Project 1

This project is due on Monday, June 10 at the beginning of class. It is optional and may be completed in lieu of assignment 1. This project is more challenging than the assignment but it is worth 120% of the points.

Task

Your task is to write a program to evaluate statements of propositional logic. That is, given a proposition, your program should determine whether that proposition is a tautology. Hint: use sequents. Sequent calculus is a powerful and simple tool that can be used to verify the truth of a statement in propositional logic. For a helpful list of inference rules see the section titled "Proving logical formulas" on this page.

Requirements

- Your program should be able to handle input with the following boolean operators: $\wedge, \vee, \Rightarrow, \neg$
- Also, the user should be able to specify order of operations with parentheses
- Your program should display the sequence of logical steps it took to prove (or disprove) the given statement.
- For any given statement, your program should output "Formula proven" or "Formula unprovable"
  - There may be a few edge cases that loop infinitely. This is due to Hilbert's Entscheidungsproblem. However these edge cases are fairly rare in practice.

Examples

See the pages linked above on sequents and sequent calculus to understand the rules of inference used below.

- Prove a common tautology

  Input: P or not P

  Output: $\vdash P \vee \neg P$

  1. $\vdash P, \neg P$
  2. $P \vdash P$: TRUE

  Formula proven.

- Prove relation by implication

  Input: (P implies Q) implies (not P or Q)

  Output: $\vdash ((P \Rightarrow Q) \Rightarrow (\neg P \vee Q))$

  1. $((P \Rightarrow Q) \vdash (\neg P \vee Q))$
  2. $((P \Rightarrow Q) \vdash \neg P, Q$
  3. (split) $\vdash \neg P, Q, P$: TRUE
  3. (split) $Q \vdash \neg P, Q$
  4. $P \vdash Q, P$: TRUE

  Formula proven.