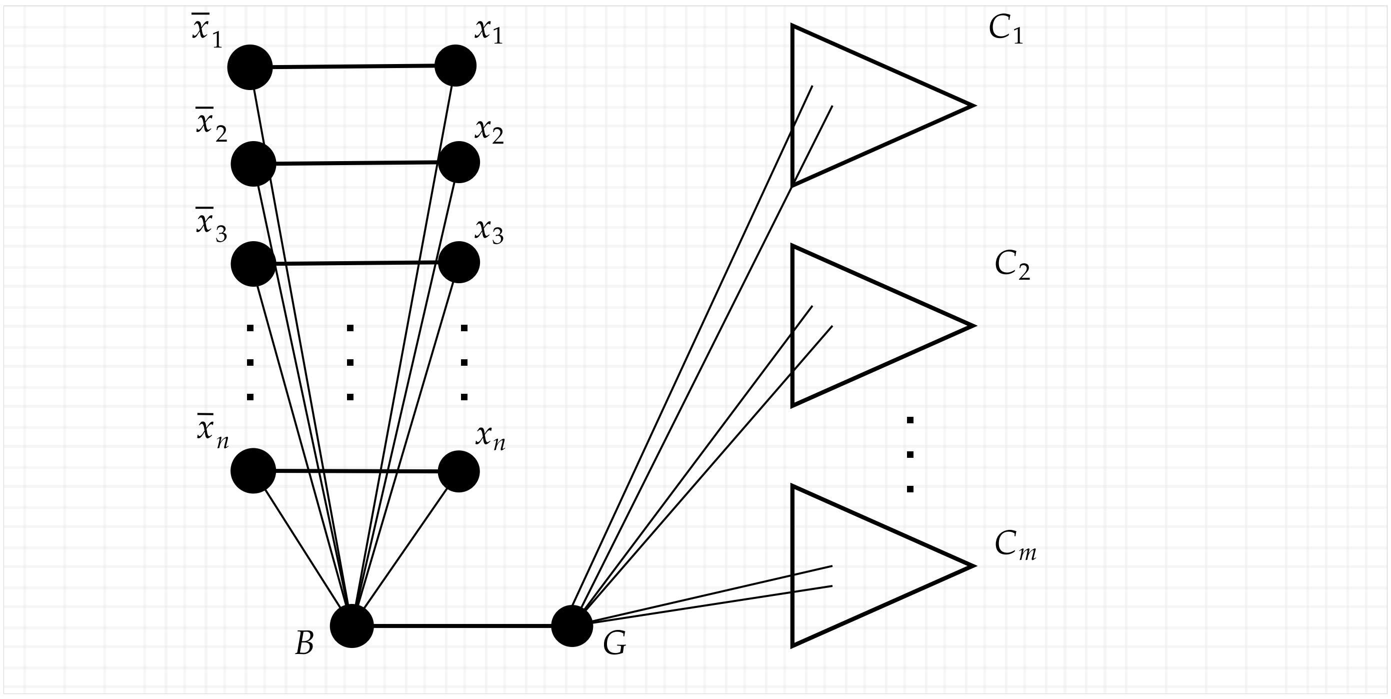
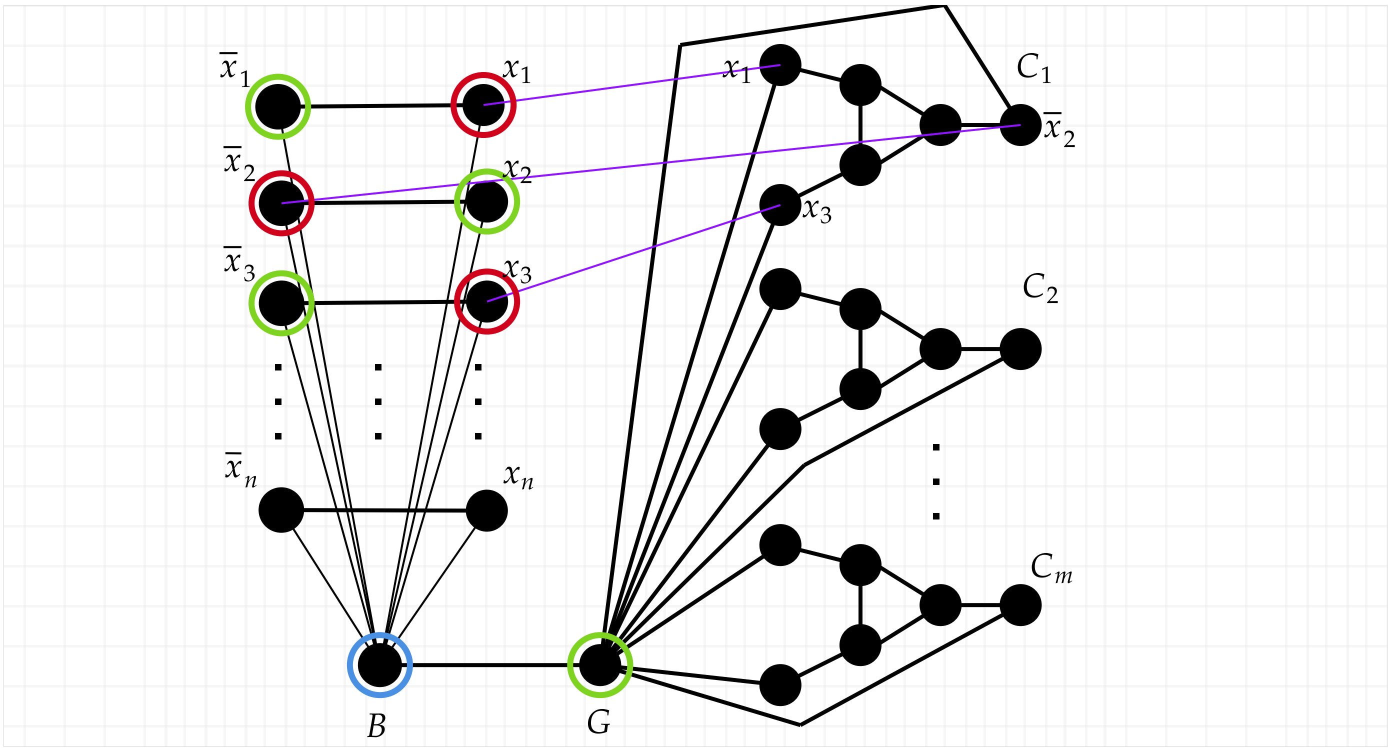
CSE491/596 Lecture Wed. 11/4: NP Completeness By Component Design III

The reduction to Graph Three-Coloring shown Monday did not show **3SAT** $\leq_m^p$ **G3C** directly, but rather **NAE-3SAT** $\leq_m^p$ **G3C.** Of course, by just happening to choose **3SAT** as the "language $A \in \mathrm{NP}$" in the Cook-Levin proof, we get **3SAT** $\leq_m^p$ **NAE-3SAT,** so **3SAT** $\leq_m^p$ **G3C** follows by transitivity. But it is useful to illustrate **3SAT** $\leq_m^p$ **G3C** directly.

The first thing we need is to add to the B node a second node G so that the colors used for those nodes wlog. count as "blue" and "green". Connections from the G node to the clause gadgets can fix the problem of symmetry betweed "red" and "green", which we need to do for reduction from **3SAT** though not from **NAE-3SAT**.

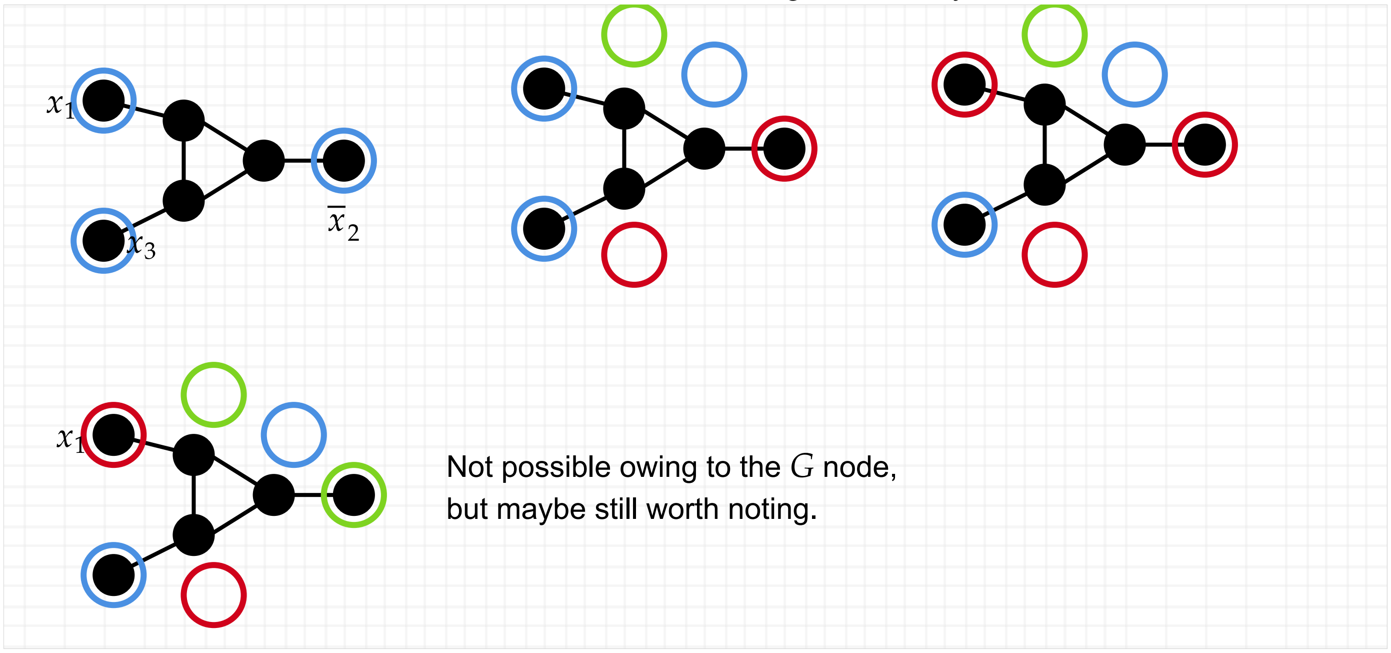


The second change is to include an outer layer of 3 nodes in the clause gadgets. The nodes will get an automatic "greenlock" from the G node. If they get a "redlock" from the rungs---which we want to mean all three literals being made false---then the 3 nodes are forced to be blue. This is without connecting the outer 3 nodes to each other. The resulting "bluelock", however, will prevent an inner triangle of each clause from being 3-colored. If, however, all three literals in the clause are made true, then the outer layer will see "greenlock" twice, and that is no problem. Here is the idea abstractly, showing only crossing edges between the rungs and the first clause $(x_1 \vee \bar{x}_2 \vee x_3)$:



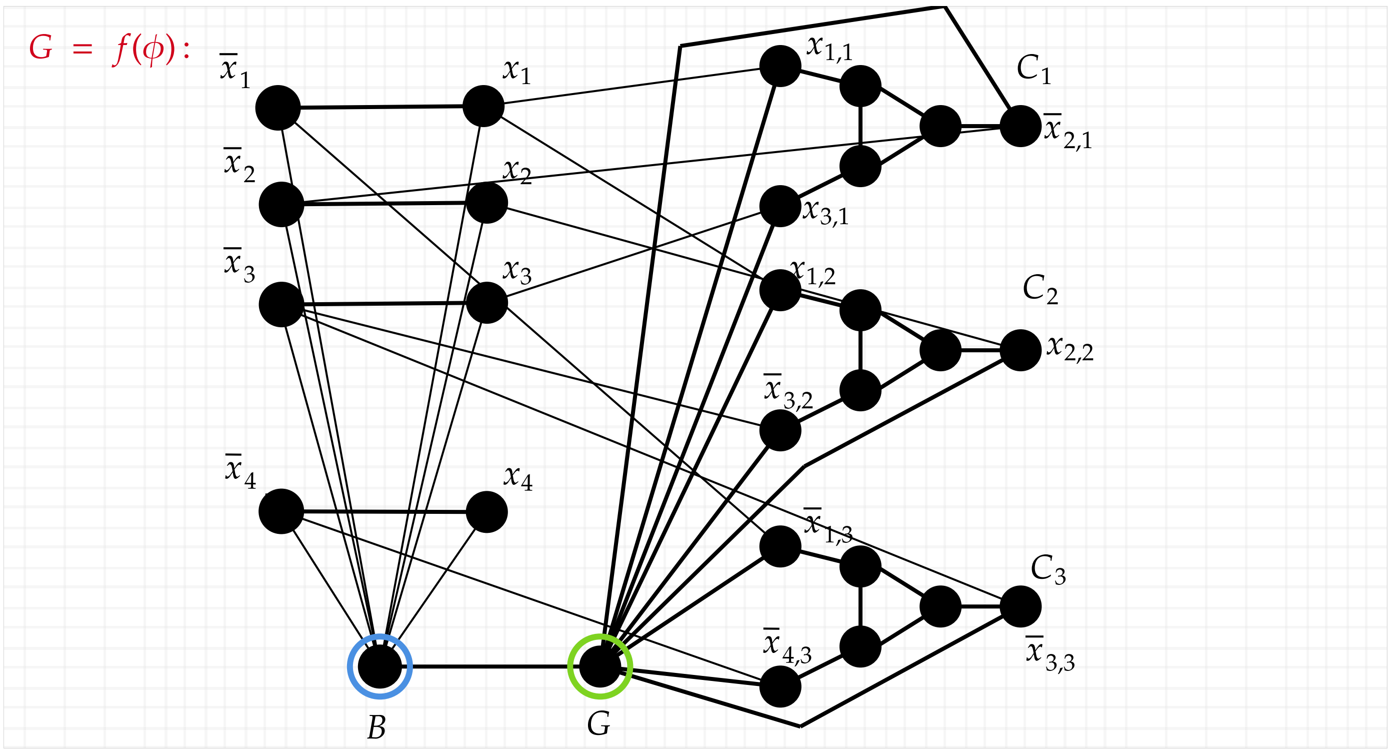
(Note: These are the opposite connections from the ALR notes, where I made the opposite choice of connecting x_1 in a clause to $\overline{x}_1$ in the rung to stay consistent with the reduction to **IND. SET**.)

If x_1 and x_3 are made false and x_2 true, so that clause C_1 fails, then each of the outer nodes of the C_1 gadget "sees red" as well as green from node G. This forces each outer node to be blue, but then the inner triangle of the C_1 gadget cannot be 3-colored. Any other assignment, however, allows using two different colors for the outer nodes, and then the inner triangle can always be 3-colored:


x_1
$\overline{x}_2$
x_3
x_1
Not possible owing to the G node, but maybe still worth noting.

Here is the whole reduction carried out for our example formula

$\phi = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_4)$

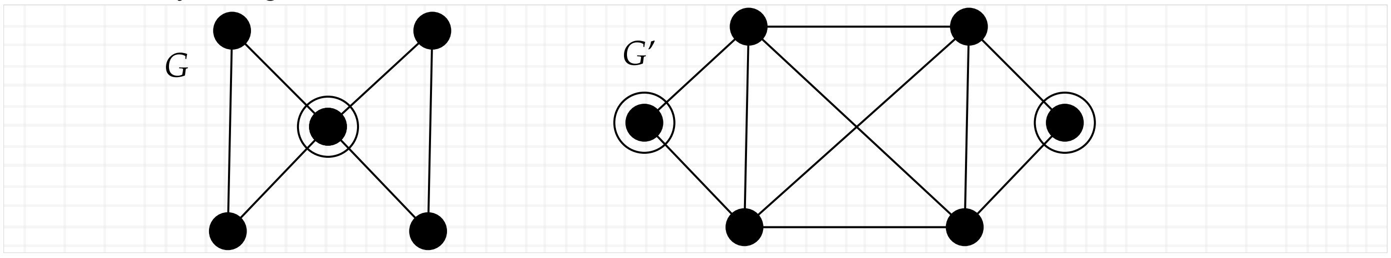


Again the reduction is linear-time computable in one sweep through ϕ. Correctness still needs to be re-checked in the other direction: If G has a good 3-coloring, then for every clause C_j, at least one of its 3 outer nodes x_{ij} or $\bar{x}_{ij}$ must be colored R. Since we are now sending crossing edges to the rung node with the *same* sign, this means the same-sign rung node must be colored G. In turn, this means the literal satisfies C_j. Thus any good coloring uniquely yields an assignment that satisfies all clauses. ☒

Now we will consider a different reduction where both the "rungs" and the "clause gadgets" get different treatment. The target problem is:

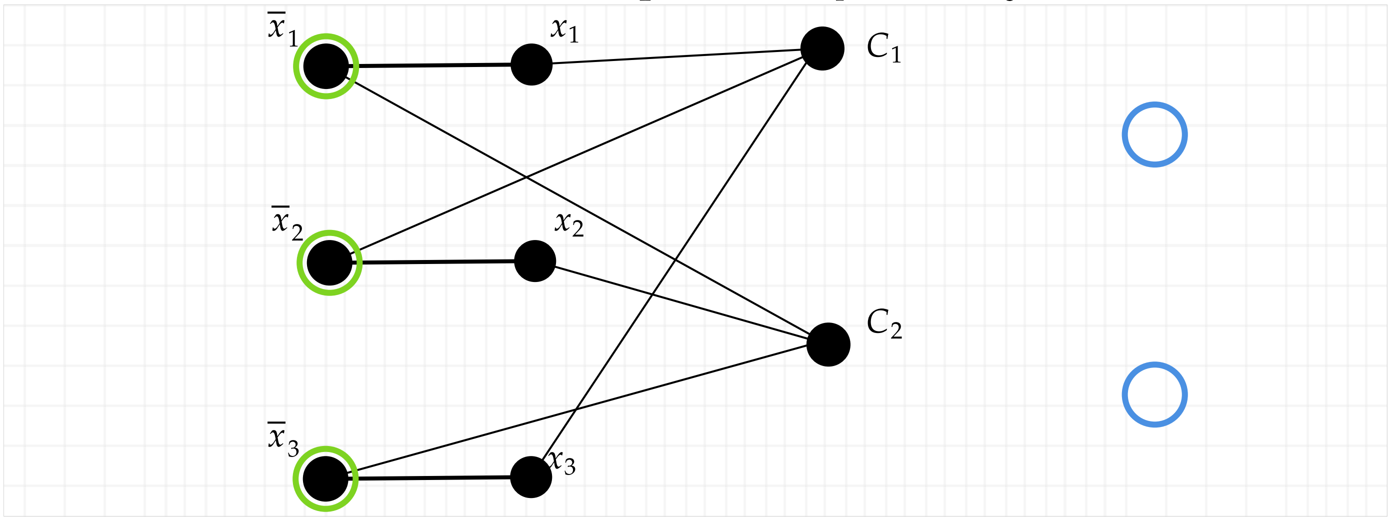
Dominating Set (**Dom Set**)
Instance: An undirected graph $G = (V, E)$ and an integer $k \geq 1$.
Question: Is there $S \subseteq V$, $|S| \leq k$, such that every node **not** in S is adjacent to a node in S?

The difference between a dominating set and a vertex cover is that the nodes don't have to cover every edge. The bowtie graph has a dominating set of size just 1, while its line graph needs $k = 2$ but can do so even by taking the two non-central nodes:



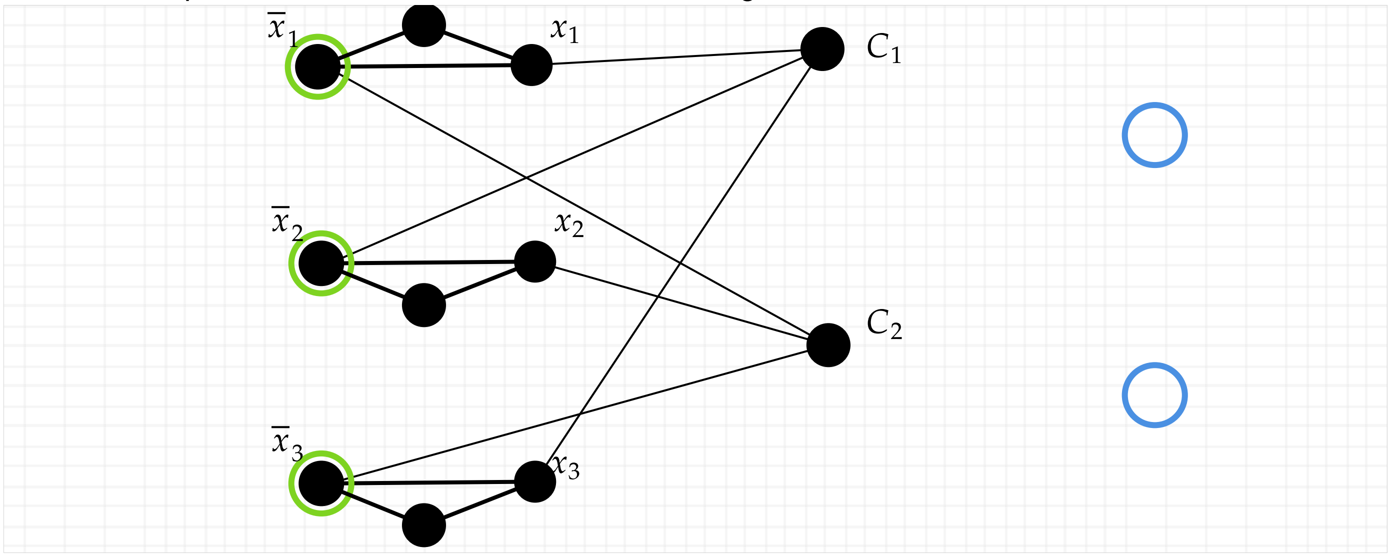
(Does the line-graph function give a reduction from **Edge Cover** to **Dom Set** or vice-versa? Hmmm... But we still want to reduce **3SAT** to **Dom Set** directly.)

The first key idea is the same: the rung nodes chosen in S correspond to those literals set true. The second key idea is simple: *make that true literal dominate every clause it satisfies.* This needs only one node per clause, and suggests taking $k = n$, irrespective of the number m of clauses. Here is how that looks for a simpler formula, $\psi = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3)$:



Setting all three variables false dominates the two clause nodes. So would setting them all true, whereas moving just x_2 to be true would fail to dominate (or satisfy) C_1. Is this all we need?

The flaw is that we have not enforced that in each rung, either x_i or $\bar{x}_i$ **must** be in S. This case allows a "surprise" domination by two nodes outside the rungs: use C_1 and C_2. To enforce the correspondence between possibly-good choices of S and truth assignments, and make sure $k = n$ is the minimum possible, we use a third node in each "rung":



Those extra nodes can only be dominated from the rung, and they do not help dominate each other, so n separate nodes are needed to dominate them. This fixes the problem. Defining the reduction formally in general and proving it correct is a self-study exercise.