

Homework 5: a computer activity

Randall Holmes

October 22, 2020

I have written an interactive program in Python which allows you to show that statements are derivable in Post's system (and so in the propositional logic fragment of *Principia Mathematica*). You type formulas one after the other to the computer, and it tells you whether it can derive them from the axioms and formulas you have previously derived, and how it did so.

You will need to download and install Python. When you have done so, the way to use the program is to edit the source `post.py` in IDLE (which is installed along with Python) then choose Run from the toolbar and press Run Module to get a window in which the commands in `post.py` have been executed and you can type more Python commands. You do not need to know how to program in Python to do the lab.

The only commands you need to know are

```
propose("<a formula>")
```

which will attempt to prove the line `<formula>` and return either 'bad' or a demonstration. It will only be able to derive a line if it can deduce it fairly immediately from applications of Rules II and III of lines you have derived. If it is derivable from lines you have proved by a single application of Rule II or Rule III it will be able to do it. It is slightly smarter: it can look for extra applications of Rule II to make *modus ponens* (Rule III) applicable.

and

```
showhistory(n)
```

which will show you the n most recent propositions proved, or all of them if n is 0.

Formulas are of four kinds

propositional letters: The letter p , q , or r followed by a string of digits is a proposition. These letters by themselves are not propositions: you have to have an index.

negations: \sim followed by a formula is a formula.

binary connectives: If F and G are formulas, $(F \vee G)$, $(F \wedge G)$, $(F \supset G)$, $(F = G)$ are formulas, disjunction, conjunctions, implications, and biconditionals respectively. The program knows the definitions of the non-primitive connectives.

line references: $*$ followed by a string of digits is an abbreviation for that line in the history. If you want to show that P follows from line 14, propose $*14 \supset P$

note about whitespace: You can put as many spaces as you want next to one of the binary connectives. You cannot insert spaces after \sim or $*$ or after $($ or before $)$.

note about parentheses: The `propose` command will supply the outermost pair of parentheses if you leave it off. Otherwise, all parentheses must be shown.

The assignment is to replicate the proofs of Excluded Middle, Lemma 1, and Lemma 2 in the Post notes. When you have done it, paste your conversation with the computer into a document and email it to me. I have done them.

You will not need to type every line I have typed: you can explore which ones are needed. All you will be doing is entering lines: the software will tell you whether the line follows by the rules from the earlier lines given.

In the proof of Lemma 2, a hint is needed. The software does not know the rule of transitivity of implication.

when you have

13. $A \supset B$

25. $B > C$

you will not usually be able to prove $A > C$ just by typing it. But there is a trick. Using Axiom 5, the prover will be able to prove $(A > B) > (A > C)$, which you can write in the shorthand form $*13 > (A > C)$. Further, if the theorem proves an implication and notices that the antecedent of the implication is also provable, it immediately automatically proves the consequent, so proposing $*13 > (A > C)$ will cause it to prove $A > C$ as well. $*25 > (A > C)$ would also work if you have proved Lemma 1. Of course the line numbers in your proof will be different.

You can certainly start working on this right away if you can figure out setting it up. I'm planning to have us all work on it in class on Monday.

If you watch the Wednesday video you can see this program working. I explained something about how it works in the video, and these concepts are of interest to us (quite independently of the program).