1. Find the definition of co-NP
   1. We just show that you can take the set of equations and use that as our certificate polyspace to the input length to verify and it is really clear how to evaluate in poly-time
2. Would it suffice to show RGB-SAT is in NP and use the fact that NP is a subset of P-SPACE or do I have to c one up with an algorithm
   1. Reduction would be harder than to just create the following algorithm:
      1. Create a gadget that fixes each setting of the red inputs:
         1. Create a gadget that fixes each setting of the green inputs:
            1. Check if circuit-SAT satisfied for blue inputs
      2. This takes (2^n)\*(2^n) calls to circuit-SAT
      3. Each call to circuit SAT takes at most poly-space
      4. Binary counter to track where in the input string we are (incrementing the red input binary string vs. green input portion), takes O(n) space
      5. Therefore, this must be poly space
3. Following algorithm — is this read once or read-only?
   1. Use stack with 2 elements — last read operator, prefix literal (with default as null)
   2. Starting from leftmost element of the formula and reading rightwards:
      1. 3 CASES:
         1. Start of new formula – pop an element to the stack with NULL literal and NULL operator
         2. Operator – update the operator in the stack’s tuple to be this operator (on a well formed input, we cannot have an operator occur at the same depth right after another, so this must have occurred after we have both a literal and a NULL for operator)
         3. Literal – If our literal stored is not NULL, evaluate the operator, updating the literal with the stored operator and change the operator to NULL. If there does not exist a literal, write the literal read at this step as a literal
         4. END of formula, pop this stack element, treating the result as a read literal to the previous step
      2. Push the literal on the stack if there is no literal
      3. If its an operator
      4. If max\_paren\_depth
         1. EVAL until end parenthesis reached  — starting at the beginning of sub-formula at depth d, by taking the operators and using them to join all the terms into a single term, popping off the most interior parenthesis when complete
   3. Return result

4:

Look at first non-deterministic lecture which showed NP-Verifier is the same as non-deterministic – lecture 9