Jason Wang
CS 230
Professor Donald
September 25, 2017

Short Assignment 8

```
(define atom?
  (lambda ((o <obj>))
    (not (pair? o))))

(define nesting-depth
  (lambda ((o <obj>))
    (if (atom? o)
        0
        (max (+ 1 (nesting-depth (car o)))
             (nesting-depth (cdr o))))))
```

INDUCTION ON: An object obj

STATEMENT: (nesting-depth obj) returns the correct nesting depth

BASIS: obj = an atom
(nesting-depth ( )) = 0 by the substitution model because (if (atom? o) = #t as (not (pair? o)) = #t so it is equal to 0 which is true as an atom has a depth of 0

INDUCTION:
Inductive Hypothesis:

Assume (nesting-depth obj') returns the correct nesting depth

Must show that (nesting-depth s.obj') returns the correct nesting depth for all s

(if (atom? [s.obj'])) = #f by definition, as it contains at least s

By the substitution model then:
(nesting-depth s.obj') = (max (+ 1 (nesting-depth (car [s.obj'])))
(nesting-depth (cdr [s.obj'])))

Because they are lists:
= (max (+ 1 (nesting-depth s))
(nesting-depth obj'))

If the first comparator, (+ 1 (nesting-depth s)) is greater then by the inductive hypothesis:
= (+ 1 ([correct nesting depth of s])
Which is right as s is part of another list obj' so have to add 1 to that depth

If the second comparator, (nesting-depth obj’) is greater then by the inductive hypothesis:
= Correct nesting depth of obj’

Therefore, by the above, (nesting-depth s.obj’) will return the correct nesting depth of s.obj’ for all s.

Therefore, by structural induction, (nesting-depth obj) returns the correct nesting depth

Run Time:

In the worst case with a list of n elements and d depth, the worst case would be that each element goes down that d depth. Therefore it would be O(n*d) depending on the size of n and d.