

Homework 4: If Only...

Due Thursday, March 5 at 9:55 am

Tip: This is not due until after the exam...but completing it beforehand may be an effective way of studying the last two lectures.

Let the SuperCalc language be defined by the following grammar:

Expression Domain:

```
<EXP> ::= <LITERAL> | <CONDITIONAL> | <BINOP> | <NOT>

<LITERAL> ::= <NUM> | <BOOLEAN>
  <BOOLEAN> ::= #t | #f
  <NUM> ::= [+ | - ] DIGIT+
  <DIGIT> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<CONDITIONAL> ::= <IF> | <AND> | <OR>
  <IF> ::= ( if <EXP> <EXP> <EXP> )
  <AND> ::= ( and <EXP> <EXP> )
  <OR> ::= ( or <EXP> <EXP> )

<BINOP> ::= <ADD> | <SUB> | <EQ>
  <ADD> ::= ( + <EXP> <EXP> )
  <SUB> ::= ( - <EXP> <EXP> )
  <EQ> ::= ( = <EXP> <EXP> )
  <NOT> ::= ( not <EXP> )
```

Value Domain:

```
<VALUE> ::= <BOOLEAN> | <NUM>
```

Semantics:

- The value of a LITERAL expression is itself.
- The value of an ADD (or SUB) expression is the sum (or difference) of the values of its subexpressions, which must evaluate to NUMs.
- The value of a NOT expression is #f if the subexpression's value is not #f, otherwise it is #t.
- The value of an EQ expression is #t if both expressions have the same value and #f otherwise.
- The value of an IF expression is the value of the second subexpression if the first subexpression's value is not #f, otherwise it is the value of the third subexpression.
- The value of an AND expression is the value of the second subexpression if the first subexpression's value is not #f, otherwise it is #f.
- The value of an OR expression is the value of the second subexpression if the first subexpression's value is #f, otherwise it is #t.

Note that these semantics largely follow those of the equivalent forms and procedures in Scheme.

50 points. Complete the partial implementation of this language available from <http://cs.williams.edu/~morgan/cs334/handouts/hw04-starter.ss>. Your solution must implement AND, OR, and NOT by translating them into IF expressions. Either submit only the portions of the code that you changed or highlight what you changed in your solution.

Challenge 1 (no points): Implement WITH as well.

Challenge 2 (no points): Rewrite the interpreter using define-type and type-case. Where does the AND etc. go now?