

Assignment 04 - Some LISP (DUE: Tuesday April 24, 2018)

For this assignment you are to first refactor and then extend the provided collection of LISP functions that provide an abstraction for sets.

The given file, `sets.lsp`, presents a collection of LISP functions that each perform an operation involving sets. Recall from mathematics that a set is defined as an unordered collection of elements drawn from some universe. The basic element in LISP is known a "symbolic expression" and a symbolic expression may either be a list for an atom. Generally, atoms may be symbolic (ex: `x`, `Total2018`, `PMJ`), numeric (ex: `17`, `-42.37`), or could be a string.

Refactor

Each of the functions as given to you utilize the `COND` function to express their logic. You are to refactor every function so as to use the `IF` function instead of `COND`. The operation of the functions are to remain unchanged. For the most part you should find this refactoring to be very straightforward, with the end result merely being an alternative way of expressing what is already expressed. It is probably not the case that an "all `COND`" implementation or an "all `IF`" implementation is best, but rather that both `COND` and `IF` have their place and some implementation that wisely uses `COND` or `IF` in each situation might be better.

An interesting aspect of this part of the assignment has to do with knowing that you have not "broken" any of the functions. Thus, you are to consider how you test your refactoring.

For this part of the assignment you are to submit a file named `setsR.lsp` that contains your refactored functions. In addition, you are to submit a MS Word readable document file (named `setsR.docx`) that presents an annotated illustration of your testing of the refactored functions.

Extend

The functions as given in `sets.lsp` behave appropriately when the elements in the sets are atoms, their behavior breaks down when sublists are used as elements. This anomaly is related to the fact that each set is merely a LISP list, and can be illustrated by the scenario given to the right.

Now the basic problem here is that since each set is represented simply as a list and since a list is a sequence of symbolic expressions and since lists are symbolic expressions, the symbolic expression `(STAN OLLIE)` is allowed in the list more than once. This is because each instance of the list `(STAN OLLIE)` is uniquely represented in LISP as a separate symbolic expression. On the other hand every instance to some unique atom references that unique atom. Thus, note that the attempt to include `HARPO` in the set that already contains `HARPO` results in an unchanged set; an element should not be a member of a set more than once.

```
>(load "C:\\LISP\\sets.lsp" :print t)
;; Loading file C:\\LISP\\sets.lsp ...
NIL
SEMPITYP
SMEMBERP
SINCLUDE
PREFIX
SUFFIX
SEXCLUDE
SUNION
SINTERSECTION
SDIFFERENCE
SSYMMETRICDIFFERENCE
SSUBSETP
SEQUALP
SPROPERP
;; Loaded file C:\\LISP\\sets.lsp
T
>(setq funny nil)
NIL
>funny
NIL
>(SEMPITYP funny)
T
>(setq funny (SINCLUDE 'groucho funny))
(GROUCHO)
>(setq funny (SINCLUDE 'harpo funny))
(GROUCHO HARPO)
>(setq funny (SINCLUDE '(stan ollie) funny))
(GROUCHO HARPO (STAN OLLIE))
>(setq funny (SINCLUDE 'chico funny))
(GROUCHO HARPO (STAN OLLIE) CHICO)
>funny
(GROUCHO HARPO (STAN OLLIE) CHICO)
>(setq funny (SINCLUDE 'harpo funny))
(GROUCHO HARPO (STAN OLLIE) CHICO)
>funny
(GROUCHO HARPO (STAN OLLIE) CHICO)
>(setq funny (SINCLUDE '(stan ollie) funny))
GROUCHO HARPO (STAN OLLIE) CHICO (STAN OLLIE)
>funny
(GROUCHO HARPO (STAN OLLIE) CHICO (STAN OLLIE))
```

Thus, for this part of the assignment you are to modify the collection of functions (either the given `sets.lsp` or your refactored `setsR.lsp`) into a file named `setsE.lsp` that corrects the illustrated anomaly and disallows the inclusion of elements which are sublists in a set more than once.

Additionally and optionally you are to accomplish the exclusion of sublists that may be considered to be equivalent as sets themselves. Thus, in the illustration given above, not only would `(STAN OLLIE)` be prevented from being included in the set, but so too would `(OLLIE STAN)`.

For this part of the assignment you are to submit a file named `setsE.lsp` that contains your corrected functions. In addition, you are to submit a MS Word readable document file (named `setsE.docx`) that presents an annotated illustration of your testing of the corrected functions.

Good luck,
P.m.J.