

Lab 1

The primary programming activity in Scheme is the creation of recursive definitions, where problems are solved in terms of itself. After working through the increasingly difficult problems in chapter 1 of *The Little Schemer*, answer the following questions. Be sure to attempt the example schemes readily available throughout the text, frames notes preceded by “S:” concern Scheme.

Scheme expressions, or S-Expressions for Symbolic Expressions, are composed of lists or atoms. These expressions are used to construct recursive processes and manipulate recursive data-structures. Processes in Scheme use lists to represent function calls or collections of data. Primitive data types, or atoms include the following: boolean constants (t and f), numbers (integers, reals, and rationals), strings (e.g. “hello world!”), characters (e.g. #\c), and symbols (e.g. x, b11, sam).

- 1) Is *(null? (cdr '(atom)))* true or false?
- 2) Is *(eq? (car (cons red '(blue red))) car (cdr '(blue red)))* true or false?
- 3) What does *(car (cons l a))* evaluate to? Where *l* is (**apple**) and *a* is (**peach orange apple**).
- 4) What types of data may be compared using the *eq?* primitive?
- 5) Provide a list of the atoms **2**, **4**, **6** using three different schemes.