John Spicer

CS 496

Professor Bonelli

4/1/2016

I pledge my honor that I have abided by the Stevens Honor System.

Ex 1.

1. (set! x 3) → x is set to 3; no output to the console

2. x → 3

3. (set! x 4) x → 4

4. (begin (set! x 5) x) → 5

5. x x → 5 5

6. (begin x x) → 5

Ex 2.

1. (set! x 3) y → 2

2. The value of x is set, but the value of y does not change, so y does not point to x.

Ex 3.

1. (set! u ‘(3)) v → ‘(1 2)

2. Value of u is set, value of v does not change, so v does not point to u.

Ex 4.

1. (counter) → 1

2. (counter) → 2

3. (+ (counter) (counter)) → 7

4. (eq? (counter) (counter)) → #f

Ex 5.

(define stack

(let (( stk '() ) )

( lambda ( message )

( case message

(( empty ?) ( lambda ()

( null? stk ) ) )

(( push! ) ( lambda ( x )

(let ((dummy (set! stk (cons x stk)))) stk)))

(( pop! ) ( lambda ()

(let ((dummy (set! stk (cdr stk)))) stk)))

(( top ) ( lambda ()

(car stk)))

( else ( error " stack: Invalid message " message))))))

Ex 6.

(define (ex1 v1 v2)

(let ((f (stack 'push!)))

(begin

(f v1)

(f v2))))

Ex 7.

1. (set-mcdr! c 5), query d → (mcons 0 (mcons 1 5))

2. (set-mcdr! c 5), query e → (mcons 0 (mcons 1 5))

Ex 8.

1. (mcar l) → ‘a

2. (mcar (mcdr (mcdr (mcdr l)))) → ‘b

3. List l contains elements 'a and 'b.

Ex 9.

1. 2

2. 5

3. Parameters passed locally rather than as a reference.

Ex 10.

1. (mcons 1 5)

2. Parameters of mutable pairs passed as reference. Parameter modified in a function is modified also in the environment using the parameter.