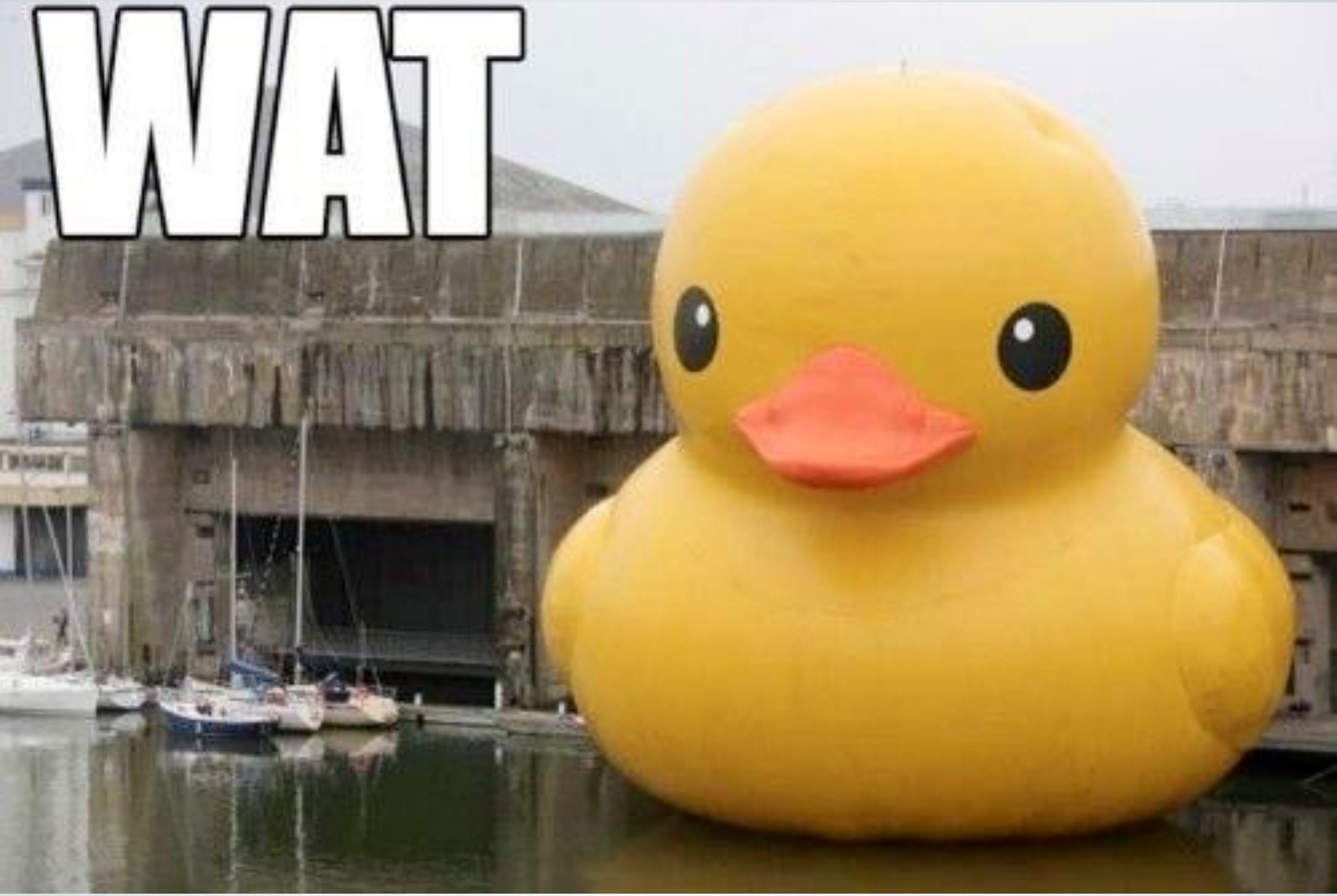
MONOIDS

Practical uses

source code & slides: https://github.com/Batou99/monoids_mad

WHAT IS A MONOID?

- An algebraic structure
- With a binary operation
  - associative
  - closed
  - With an identity element


WAT

ALGEBRAIC STRUCTURE

- Made by elements like
  - Numbers
  - Characters
  - Objects
  - Functions
  - …

BINARY OPERATION

- CLOSED: op :: a -> a -> a
- ASSOCIATIVE:(a `op` b) `op` c = a `op` (b `op` c)
- With identity element: e
  - a `op` e = a
  - e `op` a = a

SEMIGROUP

- It's a monoid WITHOUT identity element


MY WORK HERE
IS DONE
memegenerator.net

HOW ABOUT AN

example?

SOME EXAMPLES

Set	op	e
Integer	+	0
Integer	*	1
[a]	++	[]
Bool	&&	True


WHY THE FUCK
DO I WANT MONOIDS?!
imgflip.com

BENEFITS

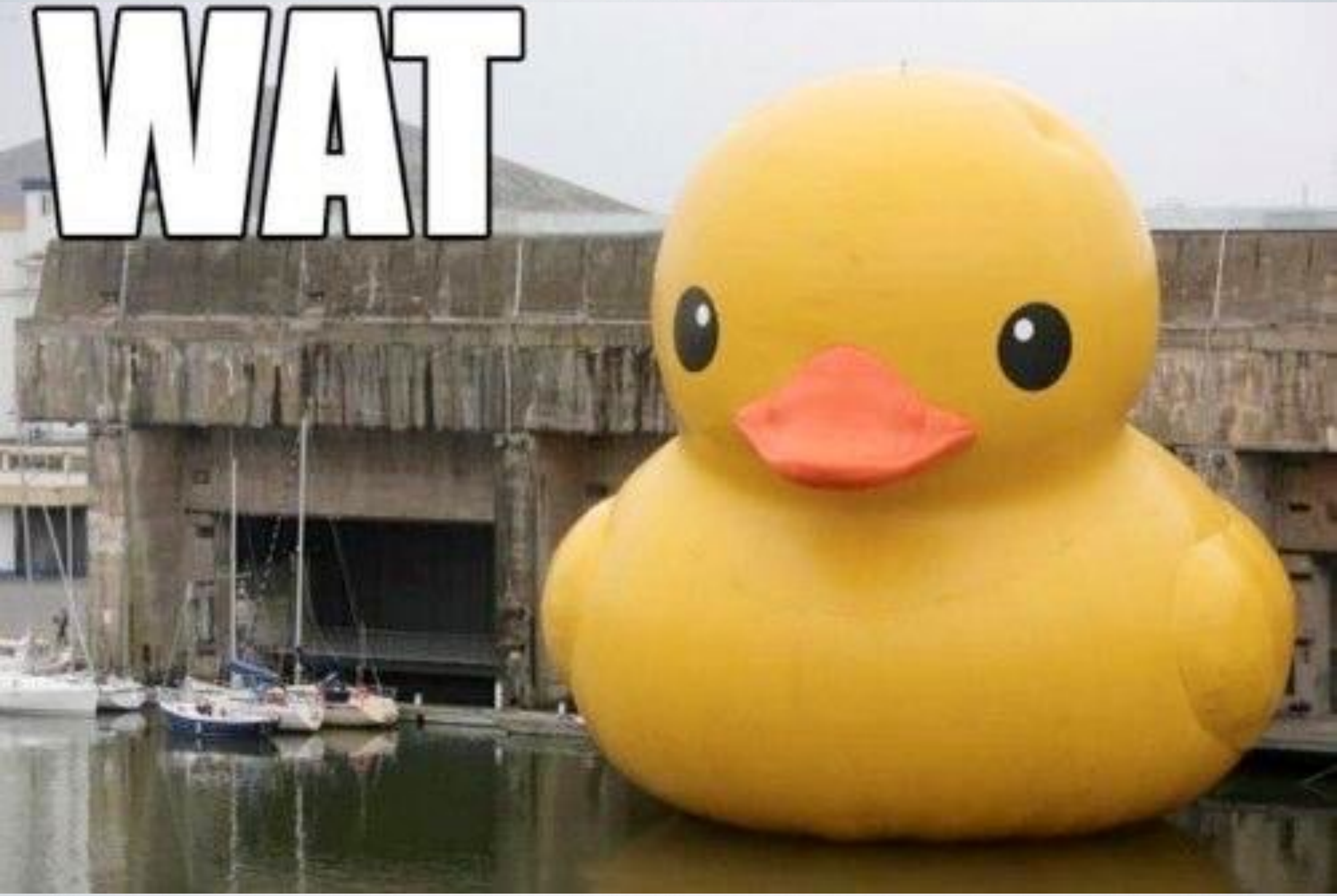
- Closed operation: fold
- Associative Op
- Divide and Conquer
- Parallelising
- Incremental calculations

A TYPICAL EXAMPLE

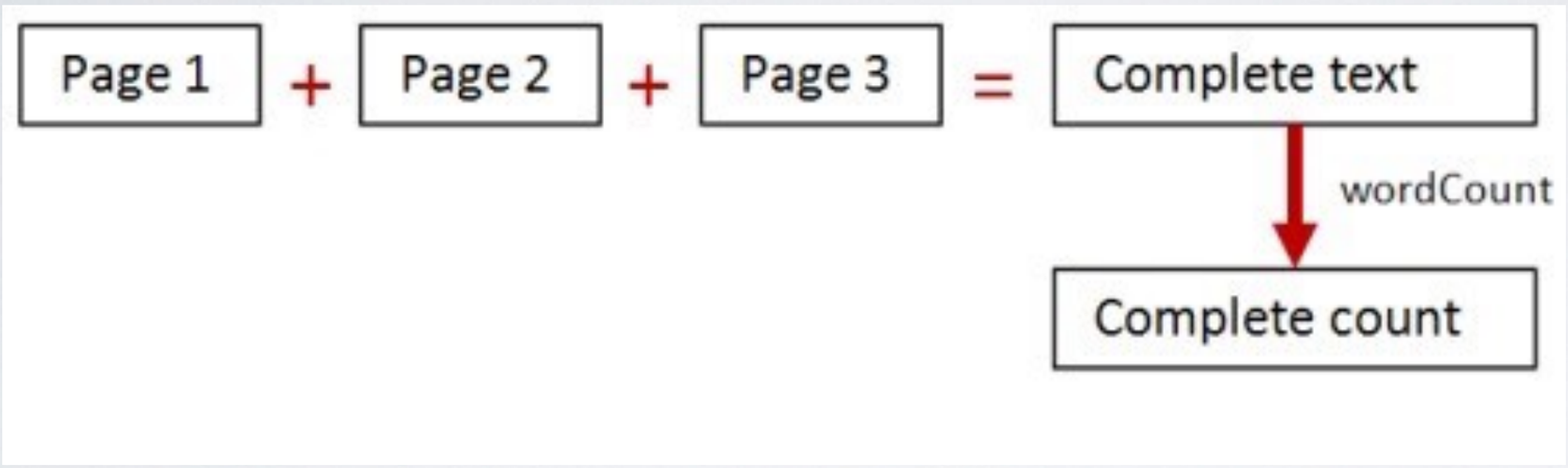
- Invoices!!
- Purchase lines
- Totals y subtotals

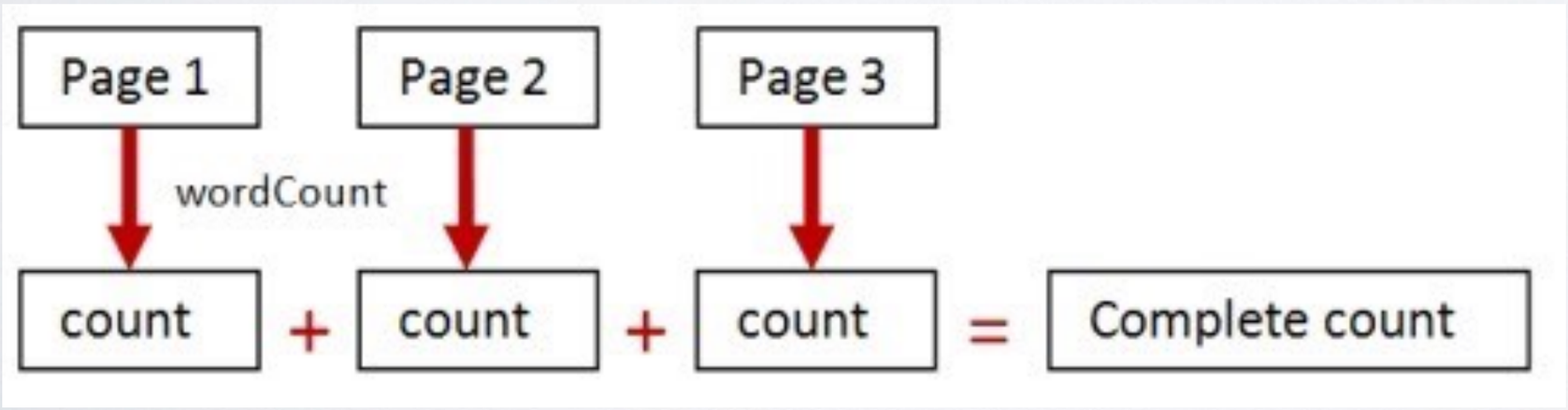
HOMOMORPHISMS

- f is an homomorphism if
  - f(a op b) = f(a) op f(b)


WAT

AN EXAMPLE

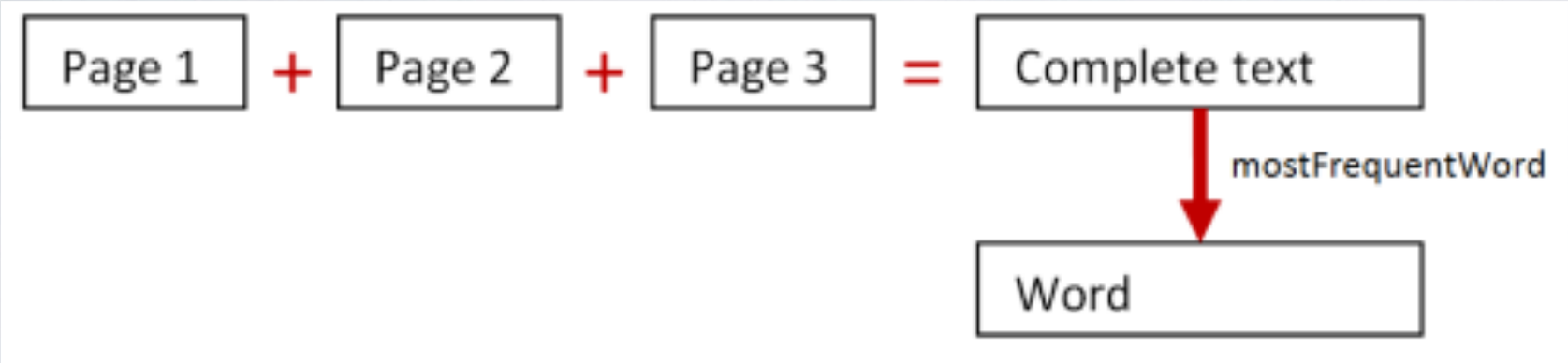

Page 1
+
Page 2
+
Page 3
=
Complete text
wordCount
Complete count

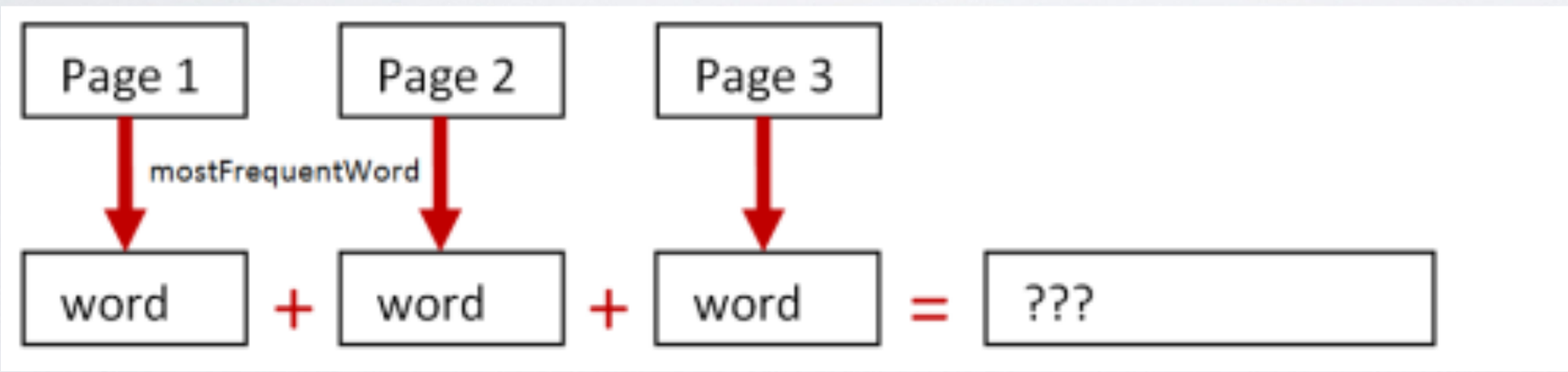

Page 1
Page 2
Page 3
wordCount
count
+
count
+
count
=
Complete count

QUESTION

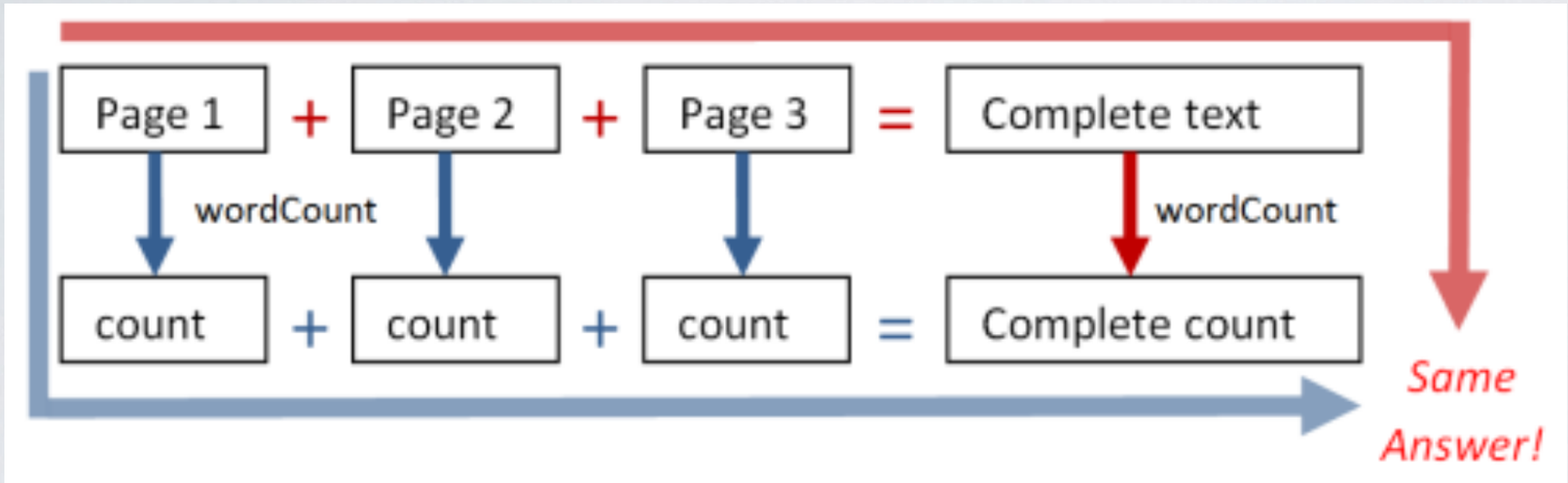
- All functions where:
- Monoid a, Monoid b => f :: a -> b
- Are homomorphisms?

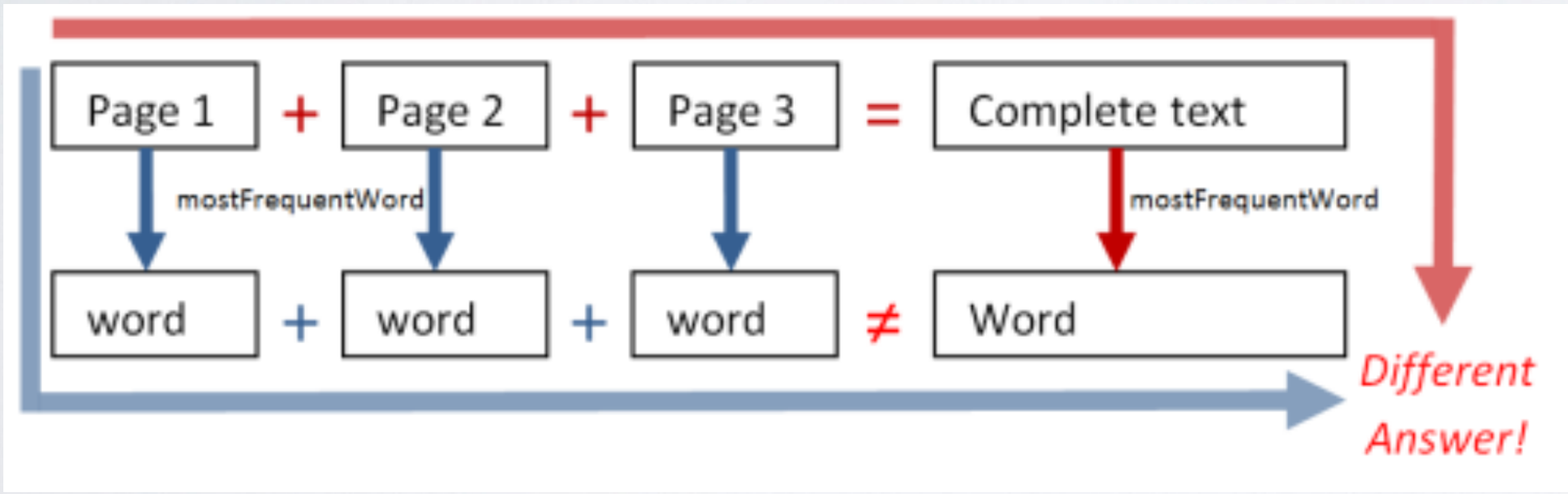
NO


Page 1
+
Page 2
+
Page 3
=
Complete text
mostFrequentWord
Word


Page 1
Page 2
Page 3
mostFrequentWord
word
+
word
+
word
=
???

REVIEW


Page 1
+
Page 2
+
Page 3
=
Complete text
wordCount
wordCount
count
+
count
+
count
=
Complete count
Same
Answer!


Page 1
+
Page 2
+
Page 3
=
Complete text
mostFrequentWord
mostFrequentWord
word
+
word
+
word
≠
Word
Different
Answer!

HOW TO WORK WITH NON MONOIDAL TYPES

- If we work with non monoidal types we need to massage them until they are

NON CLOSED OPERATION

- Extract some information (OrderLine -> TotalLine)

- Wrap with lists (a -> [a])

NON ASSOCIATIVE OPERATION

- Move the operation into the type

NO IDENTITY ELEMENT

- Create one

```
type OrderLine =
    | Product of ProductLine
    | Total of TotalLine
    | EmptyOrder
```

- Wrap with Maybe

SUMMARY

- Monoids advantages:
  - More efficient algos
  - Uniform type behaviour
  - Show intent better

CREDITS

- Scott Wlaschin
- Monoids without tears (fsharpforfunandprofit.com)