys) and assume that the inductive hypothesis

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

is true for all lists with length shorter than xs (aka that it is true for ys)

```
((o ((curry map) f) ((curry map) g)) xs)
```
= {By apply-compose law}
```
(((curry map) f) (((curry map) g) xs))
```
= {By apply-curry law}
```
(((curry map) f) (map g xs))
```
= {By apply-curry law}
```
(map f (map g xs))
```
= {By xs = (cons y ys)}
```
(map f (map g (cons y ys)))
```
= {By map-cons law}
```
(map f (cons (g y) (map g ys)))
```
= {By map-cons law}
```
(cons (f (g y)) (map f (map g ys)))
```
= {By apply-curry law}
```
(cons (f (g y)) (map f (((curry map) g) ys)))
```
= {By apply-curry law}
```
(cons (f (g y)) (((curry map) f) (((curry map) g) ys)))
```
= {By apply-compose law}
```
(cons (f (g y)) ((o ((curry map) f) ((curry map) g)) ys))
```
= {By inductive hypothesis}
```
(cons (f (g y)) (((curry map) (o f g)) ys))
```
= {By appy-curry law}
```
(cons (f (g y)) (map (o f g) ys))
```
= {By apply-compose law}
```
(cons ((o f g) y) (map (o f g) ys))
```
= {By map-cons law}
```
(map (o f g) (cons y ys))
```
= {By xs = (cons y ys)}
```
(((curry map) (o f g)) xs)
```

Thus, by induction,
`((o ((curry map) f) ((curry map) g)) xs) == (((curry map) (o f g)) xs)`
So the functions are observationally equivalent, so
`(o ((curry map) f) ((curry map) g)) == ((curry map) (o f g))`