An Algebraic Approach to Programming

$$1 + 1 = ?$$

$$1 + 1 + 0 + 0 + 0 = 1 + 1$$

$$1 * 1 * 1 * 2 = 2$$

$$2^3 + 2^2 + 2^1$$

Independent Value in Term

$$1 / (1 / 1)$$

$$1 / (1 / 2)$$

$$1 / (1 / 3)$$

$$1 / (1 / x)$$

Dependent Value in Term

$$1 + 2 + 3 + (4 + 5) = 15$$

$$1 + 2 + (3 + 4) + 5 = 15$$

$$1 + (2 + 3) + 4 + 5 = 15$$

Associative Across

$$1 + 1/(1/(1/(1/1))) = 1 + 1$$

Associative in Depth

$$1 + 1/(1/(1/1)) + 1$$

$$[1] + [[1]/[[1]/[[1]/[1]]]] + [1]$$

$$[] \otimes [[]\otimes[[]\otimes[[]\otimes[]]]] \otimes []$$

Infinite List of Trees of Trees

Infinitely Long

$$[] \otimes [] \otimes [] \otimes [] \otimes []$$

Where $\otimes$ is a binary operation
This is a Monoid

Infinitely Deep

$$[[[] \otimes []] \otimes [[] \otimes []]]$$

Where $\otimes$ is a binary operation

This is a **Monad**

$$[1] + [2] + [3] + [4] = [10]$$

Map & Fold(aka Reduce) over Each Term

x	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

Map

$$f(x) = y$$

For every input there is only one unique output

Monoid

[1,2,3,4]

Monoid

$$2^3 + 2^2 + 2^1 = 14$$

Functional Programming can be think of as composing polynomial equation