

Constructive Logic (15-317), Spring 2021

Assignment 9: Theorem Proving in Prolog

Instructor: Karl Crary

TAs: Avery Cowan, Katherine Cordwell, Matias Scharager, Antian Wang

Submit to Gradescope by Wednesday, November 10, 2021, 11:59 pm

For this homework, you will be submitting Prolog to Gradescope:

- `hw.pl` should go to Homework 9 (code).

Setting up Prolog

1. SSH into `unix.andrew.cmu.edu`
2. Execute this command: `/afs/andrew/course/15/317/bin/317setup`
3. Run Prolog with the `prolog` command (exit with control-D or “halt.”)

Implementing a theorem prover (one more time)

Now that you are experts in implementing **G4ip** in Standard ML, it is time to try doing so in Prolog.

Task 1. Implement a theorem prover for **G4ip** in Prolog. You must define the predicate `prove/1` for determining whether a formula is provable, and use the predefined logical operators (see the accompanying `hw.pl` starter file). You are not required to create the proof tree for these formulas and are simply determining if they are provable using the same **G4ip** algorithm as before. This means that, given a valid *ground* formula a , the query `prove(a)` should succeed (with *true* or *yes*).

We recommend that you implement your new theorem prover from scratch. Do not translate your Standard ML implementation into Prolog! You'll make life harder for yourself, because this problem is designed to be directly expressible as a logic program.

For your convenience, we have provided you with a shell script to test your implementation. You can invoke it by executing:

```
$ ./test_g4ip.sh
```

The test script uses another Prolog file (`g4ip_tests.pl`) to test your implementation. It employs some Prolog features that you have not been taught. We strongly urge you not to use any of those features in your solution. They are unlikely to help, and are very likely to make your code hard to understand.