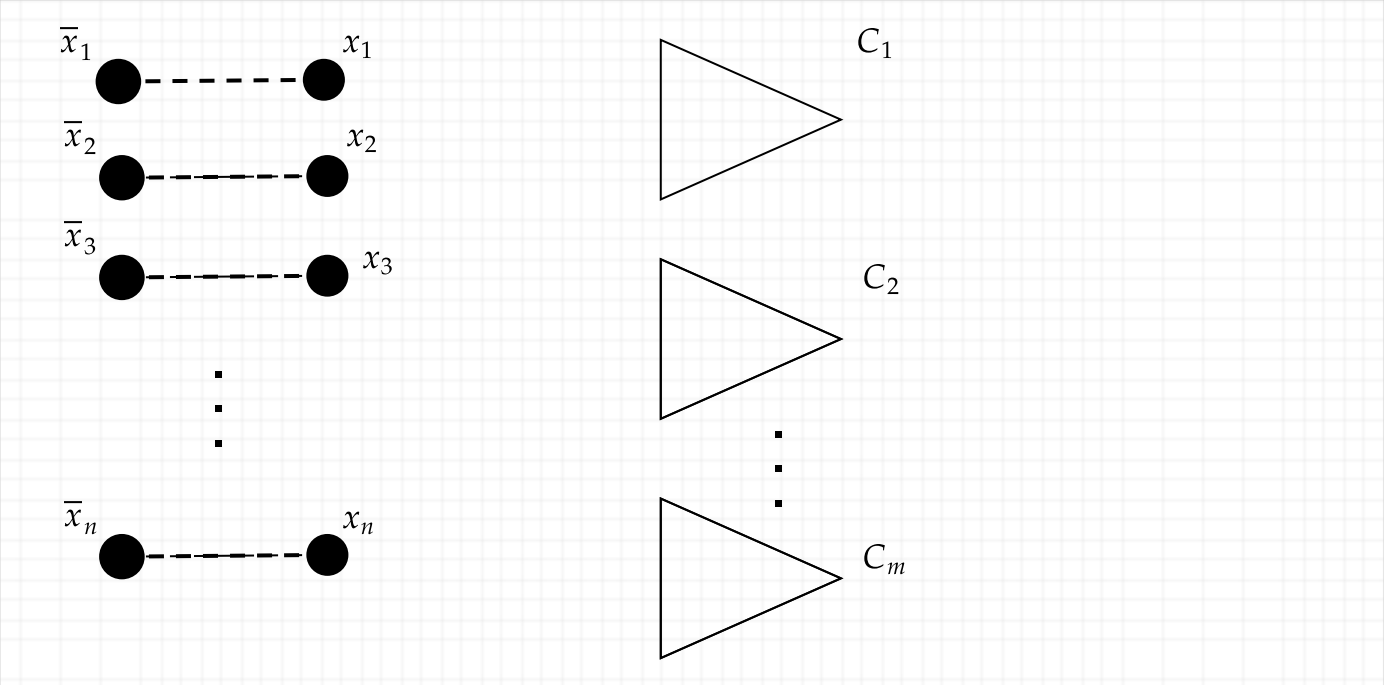
CSE491/596 Lecture Mon. 11/2/20: Reductions From 3SAT By Component Design II

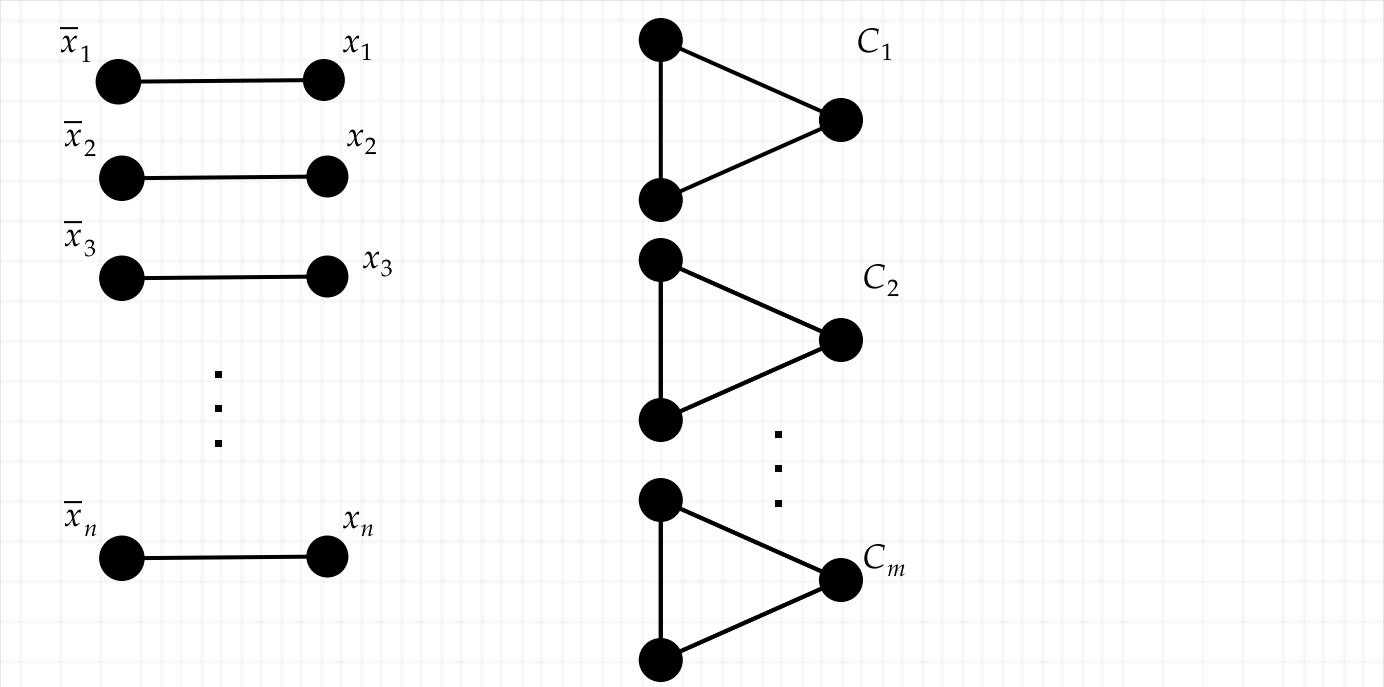
[Repeating the beginning from Friday 10/30, but restoring the first diagram the way it was before lecture so that it shows the common beginning point.]

The "Ladder and Gadgets" framework for reductionf from 3SAT: Given a 3CNF formula $\phi(x_1, \ldots, x_n) = C_1 \wedge \cdots \wedge C_m$, lay out n "rungs" of 2 nodes each and m "clause gadgets", plus (optionally) space for one or more "governing nodes":



Usually the rung nodes are connected, but not always---and sometimes an extra node or two are added to each rung.To show 3SAT $\leq_m^p$ IND SET, we need to map $f(\phi) = (G, k)$ such that G has an independent set S of size (at least) k if and only if ϕ is satisfiable.

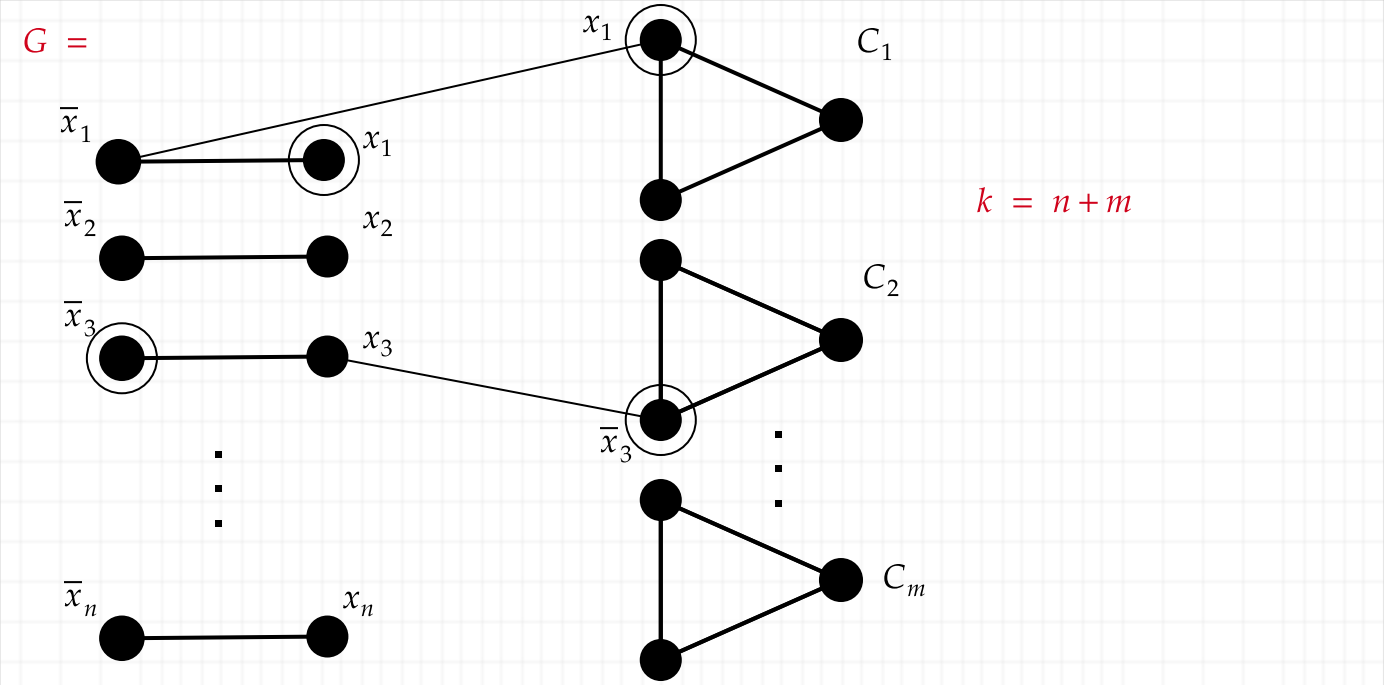
For this reduction, we make the "rungs" into actual edges between each x_i and its negation $\overline{x}_i$ and give each clause three nodes to make a triangle. Each clause node is labeled by a literal in the clause. Later we will include the clause index j, not just the variable index i, when identifying this *occurrence* of the literal in a clause to define V as a set, where $G = (V, E)$.



The immediate effect, even before we consider an example of a formula, is that the maximum possible k for an independent set S in the graph G is $n+m$. The most one can do is take one vertex from each rung and one from each triangle to make S. Note that the vertices chosen from each rung specify a truth assignment to the variables.

The final goal of the reduction is to add a third set of edges, which I call "crossing edges", to enforce that a set S of size $n+m$ is possible if and only if its corresponding assignment satisfies the formula. The basic idea, even before we consider a formula, is as follows.

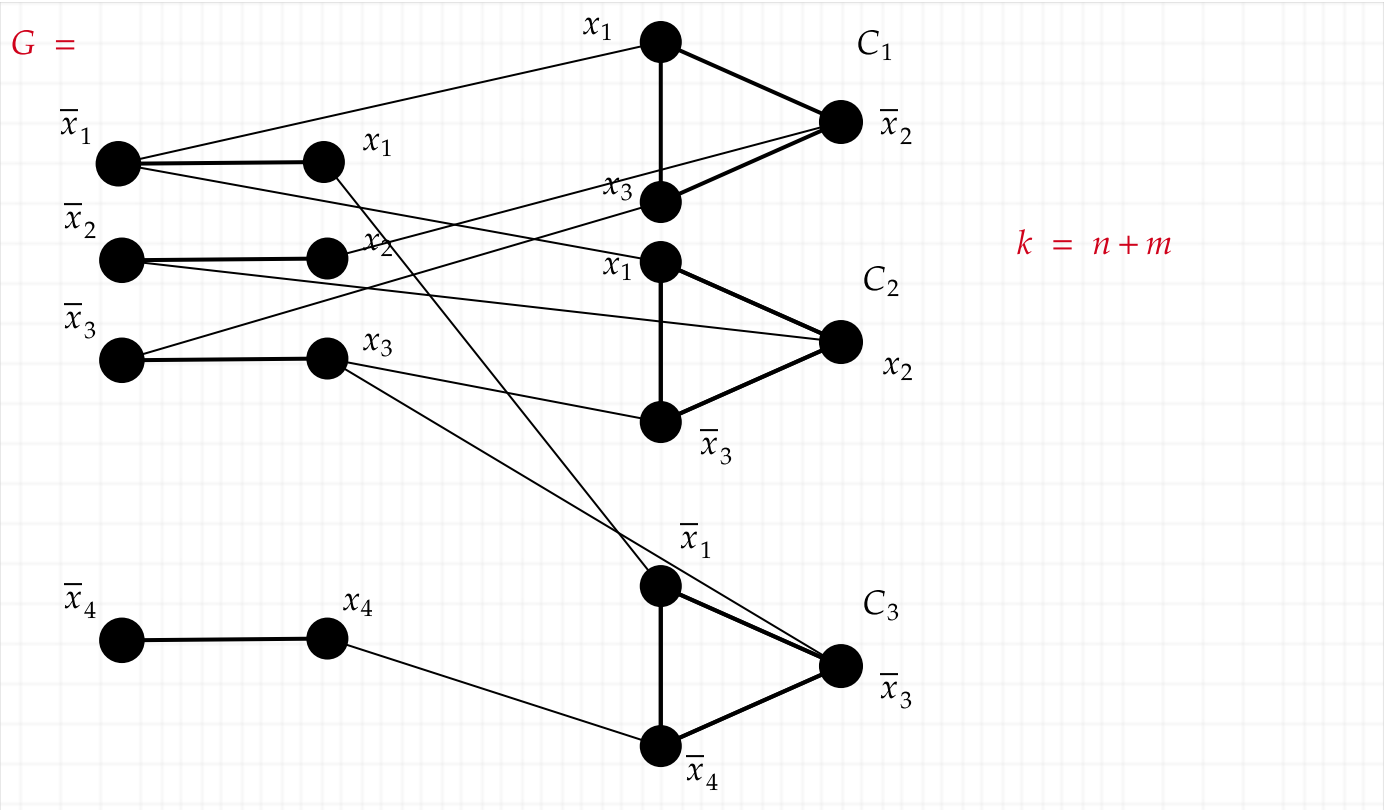
- Suppose clause C_1 includes the positive literal x_1. Then we connect a crossing edge from x_1 in C_1 to the *opposite* literal $\overline{x}_1$ in the rung.
- Suppose clause C_2 includes the negated literal $\overline{x}_3$. Then we connect a crossing edge from $\overline{x}_3$ in C_2 to the opposite literal in the rung, which is just x_3.



The edges ensure that choosing a satisfied literal in each clause will not conflict with the truth assignment. Here is an example formula.

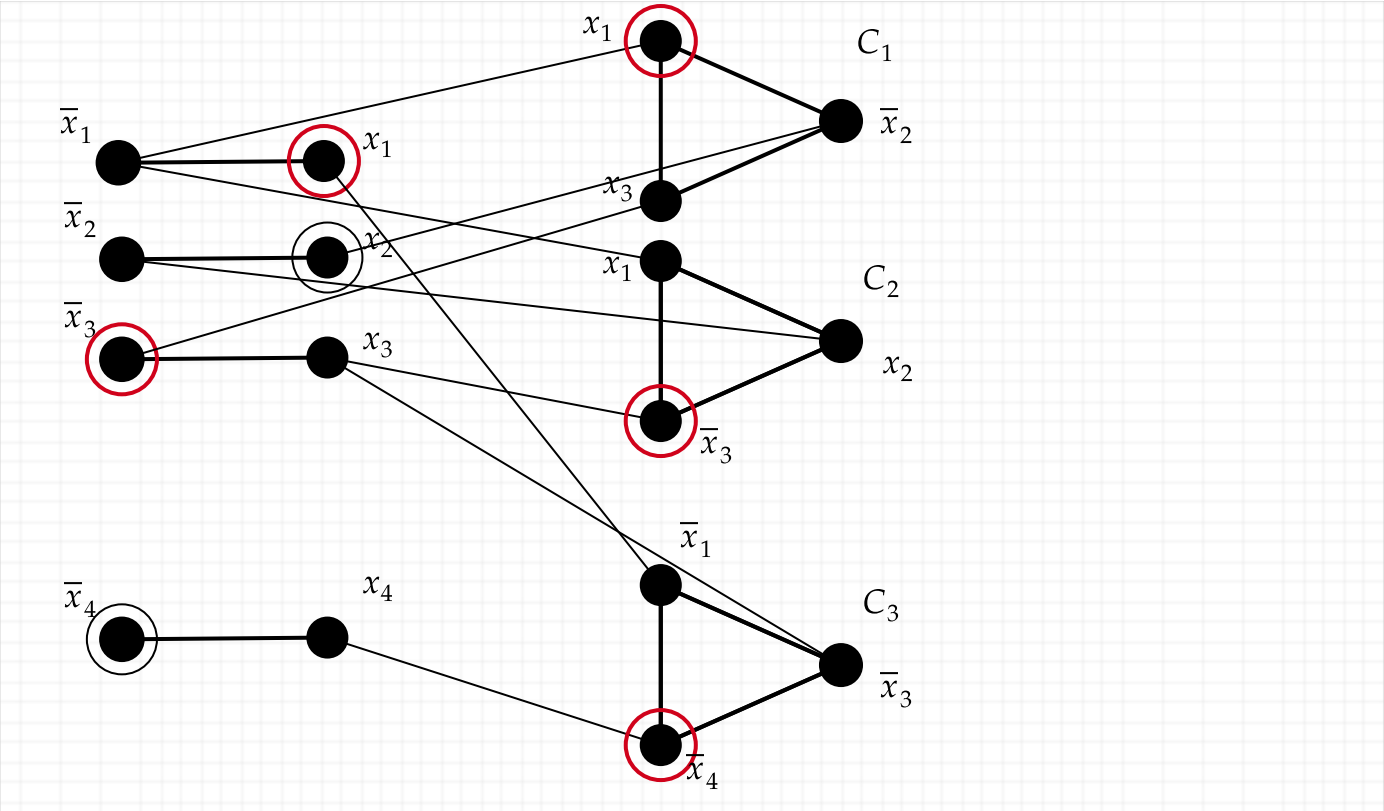
$$\phi = (x_1 \vee \overline{x}_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee \overline{x}_3 \vee \overline{x}_4) .$$

There are 9 crossing edges in all:

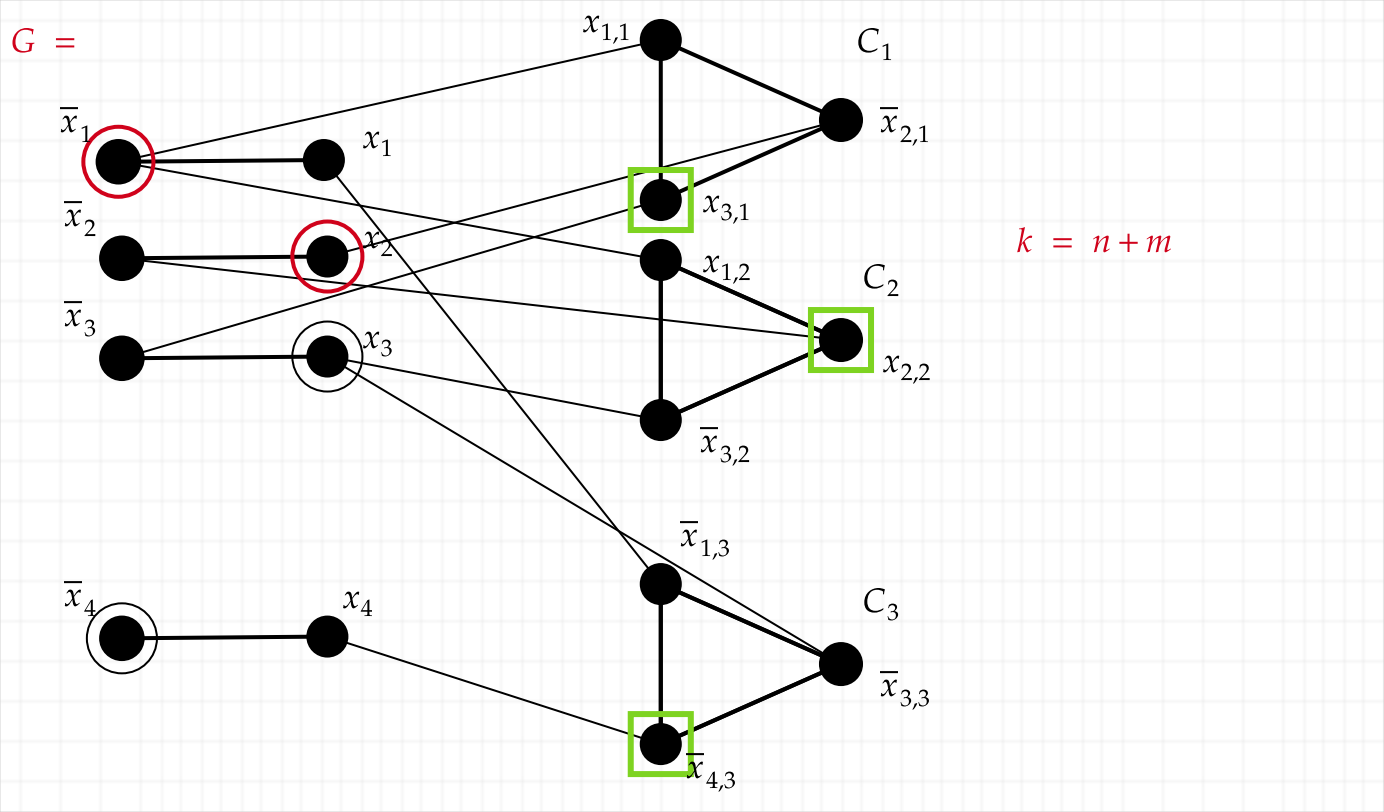


Note that a choice of vertices for S is not part of G---not part of the reduction function f itself. It is only part of the analysis of why the reduction is correct.

To illustrate the analysis, note that the example formula ϕ is satisfiable. In fact, it has many satisfying assignments. (To make a strict 3CNF formula that is unsatisfiable and not use trivialities like duplicate literals in the same clause, one needs to have at least 8 clauses.) For
$\phi = (x_1 \vee \overline{x}_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee \overline{x}_3 \vee \overline{x}_4)$,
one of them is to set x_1 true and x_3 false; then x_2 and x_4 become "don't-cares":



Or we can try setting x_1 false and x_2 true:
$\phi = (x_1 \vee \overline{x}_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee \overline{x}_3 \vee \overline{x}_4)$,



This blocks two of the literals in C_1. We have to set x_3 true. This blocks $\overline{x}_3$ in C_2 and x_1 is already blocked there. Luckily we can choose x_2 in C_2. Since we already have $\overline{x}_1$ as an option in C_3 (but not $\overline{x}_3$), and variable x_4 is not connected elsewhere, it is again a don't care.

One other thing happened in the diagram: each clause node added a subscript for the clause. This enables us to define the reduction formally by specifying the graph in set notation:

$$V = \{x_i, \overline{x}_i : 1 \le i \le n\} \cup \{x_{ij} : C_j \text{ has } x_i\} \cup \{\overline{x}_{i,j} : C_j \text{ has } \overline{x}_i\}$$
$$E = E_{rungs} \cup E_{clauses} \cup E_{crossing}$$
$$E_{rungs} = \{(x_i, \overline{x}_i) : 1 \le i \le n\}$$
$$E_{clauses} = \{(x_{i,j}, x_{k,j}) : C_j \text{ has } x_i \text{ and } x_k\} \cup \{(\overline{x}_{i,j}, \overline{x}_{k,j}) : C_j \text{ has } \overline{x}_i \text{ and } \overline{x}_k\}$$
$$\cup \{(x_{i,j}, \overline{x}_{k,j}) : C_j \text{ has } x_i \text{ and } \overline{x}_k\}.$$

[Side Q: Do we need to add " $\cup \{(\overline{x}_{i,j}, x_{k,j}) : C_j \text{ has } \overline{x}_i \text{ and } x_k\}$?]

$$E_{crossing} = \{(x_i, \overline{x}_{i,j}) : \overline{x}_i \in C_j\} \cup \{(\overline{x}_i, x_{i,j}) : x_i \in C_j\}.$$

[Side Q: Do we need the second set here? Could we "nicely" condense it using notation $\ell_{i,j}$?]

And, of course, $k = n+m$ completes the definition of the reduction function $f(\phi) = (G, k)$. The one benefit of laying out these sets is that they show exactly how to *compute* the graph, and how big it gets. We have $|V| = 2n+3m$ and $|E| = n + 3m + 3m = n + 6m$. Both are in fact *linear* in the size

order-$(n+m)$ of ϕ. The edge lists can be streamed in one pass through the variables and clauses. [Note that although I have not settled on any one formal definition of "streaming algorithm", the idea of them is useful to sharpen the understanding of how the reductions are efficiently computable.] This is indeed a quasilinear-time (**DQL**) reduction.

So we have given the **C**onstruction, shown that its **C**omplexity is well within polynomial time, so it remains to show **C**orrectness: $\phi \in 3SAT \iff f(\phi) \in INDSET$. That is, we need to show G has an independent set S of size $k = m+n$ (the max possible size) $\iff$ the 3CNF formula ϕ is satisfiable.

($\implies$): Given S, it has exactly n nodes from rungs and one node from each clause. For each i, S has either x_i or $\overline{x}_i$. The choices determine a unique truth assignment a. Now consider any clause C_j and let $x_{i,j}$ or $\overline{x}_{i,j}$ be the label of the node chosen. In the former case, there is a crossing edge from $x_{i,j}$ to $\overline{x}_i$. Now $\overline{x}_i$ cannot be the node in S from the i-th rung because that would give S a clash. So the rung node in S must be x_i, so the corresponding assignment makes x_i true, and that satisfies the clause C_j. In the latter case, there is a crossing edge from $\overline{x}_{i,j}$ to x_i. Now x_i cannot be the node in S from the i-th rung because that would give S a clash. So the rung node in S must be $\overline{x}_i$, so the corresponding assignment makes x_i false. Since C_j has $\overline{x}_{i,j}$ in this case, that likewise satisfies the clause C_j. Since C_j is arbitrary, this means a satisfies ϕ.

($\impliedby$): Suppose a satisfies ϕ. Form S by taking the n rung nodes set true by a and choosing one node from each clause that is satisfied. Then by similar reasoning about the crossing edges, S is an independent set of size $n+m$ in G. [Note that even after fixing a, where you've made choices also for "don't-care" variables, there may be multiple S sets because two or three nodes might be satisfied in any given clause. So it is not a 1-to-1 correspondence. But it does have the property Levin cared about, which is that a choice of S uniquely identifies a satisfying assignment.] ☒

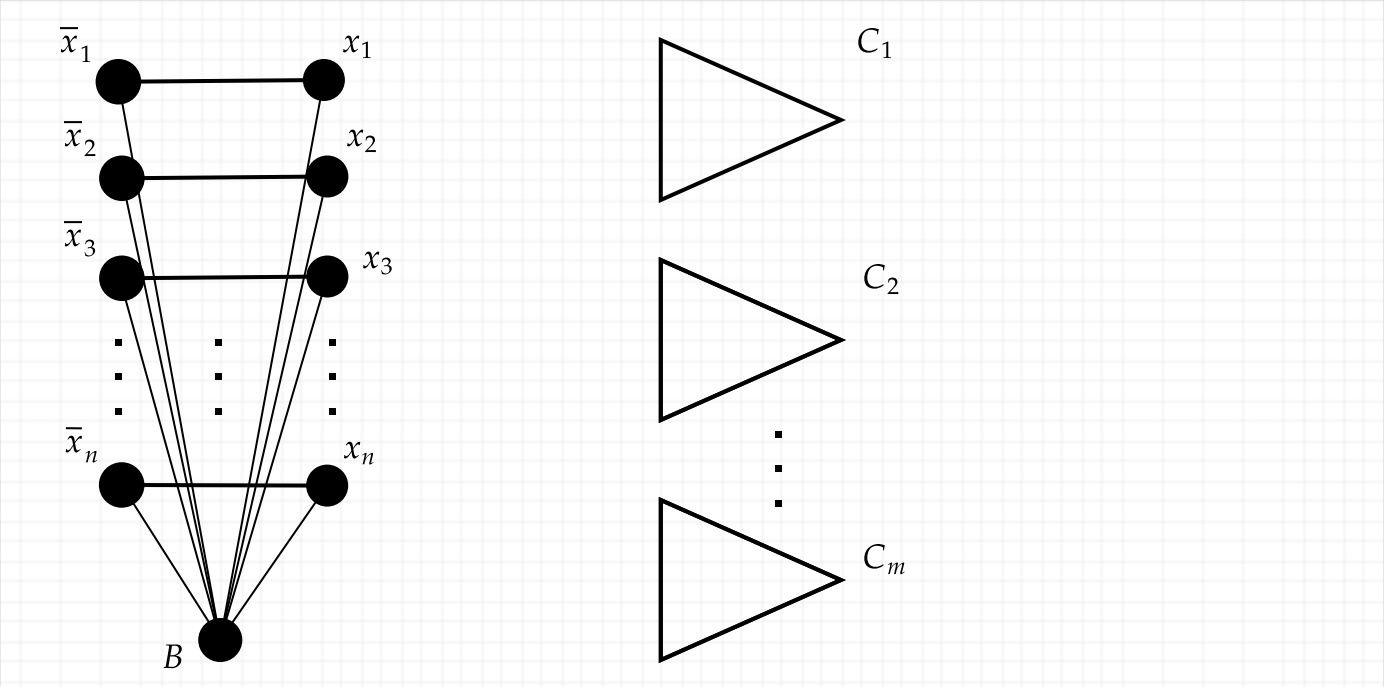
Since $IND\ SET \leq_m^p CLIQUE$ and $IND\ SET \leq_m^p VERTEX\ COVER$ these problems (which we showed to be in NP) are also NP-complete. Now we consider the Graph 3-Coloring Problem.

G3C
Instance: Just an undirected graph $G = (V, E)$ (no k).
Question: Is there a map $\chi : V \to \{R, G, B\}$ such that for all $(u, v) \in E$, $\chi(u) \neq \chi(v)$?

The Greek chi for "chromo-" meaning "color". The language of 3-colorable graphs is clearly in NP: we just guess the coloring, which is a string in $\{R, G, B\}^n$, and verify the coloring on each of $m \leq \binom{n}{2} = O(n^2)$ edges. To show it is NP-complete, we use the same basic rungs-and-gadgets layout, but with one or two twists.

The first thing to think about is how to establish a correspondence between colorings and truth

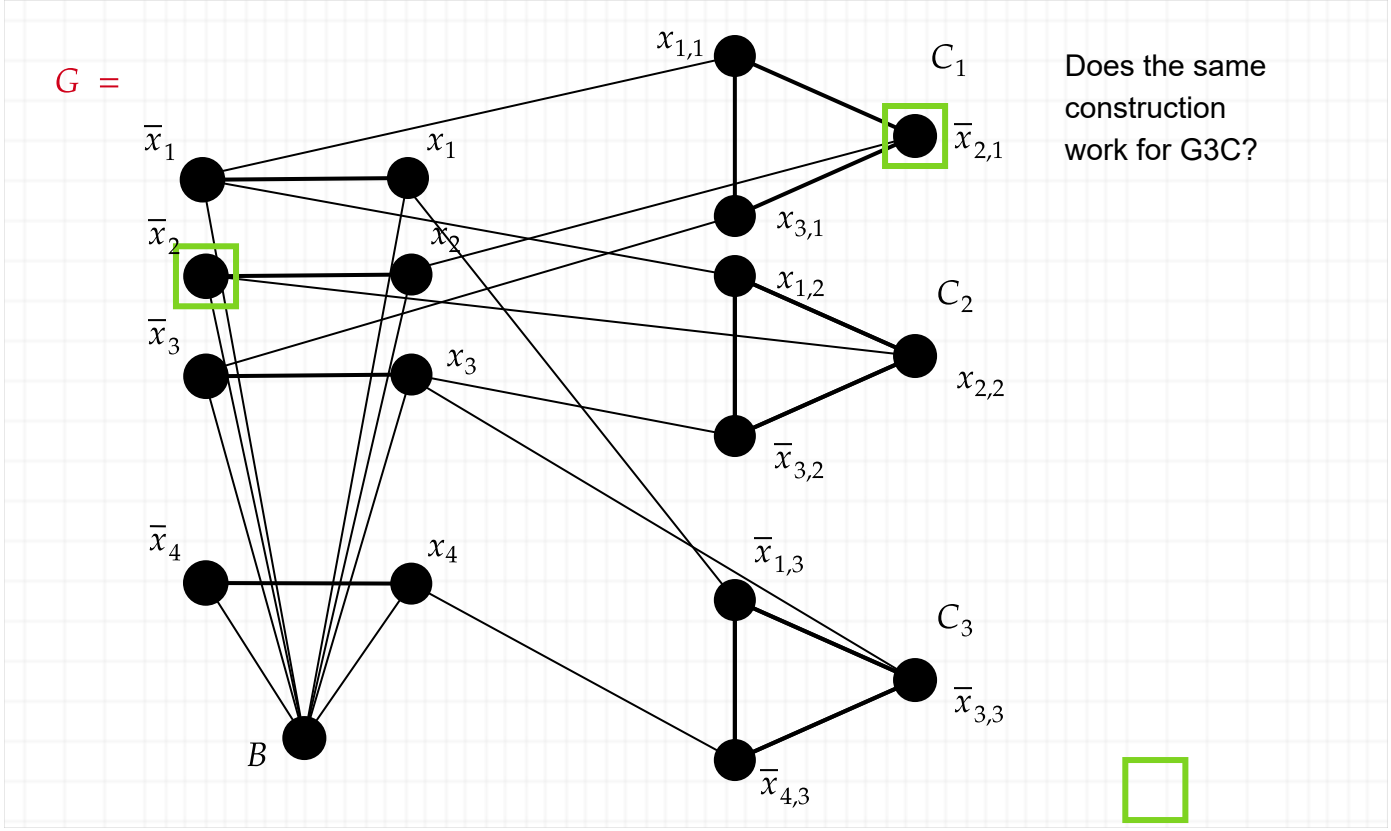
assignments to begin with, before thinking about "good" colorings (i.e., those that meet the "such that" property of having no monochrome edges) *vis-à-vis* satisfying assignments. The natural idea is to give each rung an edge so that each x_i and $\overline{x}_i$ pair must be given different colors so that one color stands for true and the other for false. Well, we have to limit that to two colors for each rung, so we do so by connecting *all* $2n$ rung nodes to a special node called B for the intent to color it blue. So on the ladder side, we have:



This forces each rung to use one R and one G. Now incidentally, $\phi(B) = B$ is not something the reduction is able to define---it is not part of G. But any good coloring remains good under any of the 6 permutations of the colors, so it is "wlog." that we presume $\phi(B) = B$. This leaves R and G for the rung nodes. It is natural to have G stand for the literals that are made true, R for false, but this is where we have to be careful. The permutation that swaps R and G while keeping B fixed stays good, but if flipping an assignment a like 1010 to 0101 satisfies ϕ one way but not the other, there could be a mismatch on correctness requirements.

Let us go ahead. The next question is, can we re-use the clauses-as-triangles idea? With the same crossing edges? Let's try it for the same example formula:

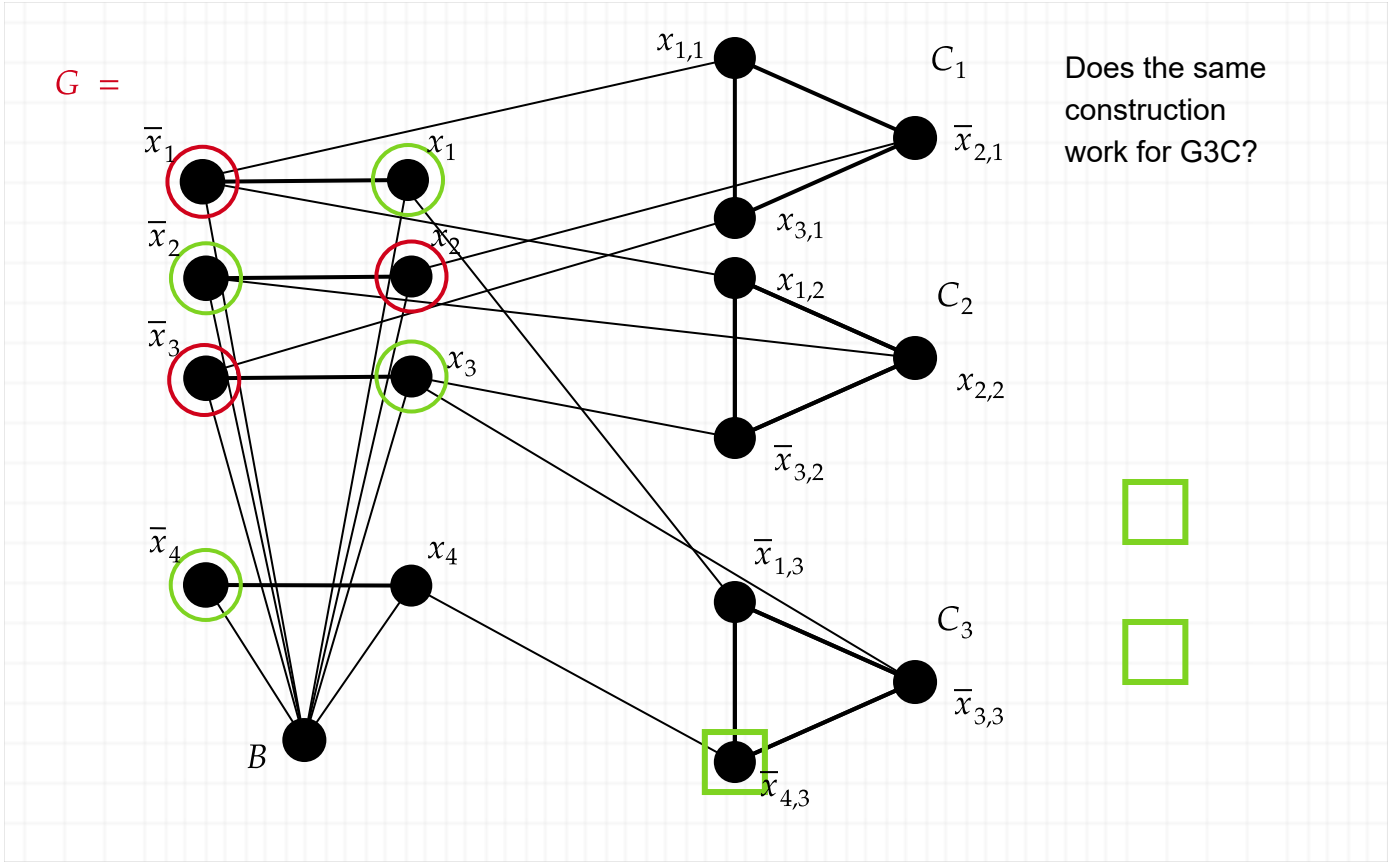
$$\phi = (x_1 \vee \overline{x}_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee \overline{x}_3 \vee \overline{x}_4)$$



Here's the deal: If we have a 3-coloring χ, it has to use G once in each clause triangle and once in each rung. If x_{ij} is green in clause C_j then its crossing edge goes to $\overline{x}_i$ in rung i. This had to be red, so x_i in the rung is green. This means x_i was set true, so C_j is satisfied. The reasoning for a negative literal $\overline{x}_{ij}$ being green in C_j is symmetrical: the crossing edge goes to x_i in the rung, which must be red, so x_i is set false, so $\overline{x}_i$ satisfies C_j. Therefore we get the ($\Longleftarrow$) direction that G being 3-colorable implies ϕ is satisfiable.

The $\Longrightarrow$ direction hits a possible snag, however: Suppose ϕ is satisfiable, but only by assignments that make all three literals in some clause true. It's not just that we can't color all three nodes in the clause green, it's that their crossing edges go to red nodes in the rungs. Suppose this happens for clause C_1 in our example:

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



Now the clause C_1 is "redlocked": we can't color any of its nodes red, so we cannot color it. Note, however, that when an assignment fails to satisfy a clause, the resulting "greenlock" is exactly what we want for correctness in the $\Longrightarrow$ direction. This is what happens to C_3 if we set x_1, x_3, and x_4 all true. So we cannot fix the "redlock" issue without damaging the "greenlock" feature.

Unless, that is, we can invoke an extra condition that "redlock" never happens: that no assignment can satisfy all three literals in a clause. This is a condition that the initial "scholia" on the Cook-Levin reduction allow us to invoke. Then the $\Longrightarrow$ direction goes through: In every C_j, take one node that is satisfied and the other not satisfied. The crossing edges make it good to color the former green and the latter red. The blue color B can then be used for the third node in the clause.

Thus if ϕ is "Not All Equal"-satisfiable then G is 3-colorable. And the original ($\Longleftarrow$) direction also works this way: the red node in the clause cannot be satisfied. Thus we actually get **NAE-3SAT** $\leq_m^p$ **G3C**. This is good enough to show that **G3C** is NP-complete. And to top it off, if a is an "NAE" satisfying assignment, then so it its flip a'. So the symmetry in the coloring is a feature, not a bug.

If, on the other hand, we want to do the reduction strictly from 3SAT without special Cook-Levin appeal, then we need to modify G---as the ALR chapter does. This builds on the "governing blue node" idea to enforce an asymmetry between red and green as well.