Project #2 – Writing a Basic Prolog Interpreter
CS 152 Section 5 – Fall 2020
Michael McThrow
San José State University

Introduction

Your task for Project #2 is to write a basic Prolog interpreter that reads in a Prolog source code file that contains facts and accepts queries from the command line.

For example, suppose you have a source code file named pokemon.prolog that contains the following facts:

```
evolution(bulbasaur,ivysaur).
evolution(ivysaur,venusaur).
evolution(charmander,charmeleon).
evolution(charmeleon,charizard).
evolution(squirtle,wartortle).
evolution(wartortle,blastoise).

grass(bulbasaur).
grass(ivysaur).
grass(venusaur).

poison(bulbasaur).
poison(ivysaur).
poison(venusaur).

fire(charmander).
fire(charmeleon).
fire(charizard).

flying(charizard).

water(squirtle).
water(wartortle).
water(blastoise).
```

Your goal is to implement a program that accepts queries with the following prompt:

```
prolog> evolution(bulbasaur,ivysaur)?
true.
prolog> evolution(bulbasaur,X)?
X = ivysaur.
```

```
prolog>
```

You need to implement support for the following features in addition to whatever is implied in the above examples:

- Rules (including the :- syntax and conjunctive queries on the right-hand side).
- Conjunctive queries (e.g., `fire(X),flying(X)?` should result in `X = charizard`.)
- Unification
- Lists

Please use the resolution and unification algorithms described in *The Art of Prolog*.

Do not worry about adding support for numbers, type predicates, cuts, negation, and other advanced features. The only features that we will need to concern ourselves with are the features from the first six chapters of *The Art of Prolog*.

Rules

- **Please stick to the project specification regarding input formats.** I reserve the right to deduct points for implementations that deviate from the specification.
- **You may choose to work with a partner enrolled in this class for this assignment.** This is entirely optional. Only one partner is responsible for turning in the assignment, but both partners will earn the same grade. **Place both partners' names on your submission files.**
- You may choose Java, C, C++, Python 3, or Scheme/Racket as your implementation languages. If you choose to use Scheme/Racket, make sure your code runs in DrRacket using `#lang racket`.
- If you choose to use Scheme/Racket, there are no restrictions on the use of mutation and side-effects. Please write your code as you feel fit.
- Please refrain from using third-party libraries in your code. A third-party library is a library that is not part of the standard of the language you choose. If you are unsure whether a library is a third-party library or not, ask me.
- If you choose to use C or C++, keep in mind that your programs will be run in a Linux environment when I grade them. Please code in such a way where your code can work on either the GCC or Clang compilers.
- **Your code must compile or be interpreted without any syntactical errors in order for it to receive credit.**
- No matter which implementation language you choose, your code *must* be runnable from the command line, and your code must support the prompts *exactly* as described above. This is to facilitate the use of automated testing tools for grading, which makes grading easier and quicker.

- Your submission will be in a *.zip file that contains your source code, a README file describing instructions for how to compile your code (if you used Java), a Makefile if you're using C or C++, and how to execute your code on the command line.

Grading Rubric

- I will be running your Prolog implementation against a battery of tests. If all of these tests pass perfectly, you receive 100% before any applicable bonuses and penalties. If any of the tests fail, you will receive deductions based on the severity of the error: minor mistakes will get minor deductions, while major mistakes or omissions (e.g., no support for lists or conjunctions) will receive deductions of at least 10% each.
- Recall that your code must compile or run without any syntactical issues. If I am unable to compile your code, your assignment gets a grade of zero.