CSCI 162 Lab 6

Purpose: To introduce Prolog.

Objectives: After this lab you should be able to

- Edit and Load and query Prolog databases.

Background:

Prolog's name is from the phrase "PROgramming in LOGic,". It is a special-purpose language. At the heart of Prolog is a logical inference engine that, based on the data supplied by the programmer, generates answers to questions posed by the user. Unlike procedural languages (e.g., Java), Prolog is a language in which statements are logical expressions.

gprolog: This lab is based on the GNU project's gprolog, an open-source implementation of Prolog. Each implementation of Prolog usually offers features and even syntax that differ from other implementations, so the examples here may not all work in another version of the language. gprolog is available for a variety of systems, including Windows, UNIX, and most UNIX-like systems (including Linux).

Prolog Program Components: Technically, you don't create a Prolog program; you create a Prolog database.

In the database are two types of clauses: Facts and Rules.

As the name suggests, a fact is a single piece of information. To represent the fact that the sky is blue, you could use the clause **blue(sky).** . Similarly, **mammal(rabbit).** says that rabbits are mammals. Those are both examples of single argument, or unary, predicates. Facts can have multiple arguments; for example, **plays(john,hockey).** indicates that john plays hockey. Worth noting: Prolog constants are in lower case, variables are in upper case, domains are not defined explicitly, and entries always end with a period.

Tracing: Frequently, your database will cause Prolog to produce answers that you know aren't correct. This means that your facts or rules are incorrect, but finding out which ones are wrong can be difficult. To help, Prolog provides a tracing command that allows you to watch Prolog's reasoning as it happens. It's hard to follow until you see it a few times, but it's very helpful, not only for debugging, but also for learning about how Prolog works.

To turn on tracing, the command is trace. While tracing is on, you can issue any query, and Prolog will step you through its logical process.

Instructions:

1. Create a file called family.pl with the following contents:

```
/* This file contains a database of parents and
children */
parent(hank,ben).        % This means that hank is
parent of ben
parent(hank,denise).
parent(irene,ben).
parent(irene,denise).
parent(alice,carl).
parent(ben,carl).
parent(denise,frank).
```

```
parent(denise,gary).
parent(earl,frank).
parent(earl,gary).

grandparent(X,Z) :- parent(X,Y) , parent(Y,Z).
ancestor(X,Y) :- parent(X,Y).
ancestor(X,Y) :- parent(Z,Y) , ancestor(X,Z).
```

Don't forget the period at the end of each line! Blank lines are OK.

2. With the file created, start gprolog. At the shell prompt, type:

   `gprolog` and press Enter.

You'll see something like this:

```
GNU Prolog 1.2.16

By Daniel Diaz

Copyright (C) 1999-2002 Daniel Diaz

| ?-
```

The last line is the Prolog prompt; it's ready for you to type a command.

3. Load your family database into Prolog, using this command at that prompt:
   `[family].`
   If Prolog finds that everything is in order, it will say something like this:

```
compiling
/home/faculty/beestonj/csci162/prologExamples/family.pl
for byte code...

/home/faculty/beestonj/csci162/prologExamples/family.pl
compiled, 14 lines read - 1832 bytes written, 9 ms
```

Yes

If it's not happy, it will give an error message that probably won't make much sense. Don't panic! It will report the line in the file on which it found the error. Go back to your file, find that line, fix the error, and try again.

4. With the database loaded, you can type in queries. In Moodle give the answers to these queries:

| ?- parent(hank,denise).

| ?- parent(denise,hank).

| ?- grandparent(irene,frank).

When Prolog prints a response and follows it with a question mark, it is asking if you want it to keep searching for more answers. Typically, you do. If so, just press the semicolon, which tells Prolog to keep going.

| ?- parent(hank,X).

| ?- grandparent(hank,X).

| ?-

When you're done and want to leave gprolog, just enter Ctrl-D (that is, hold down the Ctrl key on the keyboard, and press D). You'll be back at the shell prompt.

5. Turn on tracing with the command trace.
   run the command grandparent(hank,X). What trace does it give?
   Note: that you press Return at the question marks after the trace output, but semicolons after the normal results output.
   Turn tracing off with the command notrace.

What to hand in:

Copy and paste your answers to part 4 and part 5 into a .txt file and submit it via Moodle.