Lab 1

Due March 13, 2020 by the end of the day

OPTION 1

The Clay Mathematics Institute has offered cash prizes if one can solve their fairly trivial problems. One such problem is the P vs NP problem. If you want to get credit for this option, you must complete this problem and split the cash prize money with the teaching team.

OPTION 2

Alternatively, if you have enough money because you got a sweet, sweet, signing bonus, you can choose to explore a game that is NP-Complete. If you recall from the homework for this module, you already saw how Sudoku can be formulated in a way that can be solved by a SAT solver. The purpose of this option is to go a little bit beyond that: to be able to prove the NP-Completeness of these games as well providing a formulation (and potential solver) to the game you choose.

Feel free to do your own Googling (or Binging, or DuckDuckGoing) to find a game, but here are some options in case if you are not feeling the creativity:

- Tetris
- Battleship
- Rubik's Cube
- Set
- Will Derksen master hacker simulator (not sure if this is NP Complete, but if you can prove that it is you get some sort of prize . . . well maybe depending on how much we actually care).
- and more . . .

Note that if your game reduces from integer programming (aka 0-1 programming), we will be covering that topic later. You can either choose to learn about that early (if you think we are incompetent teachers), or wait to have the full AA experience. An additional note that you should refrain from choosing sudoku as the game for this lab, unless you are on a single person team (and can't find other games well-suited for your learning goals).

Submission

Your submission for this lab should include the following components:

Description

A description of the game and, if necessary, the description of the specific formulation of the game that is NP complete. (Remember that pictures are worth 1000 words and also that we are functionally illiterate, so while long descriptions will be helpful, images and figures will be too.)

Proof

You need to prove that your problem formulation is NP-Complete. To do this, you should include a proof of reduction from one of of the Karp's 21 NP-Complete Problems or the Traveling Salesman Problem:

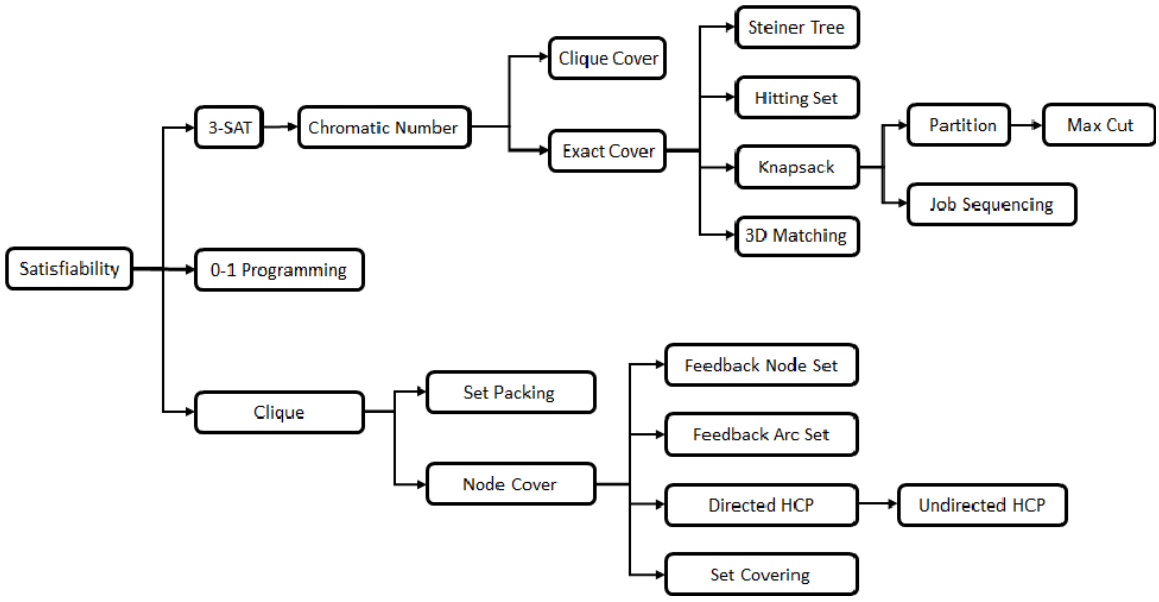

Figure 1: This image shows Karps 21 reductions as well as what they reduce to. Source

You might have to provide additional constraints or information to perform this reduction, and that's fine. It is also fine to use a modification of the original game in order to prove NP Completeness. Additionally, if your reduction necessitates more intermediate reductions in order to complete your final reduction, include them.

Implementation

So here you have two options (depending on how much work you did for the other parts). You could either a) try to code your own custom solver for your problem based on the reduction (feel free to also outsource any necessary starter code if you want), or b) run a CNF-SAT configuration (which you will have to figure out and explain to us) of your problem through an existing SAT Solver and analyze the results of that (in terms of game 'accuracy,' time, ease of use, etc.)

Edit: We anticipate you testing your implementation via test cases. Please upload any files related to the tests to your Github repository, and include at least one sample test case and output in your submission write-up.

Resources

If you relied on any other sources other than your brain, you should probably cite them.

Final Note

Different games will have different levels of work for each of these sections. You should choose what you are more interested in and try to find games that fit your own personal learning goals.

Happy lab-ing!