Now that we *know* that SATISFIABILITY is NP-complete, how can we use it to show that other problems are NP-complete, without going to the same amount of trouble all over again?

Once you've established that SATISFIABILITY (SAT) is NP-complete, you can use this knowledge to show that other problems are NP-complete without going through the same amount of reduction work for each problem. This is achieved using the concept of "reduction chains" or "transitive reduction," based on the following principle:

If you can demonstrate a polynomial-time reduction from a known NP-complete problem (e.g., SAT) to another problem (let's call it Problem X), then you can conclude that Problem X is NP-complete without repeating the reduction process for Problem X.

Here's how this process works:

1. Start with SAT: You know that SAT is NP-complete, and you've already shown this through a reduction.

2. Choose a New Problem: Identify the problem you want to prove as NP-complete. Let's call it Problem X.

3. Prove the Reduction: Construct a polynomial-time reduction from SAT to Problem X. In other words, show that you can take any instance of SAT and transform it into an equivalent instance of Problem X in polynomial time.

4. Conclude NP-Completeness: Once you've successfully demonstrated the reduction from SAT to Problem X, you can now conclude that Problem X is NP-complete. This is because you've shown that SAT, a known NP-complete problem, can be reduced to Problem X, which implies that Problem X is at least as hard as SAT.

This method saves you from having to perform the full reduction process for each new problem you want to prove as NP-complete. Instead, you leverage the existing knowledge of SAT's NP-completeness to establish the NP-completeness of other problems efficiently.

Keep in mind that, for this method to work, the reduction from SAT to Problem X must be valid and polynomial-time. If you can successfully create this reduction, you can confidently add Problem X to the list of NP-complete problems without repeating the entire reduction process for each problem. This is why NP-completeness proofs often rely on a well-established set of known NP-complete problems and a transitive reduction process.