Jason Wang jsw50
Professor Donald
CS 230
September 3, 2017

Short Assignment 1

Expression	**Rule**
1. (mystery 7 2)	Evaluate mystery
2. [{proc ((a <number>) (b <integer>)) ...} {7} {2}]	Apply compound proc to 7 and 2
3. (cond ((zero? {2}) 0)	Evaluate cond

3. (cond ((zero? {2}) 0)
    ((odd? {2})
     (+ {7} (mystery (+ {7} {7})
                 (quotient {2} 2))))
    (else
     (mystery (+ {7} {7})
          (quotient {2} 2))))

4. (zero? {2} 0)	Evaluate (zero? {2} 0)
#f	
5. [cond ({#f})	Apply #f to the zero? cond
((odd? {2} ...))]	
6. (odd? {2} ...)	Evaluate (odd? {2} ...)
#f	
7. [cond ({#f})	Apply #f to the odd? cond
({#f})	
(else ...)]	
8. (else (mystery (+ {7} {7})	Evaluate (else...)
(quotient {2} 2)))	
9. (+ {7} {7})	Evaluate (+ {7} {7})
10. [add 7 7]	Apply add to get 14
14	
11. (quotient {2} 2)	Evaluate (quotient {2} 2)
12. [quotient {2} 2]	Apply quotient to get 1
1	
13. (mystery {14} {1})	Evaluate mystery
14. [{proc ((a <number>) (b <integer>)) ...} {14} {1}]	Apply compound proc to 14 and 1
15. (cond ((zero? {1}) 0)	Evaluate cond

15. (cond ((zero? {1}) 0)
    ((odd? {1})
     (+ {14} (mystery (+ {14} {14})
              (quotient {1} 2))))
    (else
     (mystery (+ {14} {14})
         (quotient {1} 2))))

| 16. (zero? {1} 0) | Evaluate (zero? {1} 0) |
| #f | |

17. [cond ({#f})   Apply #f to the zero? cond
        ((odd? {1} …))]
18. ((odd? {1})   Evaluate (odd? {2} …)
    (+ {14} (mystery (+ {14} {14})
                (quotient {1} 2))))
    #t
19. [{#t}   Apply #t to the odd? cond
    (+ {14} (mystery (+ {14} {14})
            (quotient {1} 2)))]

20. (+ {14} {14})   Evaluate (+ {14} {14})
21. [add 14 14]   Apply add to get 28
    28
22. (quotient {1} 2)   Evaluate (quotient {1} 2)
23. [quotient {1} 2]   Apply quotient to get 0
    0
24. (mystery {28} {0})   Evaluate mystery
25. [{proc ((a <number>) (b <integer>)) …} {28} {0}] Apply compound proc to 28 and 0
26. (cond ((zero? {0}) 0)   Evaluate cond
        ((odd? {0})
         (+ {28} (mystery (+ {28} {28})
                    (quotient {0} 2))))
        (else
          (mystery (+ {28} {28})
                    (quotient {0} 2))))
27. (zero? {0}) 0   Evaluate (zero? {0}) 0 to get #t
    #t
28. [{#t}]   Apply #t to zero? cond
    0
29. (+ {14} {0})   Evaluate (+ {14} {0})
30. [add {14} {0}]   Apply add to get 14
31. 14   ***Evaluation Complete***

The mystery function essentially multiplies the first parameter by the second.