

① Violates intuition about "looking up"

[id (x) Error]

② Efficiency?

③ Keep source text around

(define (f x)

(+ (- (* x) 1) 2) 3))

(f 2)

(f 3)

PETER SUBSTITUTION

subst : number or symbol x Expr C \rightarrow Expr C

①

②

③
x

BINDING [symbol \rightarrow number

ENVIRONMENT

(define-type Env

[mt]

[binding (x : symbol?)

(n : number?)

(rest : Env?)])

interp : ExprC x list fds x Env \rightarrow number

(define (interp e fds env)

[numC (n) n]

[plusC (l r) (+ (interp l fds env) (interp r fds env))]

[idC (x) (lookup x env)]

[appC (f a) (local ([define fd (get-fd f fds)])

(interp (fd-body fd) fds

(binding (fd-arg fd)

(interp a fds env)

mt **env**)))]

lookup: symbol x Env \rightarrow number

(define (lookup x env)

(type-case env

[mt (error "Unbound")]

[binding (y n rest)

(if (symbol=? x y)

n

(lookup x rest))))))

interp-subst [?] = interp-env

(define (f x) y) (binding x 2 (binding y 3 mt))

(define (g x) (f 2)) (binding y 3 mt)

(g 3)

mt

$\frac{\text{error}}{S} \neq \frac{\text{error?}}{E}$

(define (f x) x)

(define (g x) (f 4))

(g 5)

$\frac{4}{S} = \frac{4}{E}$

interp-subst
interp-env-fixed

interp-env

LEXICAL SCOPE

DYNAMIC SCOPE

