

Conceitos 9/4/18

(126)

F1WAE  $\rightarrow$  FWAE

{ with { 'f' { fun 'x' { add { num 2 } { id 'x' } } } } }  
{ app { id 'f' } { num 4 } }

{ with { 'name' valor } expr }  
{ app { fun name expr } valor }

Exemplo:

{ app with { ('x') { num 3 } } { add { id 'x' } { num 2 } } }  
{ app { fun ('x') { add { id 'x' } { num 2 } } } { num 3 } }

AE  
↓  
WAE

F1WAE  $\rightarrow$  F1WAE'

FWAE

F<sup>↓</sup>AE

FAE'

closure

(define (interp expr)  
 [num (n) expr])

(num 4) → (num 4)

(add (num 2) (num 3))

( $\oplus$  (num 2) (num 3))  
 ↗ add-number

Expr

(app fun × (add {id 'x} {num 1})) (num 41)  
arg-expr                                    fun-expr

arg-expr      param      body

f → (fun  $\oplus$  Tadd {id 'x} {num 1})

⇒ (subst (add {id 'x} {num 1})  $\times$  (num 41),  
 (add (num 41) (num 1)),  
 (add-number (num 41) / num 1))

Conceitos

• with

{ x 3 }

{ with f }

{ with (x)s }

escopo dinâmico

(128)

{ fun().f }

{ f 4 } }

{ + Q }

y } }

y } }

9 //

Solução → closure

def função contém as associações

(closure V 'y (add { id 'x } { id 'y }) { x (numV 3) (intSub) })

(intSub { add { id 'x } { id 'y } } (adub 'y { num 4 } (adub 'x { num 3 } (intSub))) → (numV 7)

(define h 0.000001)

(define d/dx f)

(lambda (x))

( / ( -

(+ x h) (f x) h))

(define (difraiz x) (d/dx (+ x x))) → ver livro.

pg 50

(define difraiz [ (d/dx (lambda (x) (\* x x))) ])

f