CPSC 320: Intermediate Algorithm Design and Analysis

Tutorial: Week 13 (NP-completeness worksheet solution)

Author: Susanne Bradley

Today we're going to go through a step-by-step process for proving a problem is NP-complete. The following problem statement is taken from the CPSC 320 2016W1 offering, written by Steve Wolfman.

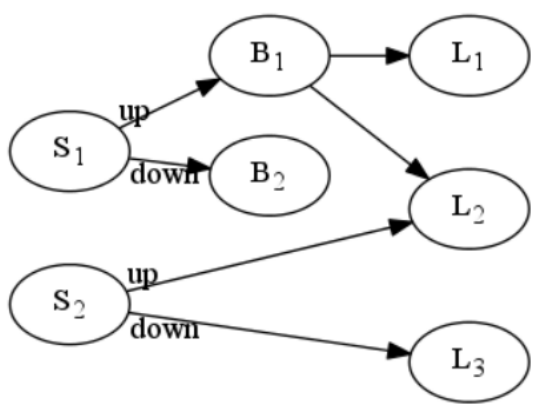
Transformers

In the ELEC problem, you're given a network of electrical wires which can be represented as a directed, acyclic graph (DAG) with three types of nodes:

- "Switch" nodes supply power. They have **no** wires coming in and two wires going out labeled "up" and "down". They also have a switch. If the switch is in the up position, then power (electricity) flows into the up wire. If the switch is in the down position, then power flows into the down wire.
- "Branch" nodes can have one wire coming in (which may or may not carry power) and any number of wires going out. If the wire coming in carries power, then all wires going out also carry power. Otherwise, none of the wires carries power.
- "Load" nodes represent electrical devices that must be powered. They have one or more wires coming in and none going out. If any wire coming in carries power, the load is powered. Otherwise, it is not.

The solution to an ELEC instance is YES if some configuration of the switches powers all the loads; otherwise, it's NO.

1. Indicate a configuration of the switches in the following network that powers all the loads by writing "up" or "down" on each switch node. (Switch nodes are labeled S, branch nodes B, and load nodes L.)



SOLUTION: A working configuration would be S_1 is up (which powers L_1 and L_2) and S_2 is down (this powers L_3).

Proving ELEC is in NP

Complete the following proof that ELEC is in NP (by filling in the parts following the "..."):

A good *certificate* for ELEC is...

A configuration of the switches (i.e., whether each switch is pointing up or down).

We can check this certificate in polynomial time by...

For each switch, check which load nodes L_i are powered by it (the certificate tells us whether the switch is pointing up or down; so, follow the appropriate edge from each switch, and record which load nodes it leads to). The certificate is valid if and only if all load nodes are powered. This essentially amounts to a search from each switch S, which can be done in linear time in the number of edges in the graph (or, in the case of the ELEC problem, the number of arrows in the network).

Proving ELEC is in NP-hard

We need to find an NP-hard problem to reduce to ELEC. *Before we move on to the next page,* what problems have we encountered so far that might be good choices? (I.e., what NP-hard problem kind of *sound like* ELEC?)

Proving ELEC is in NP-hard, continued...

We are going to reduce from SAT to ELEC! (So congratulations if SAT was among your guesses to the previous question!)

Complete the following reduction (by filling in the parts following the "..."):

Define switches S_i in ELEC that represent...

The literal x_i in the SAT instance.

For each switch S_i, define a branch node B_1^i that connects to S_i's "up" wire, and a branch node B_2^i that connects to S_i's "down" wire.

Define load nodes L_j in ELEC that represent...

The jth clause in the SAT instance.

Connect branch nodes to load nodes as follows: ...

If the literal x_i is in clause j (i.e., variable x_i is true in clause j), connect B_1^i to L_j (here we are assuming that "up" is equivalent to TRUE, but we could have done it the other way as well). If the literal $\bar{x}_i$ is in clause j (i.e., variable x_i is false in clause j), connect B_2^i to L_j.

Solve the ELEC instance. Then, the answer to SAT is YES if and only if...

The answer to ELEC is YES.

Proving correctness of your reduction to ELEC

Complete the following proof of correctness (by filling in the parts following the "..."):

Consider the case where the answer to the original SAT instance is YES. This means there exists a truth assignment such that all clauses in the instance evaluate to TRUE. We can then construct a solution to our reduction's ELEC instance as follows: ...

For the truth assignment of each literal x_i, do the following:

If x_i is True, set switch S_i to be pointing up. If x_i is FALSE, set switch S_i to point down.

Because this is a working certificate for SAT, we know that clause c_j in SAT contains (at least) one variable x_i such that x_i is true in c_j. If the literal x_i appears in c_j and x_i is True, then in our ELEC instance, load L_j is connected to B_1^i, which has power going to it because the switch S_i points up. Similarly, if literal $\bar{x}_i$ appears in c_j and x_i is False, then in our ELEC instance, load L_j is connected to B_2^i, which has power going to it because the switch S_i points down.

Since all the SAT clauses evaluate to True, we know that all loads in ELEC have power with this particular configuration of switches, which means our reduced ELEC instance is a YES instance.

Therefore, if the SAT instance has answer YES, our reduction will return YES.

Now, consider the case where the answer to our reduced ELEC instance is YES. This means there exists a...

Configuration of switches such that all load nodes have power.

We can therefore construct a solution to the original SAT instance as follows: ...

For each switch configuration, do the following:

If switch S_i point up, set the variable x_i to be True. If switch S_i point down, set the variable x_i to be False.

Each load node L_j now has power from (at least) one switch, which we'll call S_k. In SAT, this means the assignment to the kth literal makes clause j evaluate to True (i.e., either x_k is in clause j and x_k is True, or $\bar{x}_k$ is in clause j and x_k is False). The fact that all load nodes have power means that every SAT clause contains at least one true literal.

Therefore, if the answer to the reduced ELEC instance is YES, the answer to the original SAT instance is also YES. This completes our proof.