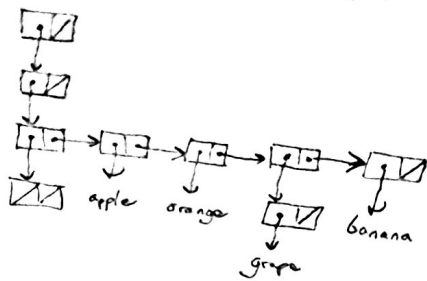
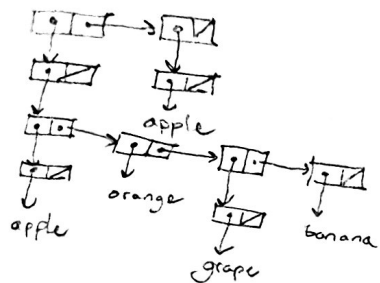


Drew Pulliam
 DTP180003
 CS 4337.002
 Assignment 2

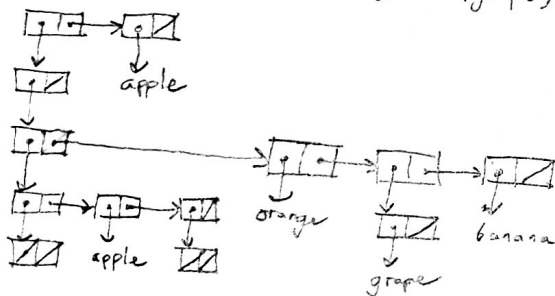
#1 (((() apple orange (grape) banana)))



((() (apple) orange (grape) banana)) (apple)



(((((() apple ())) orange (grape) banana)) apple)



- 1 See first page for drawing
-

- 2 `(caaar '((((() apple ()) orange (grape) banana)) apple))`
 `= caar '(((() apple ()) orange (grape) banana))`
 `= car '(() apple ()) orange (grape) banana)`
 `= (() apple ())`

`(caddr'((apple) orange (() (grape) banana)))`
 `= cadr'(orange (() (grape) banana))`
 `= car'(((()) (grape) banana))`
 `= (() (grape) banana)`

`(cadadr'(orange (() (grape () (apple)) banana)))`
 `= cadar' ((() (grape () (apple)) banana))`
 `= cadr'((()) (grape () (apple)) banana)`
 `= car'((grape () (apple)) banana)`
 `= (grape () (apple))`

- 3 `(car(cdr(cdr '(1 (2 3) 7)))) = 7,`
 `(car(car(cdr(car(car '(((1 (7)))))))))) = 7,`
 `(car(car(cdr(car(cdr(car(cdr(car(cdr(car(cdr(car(cdr '(1(2(3(4(5(6(7))))))))))))))))))))) = 7`
-

- 4 This scheme procedure “x” is a conditional statement. It first checks if the argument passed is a null list, if so it returns 0. Then it checks if the first element of the list argument is itself a list. If the first argument is a list it then goes into another nested conditional statement. This nested conditional checks if the first element is “#f” or false. If it is then it recursively calls itself on the remainder of the list without adding to the total “(x (cdr lis))”. If the first element is not “#f”, then it recursively calls itself and adds 1 to the total. If the first argument is a list, then the function will recursively call itself on the first argument and add it to the remainder of the list.

This function overall is used to iterate over a list, which can contain other nested lists, and it counts the amount of times something other than “#f” appears, presumably “#t”. It then returns the count at the end.