2013年2月28日
11:38

$$h \cdot foldr\ f\ e = foldr\ f'\ e'$$
$$\Leftarrow h\ e = e' \wedge h\ (f\ x\ y) = f\ x\ (h\ y)$$

```
class Monad m => MonadState s m where
    get :: m s
    put :: s -> m ()
```

```
put s >>= \() -> put s'   = put s'
put s >>= \() -> get      = put s >>= \() -> return s
get >>= \s -> put s       = return ()
get >>= \s -> get >>= \s' -> k s s'
                          = get >>= \s -> k s s
```

```
add :: MonadState Integer m =>
       Integer -> m ()
add n = do {m <- get; put (m+n)}

addAll :: ... => [Integer] -> m ()
addAll = sequence_ . map add
       = add . sum
```