Joey Troyer
11/16/22

Problem description:
The problem for Lab nine is that we needed to create functions in LISP. The frist function we need to develop is one that counts the number of logical operators in a list. So if there we were checking for ‘and and it was in our list once we get a result of 1. The second function we need to make is one that return only the unique variables from that list. So if we had a list (a b c a d) the function would return (a b c d). The last function we need to develop is one that reduces logical operators such at AND, OR and NOT.

UNIQUE:

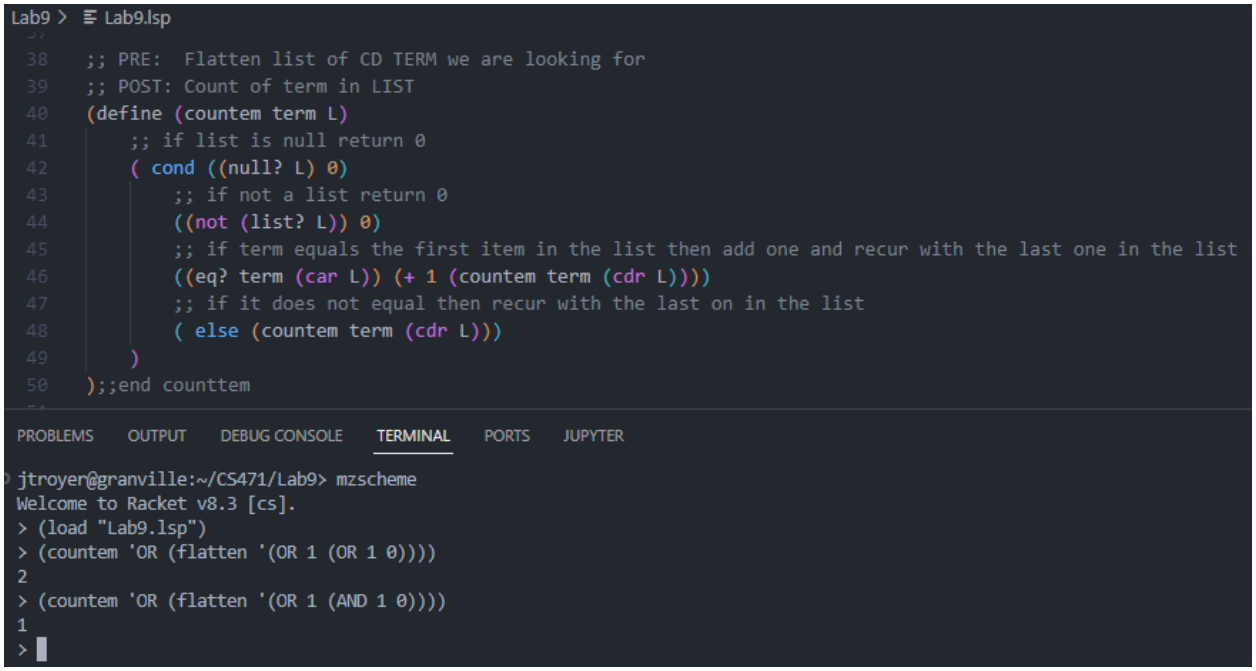
Lab9 > Lab9.lsp

```
1   ;;Name: Joey Troyer
2   ;;Date: 11/16/22
3   ;;Program: This program can do three things. One it can return the unique variables in a list. Two it can count the
4   ;;         number of logical operators in a list. Three it can evaluate the logical operators AND, OR and NOT.
5
6
7   ;; PRE: list
8   ;; POST: Displays only the unique variables in the list
9   (define (uniq L)
10      ;;if the list is empty return an empty list
11      ( cond ((null? L) '() )
12
13      ;;if not a list return an empty list
14      ((not (list? L)) '() )
15
16      ;;if the first item in the rest of the list get rid of it and recur through the list
17      (( member (car L) (cdr L)) (uniq (cdr L)))
18
19      ;;if the first item is not in the rest of the list just recur through the list
20      (else ( cons (car L) (uniq (cdr L))))
21      )
22  );; end uniq
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

```
jtroyer@granville:~/CS471/Lab9> mzscheme
Welcome to Racket v8.3 [cs].
> (load "Lab9.lsp")
> (uniq '(a b c a d))
(b c a d)
> (uniq '(a a a b b c c))
(a b c)
>
```

COUNT-OPERATOR:



Lab9 > Lab9.lsp

```
;; PRE:  Flatten list of CD TERM we are looking for
;; POST: Count of term in LIST
(define (countem term L)
    ;; if list is null return 0
    ( cond ((null? L) 0)
        ;; if not a list return 0
        ((not (list? L)) 0)
        ;; if term equals the first item in the list then add one and recur with the last one in the list
        ((eq? term (car L)) (+ 1 (countem term (cdr L))))
        ;; if it does not equal then recur with the last on in the list
        ( else (countem term (cdr L)))
    )
);;end counttem
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

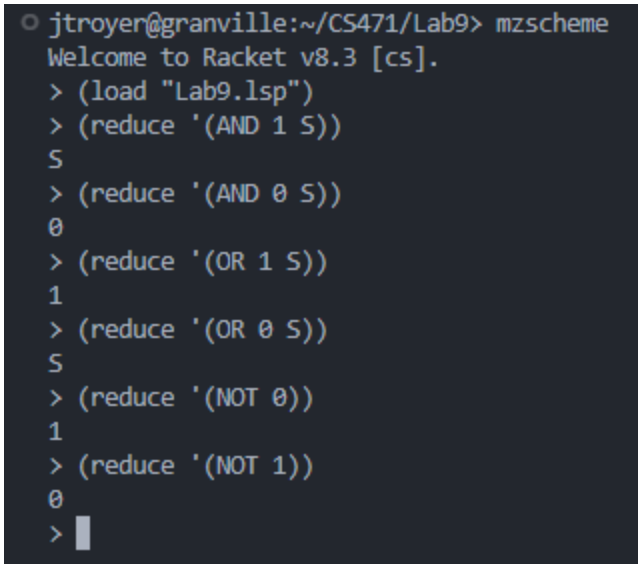
```
jtroyer@granville:~/CS471/Lab9> mzscheme
Welcome to Racket v8.3 [cs].
> (load "Lab9.lsp")
> (countem 'OR (flatten '(OR 1 (OR 1 0))))
2
> (countem 'OR (flatten '(OR 1 (AND 1 0))))
1
>
```

REDUCE:

Lab9 > Lab9.lsp

```
;; PRE:  CD
;; POST: calls appropriate function to reduce AND, OR, NOT
(define (reduce L)
    ( cond ((not (list? L)) L)
           ((eq? 'OR (car L)) (ro L))
           ((eq? 'NOT (car L)) (rn L))
           ((eq? 'AND (car L)) (ra L))
    )
);;end reduce



;; PRE:  CD
;; POST: reduces OR
(define (ro L)
    ;;reduces left hand side and right hand side
    ( let ([S1 (reduce (cadr L))]
           [S2 (reduce (caddr L))]
           )
           ;;if left hand side or right side is true displays 1
           (cond ((eq? S1 1) 1)
                 ((eq? S2 1) 1)

                 ;;if left hand side and rigth hand side are both false displays the value from the opposite side
                 ((eq? S1 0) S2)
                 ((eq? S2 0) S1)
                 (else (list 'or S1 S2))
           )
    )
);;end ro


;; PRE:  CD
;; POST: reduces NOT
(define (rn L)
    ;;reduces left hand side and right hand side
    ( let ([S1 (reduce (cadr L))]
           )
           ;;if value is 1 displays a 0
           (cond ((eq? S1 1) 0)
                 ;;if value if 0 displays a 1
                 ((eq? S1 0) 1)
                 (else (list 'NOT S1))
           )
    )
);;end rn

```

```

;; PRE:  CD
;; POST: reduces AND
(define (ra L)
    ;;reduces left hand side and right hand side
    ( let ([S1 (reduce (cadr L))]
           [S2 (reduce (caddr L))]
           )
           ;;if left hand side or right side is true displays value from opposite side
           (cond ((eq? S1 1) S2)
                 ((eq? S2 1) S1)

                 ;;if left hand side and rigth hand side are false displays 0
                 ((eq? S1 0) 0)
                 ((eq? S2 0) 0)
                 (else (list 'AND S1 S2))
           )
    )
);;end ra
```


jtroyer@granville:~/CS471/Lab9> mzscheme
Welcome to Racket v8.3 [cs].
> (load "Lab9.lsp")
> (reduce '(AND 1 S))
S
> (reduce '(AND 0 S))
0
> (reduce '(OR 1 S))
1
> (reduce '(OR 0 S))
S
> (reduce '(NOT 0))
1
> (reduce '(NOT 1))
0
>