

EECS 345 Homework #4 - due 10/18/05

1. Modify the Tiny-0 interpreter so that it evaluates expressions in infix form: $(1 + (2 * 3)) \Rightarrow 7$. You may assume that every subexpression is parenthesized and that the four arithmetic functions are only called in binary form (e.g., $1 - -x$ would be written as $(1 - (0 - x))$). Note that you do not need to replace the Common Lisp `read` function in order to accomplish this.
2. Convert the Tiny-1a interpreter into a simple symbolic algebra system by arranging for undefined symbols to evaluate to themselves. For example, if the symbol `$PI` has the value `3.14159` and the symbol `r` has not been given a value, the expression $(+ (* 2 \$PI) (* r r))$ should evaluate to $(+ 6.28318 (* r r))$ rather than signaling an error. Note that you do not need to simplify the resulting symbolic expressions. For example, $(+ x 1 (+ x 2))$ can evaluate to $(+ x 1 (+ x 2))$ rather than $(+ x x 3)$. However, you may add simplification if you wish.
3. Add `let*` to Tiny-1b.

General Guidelines for Assignments Involving Tiny Interpreter Modifications

You may define additional functions as necessary. You should turn in a complete listing of the corresponding Tiny interpreter, **with the code and comments that you modify or write in ALL CAPS** (note that Common Lisp is case-insensitive by default, so `SYMBOL` is the same as `symbol`). You should also turn in a transcript of sample runs of each modified interpreter that demonstrates its new behavior. Make sure that the work you turn in is organized in the same way as the assignment, so that the TA can easily find your answers to each part. A printed copy of your assignment must be turned in during class on the day the assignment is due in order to receive full credit.