

Reduce and map

(in Backus's FP)

! f (read: insert f) In Haskell: foldr ...
Google calls this reduce

$$!f: \langle x_1, \dots, x_n \rangle = \begin{cases} f: \langle x_1, x_2 \rangle & \text{if } n=2 \quad \leftarrow \\ f: \langle x_1, !f: \langle x_2, \dots, x_n \rangle \rangle & \text{if } n > 2 \quad \leftarrow \\ \text{undefined otherwise} & \end{cases}$$

Ex. Let f be plus
plus: $\langle 2, 5 \rangle = 7$

$$\begin{aligned} !\text{plus}: \langle 2, 5, 6, 8 \rangle &= \text{plus}: \langle 2, !\text{plus}: \langle 5, 6, 8 \rangle \rangle = \\ &= \text{plus}: \langle 2, \text{plus}: \langle 5, !\text{plus}: \langle 6, 8 \rangle \rangle \rangle = \text{plus}: \langle 2, \text{plus}: \langle 5, \text{plus}: \langle 6, 8 \rangle \rangle \rangle = \\ &= \text{plus}: \langle 2, \text{plus}: \langle 5, 14 \rangle \rangle = \text{plus}: \langle 2, 19 \rangle = 21 \end{aligned}$$

λf ^{FP}
apply-to-all f

Haskell, Lisp, "Google"
map

$$\lambda f: \langle x_1, \dots, x_n \rangle = \langle f x_1, \dots, f x_n \rangle$$