CSC 261 Assignment 3: First-Order Logic Programming Fall 2009

Assigned: Friday 9 October 2009
Due: Friday 16 October 2009
Topics: First-Order Logic, Backward Chaining, Logic Programming, Prolog
Objective: Gain greater understanding of first-order relations and inference using logic programming.
Expected Time: 2-5 hours
Collaboration: This homework assignment may be completed individually or in pairs.

Submission: Follow the instructions for submitting programs via P-Web and handing in a printed copy of source code. Your non-programming components should be typed, nicely formatted, and feature a logical organization, complete sentences, and proper grammar, spelling, and punctuation. Only one submission (both paper and digital) per group is necessary.

1 First Order Logic

[From AIMA, Exercise 8.6] Represent the following sentences in first-order logic. Briefly describe the predicates you define. For example, “\(\text{Take}(x, y, z)\) means \(x\) takes course \(y\) in semester \(z\).”

(A). Every person who buys a policy is smart.
(B). No person buys an expensive policy.
(C). There is an agent who sells policies only to people who are not insured.
(D). There is a barber who shaves all men in town who do not shave themselves.

2 Logic Programming

In the following questions, you will eventually create a single Prolog program. They may be written in the file in the order the questions are assigned. Please indicate which portion of the file is relevant to which question through the use of comments (using the \% symbol).

Recall also that variables in Prolog are given in capital letters, while constants are given in lower case letters.

Part A

Primitive predicates are those that are used only as facts in our knowledge base, rather than as the head of some rule (though they may be part of the bodies of rules). Let \(\text{male}(X)\) and \(\text{female}(X)\) be primitive unary predicates, with \(\text{child}(C, P)\) and \(\text{married}(H, W)\) as primitive binary predicates. Note that \(\text{married}\) and \(\text{child}\) are not symmetric; the former takes the husband first with the wife second, while the latter the child first with the parent second.

Using these primitives, we can write the many Prolog rules to express familial relationships. For example,

\[
\begin{align*}
\text{parent}(P, C) & : - \text{child}(C, P). \\
\text{father}(F, C) & : - \text{male}(F), \text{parent}(F, C). \\
\end{align*}
\]

\(^1\)That is, at least those familial relationships that were thought of as “traditional” in one day and age; some states, countries, and provinces may have more modern definitions. For the purposes of making your assignment easier, we’ll stick to the traditional ones here.

1
Along with parent (P,C) and father (F,C) given above, express the following familial relationships in a Prolog program. The bodies of your clauses should use as few terms as possible.

| mother (M,C) :- | wife (W,H) :- |
| son (S,P) :- | husband (H,W) :- |
| daughter (D,P) :- | sibling (X,Y) :- |
| grandfather (G,C) :- | brother (B,X) :- |
| grandmother (G,C) :- | sister (S,X) :- |
| grandson (S,G) :- | uncle (U,X) :- |
| granddaughter (D,G) :- | aunt (A,X) :- |

### Part B

The song “I’m My Own Grandpa” (Latham/Jaffe) debuted in 1947. It was summarized by Niklaus Wirth as follows:

I married a widow (call her w) who has a grown-up daughter (d). My father (f), who visited us quite often, fell in love with my step-daughter and married her. Hence my father became my son-in-law and my step-daughter became my mother. Some months later, my wife gave birth to a son (s1), who became the brother-in-law of my father, as well as my uncle. The wife of my father, that is, my step-daughter, also had a son (s2).

*Algorithms + Data Structures = Programs, 1976*

(i) Step-relations shouldn’t have been reflected in your rules above. Add to your Prolog program two new rules for parent that take this into account. (Recall that the married relation is not symmetric). Note that Prolog requires rules with the same head to be adjacent to each other.

(ii) There are many potential in-law relationships, but the one we’re interested in for this story is son-in-law. Add a new rule for soninlaw (S,P) that defines this relationship.

(iii) Using only the four primitives from part A (male, female, married, and child), add the facts present in the story above. (Use n for the narrator of the story). Note that Prolog requires that facts using the same primitives be adjacent to each other.

(iv) Using Prolog, verify the conclusions drawn by the narrator:

```
grandfather(n,n).
soninlaw(f,n).
mother(d,n).
uncle(s1,n).
```

### How to Submit Your Work

On the MathLAN, you will record your Prolog file and an interaction with the prolog interpreter to a text file, sending the result to a printer, as follows:

1. Type “script output.txt” (sans quotes) to begin recording your session.
2. Type “cat program.pl” to echo your program to the file.
3. Start the Prolog interpreter, load your program, and use the interpreter to verify the conclusions given in 2.B(iv).
4. Exit the Prolog interpreter by typing “halt.”
5. Stop the script session by typing Control+D
6. Print the record of your session by typing “enscript -2 -r output.txt”.
7. Submit your Prolog file digitally to Pioneer web.
8. Turn in your answers to Part 1 and your session printout in class.