**Piazza questions and answers from previous terms**

Problem 3: relation?

are Strings allowed in sets?. a better questions would be: What data types are allowed in sets? and for this problem are we supposed to only be looking for lists (that are sets) that contain lists that only contain two pairs in them?

so:

is this '((a b) (c d) "5")) a set? but it shouldn't be allowed into a relation

is this '((a b) . (b c))) a set? and what does the dot in the middle do to the list?

thanks!

**the students' answer,**

*where students collectively construct a single answer*

[Actions](https://piazza.com/class/is9cjqgxyh31b?cid=14)

In respect to a part of your second question, the dot denotes an improper list (that is, the cdr of a proper list is the empty list, '(), while the cdr of an improper list is the element itself).  
  
E.g. let list1 be '(1 2): (cdr list1) -> (2), and (cdr (cdr list1)) -> '()  
let list2 be '(1 . 2): (cdr list1) -> 2, and (cdr (cdr list2)) -> error! (try it yourself to see why).

**the instructor’s answer**

According to the definition of **set** that I gave in the assignments, any Scheme object is allowed in a set.

> '((a b) . (b c))  
((a b) b c)  
> '((a b) (c d) "5"))  
((a b) (c d) "5")

Each of these is a set of three items.  The . is basically an abbreviation for **cons**.  When we cons (a b) onto the front of (b c) we get a list of three items: (a b) , b, and c.

# A3 problem 3

Could someone elaborate on what a relation is? for instance why is this example (relation? '((a b) (c d) (a b)))  not a relation

**the students' answer,**

*where students collectively construct a single answer*

[Actions](https://piazza.com/class/is9cjqgxyh31b?cid=22)

A relation by definition is a set. The fact that (a b) is in the list twice makes it not a set and thus not a relation

# Assignment 3: Multi-set

For the multi-set problems in Assignment 4, does the "symbol" mentioned have to be a letter or can you have a multi-set that looks like: ((3 3) (q 4)), which would interpret as three 3's and four q's?

Edit: I actually just went ahead and let scheme decide by using (symbol?).

**the students' answer,**

*where students collectively construct a single answer*

[Actions](https://piazza.com/class/is9cjqgxyh31b?cid=8)

From TSPL: "Scheme supports many types of data values ... including characters, strings, symbols, lists or vectors of objects, and a full set of numeric data types."

To me, this sounds like numbers and symbols are different data types. There are also no tests with numbers (like your ((3 3) (q 4)) example) in the code in the assignment or the test code.

That said, I don't think it should change the implementation of the problem much, if at all.

EDIT: Looks like I was beaten to the answer

**~ An instructor (Claude Anderson) endorsed this answer  ~**

**the instructors' answer,**

*where instructors collectively construct a single answer*

[Actions](https://piazza.com/class/is9cjqgxyh31b?cid=8)

3 is a number, not a symbol.

'a is a symbol.

'abc123 is a symbol.

'+ is a symbol

(car '(a b c)) is a symbol

'1 is not a symbol

list? vs. pair?

Recall that a pair is simply a container for two values; the simplest way to make one is to apply cons.

A list is a linked list of pairs.  Each pair except the last one is a reference to the next pair in the list; the cdr of the last pair must be null, otherwise the list is improper.

pair? is a constant-time procedure that simply asks, "is this value a reference to a pair?"

list? is a linear-time operation that asks, "is this value a reference to the first pair of a proper list?

So efficiency is one basis to choose between the two tests.

I hope that the following transcript will help you better understand these procedures.

> (list? '())

#t

> (pair? '())

#f

> (list? '(a b c))

#t

> (pair? '(a b c))

#t

> (list? '(a b . c))

#f

> (pair? '(a b .c))

#t

> (pair? 'a)

#f

> (list? 'a)

#f

# What does reflexive mean?

A student wrote:

I'm having trouble understanding what I actually need to calculate. The description of what makes a relation reflexive is confusing. Is there any references that I could look at to understand what makes a relation reflexive?

Think of "(a, b) ε R" as another way of saying “In relation R, a is related to b”.  So a relation is reflexive iff for every element **a** in the domain or range of R, it is true that (**a**, **a**) ε R.

Other links that may help:

<https://en.wikipedia.org/wiki/Reflexive_relation>

<https://www.csee.umbc.edu/~stephens/203/PDF/10-2.pdf>

Example (from <http://www.math-only-math.com/reflexive-relation-on-set.html> ):

Consider, for example, a set A = {p, q, r, s}.

The relation R11 = {(p, p), (p, r), (q, q), (r, r), (r, s), (s, s)} in A is reflexive, since every element in A is R11-related to itself.

But the relation R22 = {(p, p), (p, r), (q, r), (q, s), (r, s)} is not reflexive in A since q, r, s ∈ A but (q, q) ∉ R22, (r, r) ∉ R22 and (s, s) ∉ R

Don't forget that you can trace your code!

If you are having trouble understanding what your code is doing, trace can help you see what your code is doing prior to the wrong answer or the error.  You can trace procedures that you write as well as built-in procedures.

 An extreme example:

> (trace - \*)

Warning in trace: redefining -; existing references will not be traced

Warning in trace: redefining \*; existing references will not be traced

(- \*)

> (define fact

(lambda (n)

(if (= n 0)

1

(\* n (fact (- n 1))))))

> (trace fact)

(fact)

> (fact 3)

|(fact 3)

| (- 3 1)

| 2

| (fact 2)

| |(- 2 1)

| |1

| |(fact 1)

| | (- 1 1)

| | 0

| | (fact 0)

| | 1

| |(\* 1 1)

| |1

| (\* 2 1)

| 2

|(\* 3 2)

|6

6

> (untrace)

(- \* fact)

>

# relation on hw3

Question:

Below are two test cases.

I can’t tell why the bottom one is false. It seems like the top one but with just one more pair. I don’t think it is illegal to 3 but not 4.

I’m confused what condition it fails on.

(relation? '((a b) (b a) (b b) (a a))) è #t

(relation? '((a b) (c d) (a b))) è #f

Answer:

A relation is a *set* of ordered pairs.  This one is not a set because (a b) occurs twice.

Assignment 3 Test Code multi-set? Question

(and (multi-set? '((a 2)(b 3)))  
Is anyone having trouble passing this test case offline and in the grading server?  
From my understanding it's not a multi-set because it isn't a list.

**the instructors' answer,**

It is a multi-set.  It is a list of two things.  Each of those things is a list of a symbol and an integer.

And the rest of the properties needed for a multi-set are also true.

# Question on #7

For #7, why should the answer to (ms-size '((a 2) (b 3)) be 5?

**the students' answer,**

*where students collectively construct a single answer*

[Actions](https://piazza.com/class/ke8ite9gsc64w2?cid=42)

There are 2 a's and 3 b's. 2+3=5