DOMAIN: LIST PROCESSING

Reverse

Dreamcoder takes in list processing tasks...

From a set of starting primitives...

It invents a DSL of new complex functions to filter, take the maximum, and others that build on its primitives and learned functions.

3s are in list $[9\ 2\ 4] \rightarrow [4\ 2\ 9] \qquad [3\ 1\ 3\ 3\ 1\ 9\ 2\ 3] \rightarrow \ [9\ 2\ 3] \rightarrow \ [2\ 3\ 9]$ $[4 \ 3] \rightarrow [3 \ 4]$ $[3 \ 3 \ 2 \ 3] \rightarrow 3$ $[8 7 9 2 5] \rightarrow [2 5 7 8 9]$ $[9 \ 8 \ 9] \rightarrow [8 \ 9 \ 9]$ $[3\ 4\ 5\ 6] \rightarrow [6\ 5\ 4\ 3] \qquad [3\ 1\ 2\ 3] \rightarrow \ 2$ mod map unfold fold index range if is-prime car cdr empty empty? cons length ≥? ea? is-square the number '3' f4 = (+1 (+1 1)) $f9=(\lambda (x y) (fold)$ $f2=(\lambda (x) (cdr)$ x (cons y empty) (cdr x))) $(\lambda (z u) (cons z)$ u)))) the number '4' f5 = (+1 f4) $f10=(\lambda(x y))$ (fold f0 = $f8 = (\lambda (x y) (f0 x)$ $(f2(f2 x)) y (\lambda$ $(\lambda(x y) \text{ (fold } x)$ $(\lambda(z) (is-prime(+ y$ filter (z u) (cdr (f9 u empty $(\lambda(z u))$ (if (y (mod z f5)))))) (car u)))))) z)(cons z u)u)))) $f14=(\lambda (x y) (map)$ $(\lambda(x) (car(f0 x(\lambda(y)$ $(\lambda (z) (index z)$ $f11 = (\lambda(x y) (length)$ (empty?($f0 \times (\lambda(z))$ maximum of list x))(range (y (+ $(f0 \times (\lambda(z)))$ ≥? x y))))))) 1))))) у))))) $f13 = (\lambda (x y) (f12)$ $f15 = (\lambda (x y))$ $f1=(\lambda (x) (map (\lambda$ (f0 $x(\lambda(z)) \ge y$ (y) (eq? (mod y (fold x y (λ (z $(length(f0 x(\lambda(u)$ x) 0)))) u) (cons z u)))) $(\geq ? z u))))))))))$ (fold(empty($\lambda(x)$ ($\lambda(y)$) ($\lambda(x)$ (length($f_0(x)$ (λ) $(\lambda (map (\lambda (f13 (x y) z (+ u))))$ (y) (eq? f4 y)))) 1)))(range(length u)))) (f9(x y))

Count how many

Sort

...to output **programs** in its learned DSL that solve the tasks.