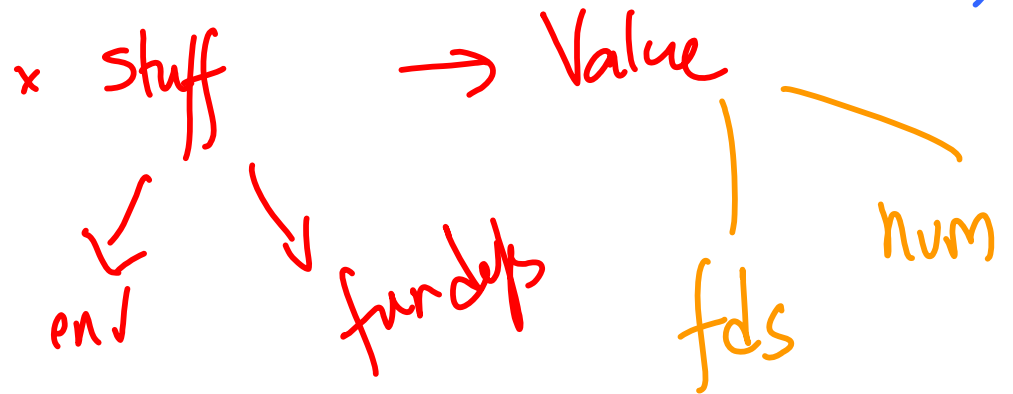


(interp <fac>) →
 a function
 (interp (plus C
 (num C)
 <fac>))

Answer / Value

interp : Program



(define-type ExprC

[numC (n : number)]

[idC (s : symbol)]

[appC (fun : ExprC) (arg : -)]

[plusC (l : ExprC) (r : -)])

~~(define-type FunDefC~~

~~[lamC~~

~~(arg : symbol)~~

~~(body : ExprC)])~~

(+ (define (f x) (* x x)))

4)

(+ (define (f x) (* x +1)))

4)

(3 4)

define THREE 3

f(3) { [(\lambda (x)

(\lambda (y)

(+ x y)))

5 } 4]

(define (interp [expr : ExprC] [env : Env] ~~[fds : (listof FunDefC)]~~)
: ~~number~~ Value

[fdC (f a b) (fun ^{kind of value} expr)]

[lamC (a b)

(closure V a b env)]

∴
[appc (f a)

(local ([define fd (interp f env)]))

(interp (clos-body fd)

(extend-env (bind (clos-arg fd)

(interp a env))

)]])

~~(clos-env fd)~~ env