Assignment 2: Scheme Finger Exercises CS351—Fall 2008

Due 09:00 09-Oct-2008. Email text file of solutions to: `barak+cs351-hw1@cs.nuim.ie`.

1. Scan the R^{5}RS manual and find a few functions that are generalized in interesting ways. Explain why you think they were generalized in that way. (Examples are `<=` or `-`, which are generalized to accept a number of arguments other than 2.)

2. Define `list-sum-squares` which takes a list of numbers and returns the sum of their squares.

   Example: `(list-sum-squares (list 1 4 1))` ⇒ `18`

3. Define `list-product-sqrts` which takes a list of non-negative numbers and returns the product of their square roots.

   Example: `(list-product-sqrts (list 4 9))` ⇒ `6`

4. Define `set-union` which takes two lists representing sets and returns a list representing their union. (Ordering is unimportant.)

   Example: `(set-union (list 1 2 3 4) (list 6 4 8 2))` ⇒ `(1 2 3 4 6 8)` (or `(3 1 8 4 2 6)` or any other rearrangement of the elements.)

5. Define `set-intersection` which takes two lists representing sets and returns a list representing their intersection.

   Example: `(set-intersection (list 3 1 2 4) (list 4 2 8 6))` ⇒ `(2 4)` (or `(4 2)`)

6. Define `set-disjoint?` which takes two lists representing sets and returns true iff the sets are disjoint.

7. Define `filter-numbers` which takes a list representing a set and returns a list representing a set containing only those members that are numbers, i.e., that pass the `number?` predicate.

   Example: `(filter-numbers '(1 one 2 two foo zero 22/7 0))` ⇒ `(1 2 22/7 0)` (or a permutation thereof.)

8. Define `set-equal?` which takes two lists representing sets and returns true iff they represent the same set.

   Example: `(set-equal? '(1 2 3) '(2 1 3))` ⇒ `#t`

   Example: `(set-equal? '(1 2 () 3) '(2 1 3))` ⇒ `#f`

9. Define `deep-member?` which takes a symbol and an s-expression and returns true iff the symbol occurs in the given s-expression, perhaps very deeply nested.

   Example: `(deep-member 'foo '(a b (c (d e foo g)) h))` ⇒ `#t`

   Example: `(deep-member 'foo '(a b (c (d e bar g)) h))` ⇒ `#f`

10. *Optional:* If you encountered any problems with the assignment, or have any comments on it, or other comments or suggestions, I would appreciate hearing them. As practice for working in industry, where weekly reports are not unusual, please embody these in a brief (1–3 page) typed report.

Hint: use recursion and make you base cases as simple as possible.

Honor Code: You may discuss these with others, but please write your answers by yourself and without reference to communal notes. In other words, your answers should be *from your own head.*