Problem Set 1

Due: 22 September 1995.

Reading: Handout 5, "A Term Rewriting Model for Scheme, I."

Problem 1. Do Problem 1 from the Term Rewriting handout.

Problem 2. Do Problem 2 from the handout.

Problem 3. For each numbered step in the sample rewriting of `(factorial 4)` in the Appendix of the handout, indicate the subexpression which gets rewritten and the rules applied ("P"–primitive application, "C"–conditional, "I"–instantiation, "L"–lambda application, "G"–garbage collection)

Problem 4. Load the submodel implementation in the 6.044 ftp directory and evaluate

```
(smeval
 '((define (factorial_0 n)
     (if (= n 0)
         1
         (* (factorial_0 (dec n)) n)))
   (factorial_0 4)))
```

Turn in the final 10 steps of output. What can you conclude about the order of evaluation of the operands in a combination in this implementation of the term rewriting model?

Problem 5. Write a "self-reproducing" Scheme expression, M: evaluation of M in the initial Scheme environment (*without* any prior user-created definitions) returns a value which prints as M. (*Hint*: Letting M be the combination

```
(make-quoted-self-application 'make-quoted-self-application)
```

would work if the definition of `make-quoted-self-application` given in class was part of the initial environment.)