CS109A Problem Set #1
Due 4:30 PM, 1/20/95

Note: there is a fifth problem on the second page.

1) (30 points) The following expressions are erroneous ML. Type each into SML/NJ and tell (just to make sure you really used SML) how many lines of error message you get. Then, explain in your own words what is wrong with each expression and give a reasonable correction thereof.

   a) `not~4<5`

   b) `3div2`

   c) `3/2`

   d) `3<4 and 5<6`

   e) `3<4 or 5<6`

   f) `if 2<3 then 4`

2) (15 points) Numbers of the form $t_n = n(n+1)/2$ are called *triangular numbers,* because marbles arranged in an equilateral triangle, n on a side, will total $\sum_{i=1}^{n} i$ marbles, which we saw in Exercise 2.3.1(a) is t_n marbles. For example, bowling pins are arranged in a triangle 4 on a side and there are $t_4 = 4 \times 5/2 = 10$ pins. Show by induction on n that $\sum_{j=1}^{n} t_j = n(n+1)(n+2)/6$. (Exercise 2.3.2, p. 41, FCS.)

   Aside: Numbers of the form $n(n+1)(n+2)/6$ are sometimes called *pyramidal numbers.* They count the number of cannonballs, e.g., arranged in a pyramid. For example, a pyramid with a side of 3 has $3 \times 4 \times 5/6 = 10$ cannonballs.

3) (20 points) The formula for the sum of n terms of a geometric series

   $$a + ar + ar^2 + \cdots + ar^{n-1}$$

   is

   $$\sum_{i=0}^{n-1} ar^i = \frac{(ar^n - a)}{(r-1)}$$

   Prove this formula by induction on n. Note that you must assume $r \neq 1$ for the formula to hold. Where do you use that assumption in your proof? (Exercise 2.3.9, p. 42, FCS.)

4) (20 points) Consider an expression E whose operators are all binary. The *length* of E is the number of symbols in E, counting an operator or a left or right parenthesis as one symbol, and also counting any operand such as 123 or `abc` as one symbol. Prove that E must have an odd length. *Hint*: Prove the claim by complete induction on the length of the expression E. (Exercise 2.4.6, p. 51, FCS.)

5) (15 points) Show that for any $n \geq 1$ there is an error-detecting set of strings of length n, using only the digits 0, 1, and 2, that has 3^{n-1} strings. (Exercise 2.3.5, p. 42, FCS.) *Hint*: You need to read Example 2.5 and learn about error-detecting codes; this material will *not* be covered in class. Then, generalize the "parity-check" idea of that example. The proof does *not* involve induction. Rather, simply show how to construct the code by describing its member strings, just as was done in Example 2.5.