

## CSE 3302 Lab Assignment 4

Due May 1, 2014

### Goal:

Understanding of Scheme and elementary functional programming concepts.

### Requirements:

1. Extend the solution to lab 2 from Spring 2013 for additional boolean operators. The assignment appears on the next page.
  - a. Implication of the form  $(I\ p\text{-exp}\ q\text{-exp})$ , interpreted as  $p\text{-exp} \supset q\text{-exp}$  i.e.  $\overline{p\text{-exp}} \vee q\text{-exp}$ .
  - b. Equivalence (iff) of the form  $(E\ p\text{-exp}\ q\text{-exp})$ , interpreted as  $p\text{-exp} \leftrightarrow q\text{-exp}$ .
  - c. Conditional of the form  $(C\ p\text{-exp}\ q\text{-exp}\ r\text{-exp})$ , interpreted as  $(p\text{-exp} \supset q\text{-exp}) \wedge (\neg p\text{-exp} \supset r\text{-exp})$ .
2. Email your program to `sourabh.bose@mavs.uta.edu` by 9:15 a.m. on May 1, 2014.

### Getting Started:

1. The solution from last Spring is available at:  
`http://ranger.uta.edu/~weems/NOTES3302/LAB2SPR13/lab2spr13.symbol.rkt`

Note that this code uses `symbol` atoms unlike the original assignment. Your solution should also use these.

2. For each expression, the provided code does three things:
  - a. Checks the syntax to assure that further processing will be successful.
  - b. Collects the set of propositions that appear.
  - c. Generates all possible assignment of truth values to the set of propositions and determines the value of the expression for each one. A truth assignment is represented by an ordered list of symbols corresponding to the symbols with the value `#t`.

## CSE 3302 Lab Assignment 2

Due February 28, 2013

### Goal:

Understanding of Scheme and elementary functional programming concepts.

### Requirements:

1. Write a Scheme program to evaluate a boolean expression under all possible truth assignments:
  - a. The boolean expression will be a nested S-expression consisting of the following sub-expressions:
    1. Single letters (atoms) for propositions.
    2. Negation of the form (N s-exp), i.e. exactly one sub-expression.
    3. Disjunction of the form (O s-exp . . . s-exp), i.e. at least one sub-expression.
    4. Conjunction of the form (A s-exp . . . s-exp), i.e. at least one sub-expression.
  - b. The S-expression should be processed to produce a list (e.g. set) of all propositions.
  - c. Code to generate all truth assignments and evaluate the S-expression for each one. Output both the assignment and the result (`#t` or `#f`).
2. Email your program to `xxxx` by 10:45 a.m. on February 28, 2013.

### Getting Started:

1. Don't be concerned about efficiency.
2. Collecting the propositions, generating the truth assignments, and evaluating the S-expression should be tested separately.
3. The truth assignments are not required to be evaluated in any particular order and may be represented any way you wish.
4. The Ten Commandments and The Five Rules from *The Little Schemer* will lead you to many days of happiness.
5. `set!` will lead to nights of suffering (and loss of points).