

CS200 - Programming Assignment 1: Computer Assisted Proofs

Motivation

As you've seen, proving things mathematically can be difficult. A current active field of research at the intersection of computer science and mathematics involves using computers to help find proofs. While some work is being done to use AI techniques to write proofs in the way a human does, most computers are used to do "proof by exhaustion," which is when a computer tries all possible cases of interest. For example, for a [proof of the Pythagorean triples problem](#), a computer had to check a trillion different cases (which took 2 days on a supercomputer). When a proof is created with the help of a computer, we call it a "computer-assisted proof."

In this programming assignment, you will write a program to help with a computer-assisted proof. Your program will take as input a complex predicate involving several variables, which is the thing we are trying to prove is true. Your program will deduce what assignments of the variables make the statement true. I suggest your program use a "proof by exhaustion" approach by testing all possible values of the variables.

Guidelines

Please read and abide by the [honor code guidelines](#) in the syllabus.

Please read the [rubric](#) so you know how you will be graded. For example, turning in a program that compiles and runs without errors but does nothing will earn you more points than a program that is close to working but does not compile or contains errors on running.

There are two options for this assignment: Standard and Challenge. (You must turn in only one of the two options.) You will be graded using [this rubric](#). Since the rubric is out of 30 points, if you earn X points, then your grade will be $X/30 \times .85$ for a Standard assignment and $X/30 \times .95$ for Challenge. Thus a poorly done Challenge program could give you a worse score than a well executed Standard program.

You may use any functions, methods, or packages that you find useful (for example, in python `eval` and in java `parseBoolean` may be helpful), and you may use code snippets that you find on, for example, Stack Overflow, but you may not use code that effectively implements the solution. I expect that you will have to figure out some new tools (like `eval` or `parseBoolean`) to do this assignment, and I expect you to do some web searching to find them and learn how to use them.

Put a multi-line comment at the beginning of your program. It should contain:

- Your name
- "Programming Assignment 1"
- "Challenge" or "Standard"
- The name of anyone you worked with and the nature of your collaboration
- Sample output from your program

- The amount of time (approximately) that you spent on this assignment

Standard Assignment

Write a program in python or java that takes as input a string that consists of a single predicate involving the variables “A”, “B”, and “C”, and the operators ‘and’, ‘or’, and ‘not’, as well as parentheses. Your program should output a list of assignments of of the variables “A”, “B”, and “C” that cause the predicate in the input to be true. If the predicate only contains the letters “A” and “B”, for example, you should still include assignments for “C” in your output.

Here is a sample output:

```
>>> deduce("((A and B) or not C) and (C or B)")
A B C
-----
T T T
T T F
F T F
```

Challenge Assignment

Write a program in python or java that takes as input a string that consists of a single predicate involving variables whose names consist of capital letters (for example “CAR”, “X”, and “UNICORN”, are all valid variable names), the operators ‘and’, ‘or’, and ‘not’, as well as parentheses. Your program should allow for any number of variables (not just 3, as in the example). Your program should output a list of assignments of of the variables such that the predicate in the input is true. The output should be formatted in a way that is easy to read, as in the example below (although you are free to format differently).

Here is a sample output:

```
>>> deduce("((CAR and X) or not UNICORN) and (UNICORN or X)")
CAR X UNICORN
-----
T   T T
T   T F
F   T F
```