

Scott Waldron
W00970369
11/9/16
CSCI301 Lab7

1. In the function definition for `set?`, what does the line `(member? (car lat) (cdr lat))` check for?

`(member? (car lat) (cdr lat))` checks that the next atom exists somewhere else in the `cdr` of `lat`. If it does, it returns `true`.

2. To make a list into a set, what restrictions must be applied to the atoms in the list?

The list must not contain any duplicates as that would no longer be a set. For example, `(a b c d a e)` is not a set because it contains `a` twice. The list `{a b c d e}` is indeed a set because there are no duplicates.

3. What are two different functions you could use to make a list into a set? In other words, what is the main difference between the two definitions for the `makeset` function?

One approach for making a set out of list is to use a helper method `member?`. It chops through the list and cons'ses it onto a new list but if the `(car lat)` is a member of the `(cdr lat)`, we don't add that one and move forward. The other approach is to cons a new list out of the old. But each time we get the `(car lat)`, we are going to use `multirember` and remove any other instance of that word in the rest of the list. Therefore, it only leaves us with unique words, thus a set.

4. Provide a definition for a function that checks for equality between three sets and returns `true` only if the atoms in each of the three sets are the same, regardless of the order of the atoms with the sets.

```
(define setEqual?
  (lambda (set1 set2 set3)
    (cond
      ((subset? set1 set2) (subset? set2 set1) (subset? set1 set3) (subset?
set3 set1)
        (subset? set2 set3) (subset? set3 set2))
      (else #f))))
```

This definition relies on the helper method `subset?` that will only return `true` if one set is a subset of another and vice versa.

5. Why is `(firsts rel)` a set?

`(firsts rel)` is a set because `firsts` returns the first expression in the `rel` and a `rel` is a list of pairs. What is returned is a list with just one pair in it.