Logic and Verification – due Feb. 11, 2004

1. Implement an ocaml function to print all propositional formulas of a given size, where the size measures the number of nodes in the abstract syntax tree (i.e. the number of symbols not counting parentheses).

   If two formulas can be made identical by renaming variables, only one of the variants should be included.

   Equivalently, only those formulas should be considered which have the following property: given a total order on variables, if the abstract syntax tree is traversed in order, then a variable \( v \) can occur only if each variable less than \( v \) has already occurred at least once.

2. Using your above function, determine the number of formulas of each size for as high as you can go before it becomes computationally prohibitive (i.e. takes longer than a few minutes).

3. Plot the proportion of formulas that are tautologies against the size. Can you generate results for large enough lengths to see a trend? Is the trend as expected?