Problem 8: Theorems AI [Email Submission]

By Ruijie Fang

1 Introduction

You are hired by the university math department on a summer job. Inspired by the recent advancements in ChatGPT-like artificial intelligence, the math professors want you to do some automation to help them churn out more theorems. Typically, mathmaticians prove theorems from a set of ground facts: for instance, they may assume no set can contain every set (to prevent Russell's paradox), or that if a proposition P is false, then $\neg P$ ("not P") must be true (this is called the "law of the excluded middle"). Starting from axioms, they incrementally prove more and more interesting facts, called theorems and lemmas (which are just simple, intermediate facts to help prove more theorems). Theorems, in turn, can imply other threorems. Such a system of proving theorems in mathematics can be formalized in mathematical logic. One logic commonly employed by mathmaticians is propositional logic. In propositional logic, we can abstractly express statements as logical formulas on boolean variables (taking either TRUE or FALSE as values) and logical connectives AND ($\wedge$), OR ($\vee$), not ($\neg$), and IMPLIES ($\rightarrow$). The "IMPLIES" operator tells us how to deduce more facts from existing statements: Given $P \rightarrow Q$, then Q is only true if P can be shown to be true. Given such a system, automated theorem proving works as follows:

1. The user specifies that a set of axioms, each axiom expressed as a boolean variable, must hold. Every boolean variable corresponding to an axiom is assumed to be true.
2. The user specifies a set of implication relationships: "If A is true and B is true, then C holds" where A, B, C are boolean variables that correspond to either axioms (already assumed to be TRUE) or other mathematical statements.
3. The user tells the computer what statement(s) to prove (i.e. to show as true), given the axioms and implication relationships: e.g. "P,Q,R,S must hold given the aforementioned axioms and implication relationships."

Given the set of axioms, implication relationships, and a proof goal, your job is to decide if the variables (corresponding to the mathematical statements) in the proof goal can be shown to be true.

2 Input

Note: All strings are assumed to be less than 100 chars in length and consists of alphanumeric characters only. The first line of the input will be a single number N ($1 \leq N \leq 500$), telling you the number of lines of input to read next. Each line following the first line will be one of the following three forms:

1. The user specifies an axiom. The line will read "axiom XXX" where "XXX" is a string representing the name of the axiom.
2. The user specifies an implication relationship. The line will read "imply Y M p_1 p_2 ... p_M", where "Y" is a string representing the name of the statement/boolean variable being implied by, and M is a number ($1 \leq M \leq 30$) specifying how many statements altogether imply Y. M strings follow, telling us the name of each such statement. The line is essentially specifying "$(p_1 \wedge p_2 \wedge ... \wedge p_M) \rightarrow Y$".
3. The user specifies a proof goal on the last line, expressed as an AND of variable names. The line will read as "prove K q_1 ... q_K" with K being a number ($1 \leq K \leq 30$) specifying the number of statements (correspondingly variables) you would need to show TRUE, and q_i being space-separated strings specifying the names of the statements.

Note: the situation in (3) is guaranteed to only appear on the last line.

3 Output

The program should output one line, "provable" or "unprovable".

4 Grading method and Submission

Please e-mail your solution (written as source code) in. We will manually run it on a manually crafted set of inputs after the competition. Make your best attempt to handle corner cases —since we manually inspect all source code, corner cases will not impact your grade should your solution be mostly correct.

Email your code (in one file, with proper file names like Theorems.py or Theorems.cpp) to this email address: coscon.submit@gmail.com. If you must resubmit, respond to the thread where you sent your original submission; we cannot guarantee that your resubmission will be graded otherwise.

5 Examples

The following input should be "provable":

```
6
axiom ax1
axiom ax2
imply 2 ax1 ax2 thm1
imply 1 thm1 thm2
imply 1 thm1 thm3
prove 1 thm3
```

The reasoning behind this is that if we assume ax1 and ax2 to be true, they altogether imply thm1 to be true. thm1 in turn implies thm2, and thm3, so thm3 should be provable. The following input should be "unprovable":

```
7
axiom ax1
axiom ax2
imply 2 ax1 ax2 thm1
imply 1 thm1 thm2
imply 1 thm1 thm3
imply 1 thm4 thm5
prove 1 thm5
```

The reasoning behind this is that if we ax1 and ax2 to be true, they only imply thm2 and thm3. thm5 is implied by thm4, but thm4 is not necessarily true — it is not implied by an axiom. By contrast, the following input should be "provable" (shown next page):

```
10
axiom ax1
axiom ax2
axiom ax3
imply 2 ax1 ax2 thm1
imply 1 thm1 thm2
imply 1 thm1 thm3
imply 1 ax3 thm4
imply 1 thm4 thm5
prove 1 thm5
```