CSC 261 **Lab 6: First-Order Logic** **Spring 2018**

Assigned: Tuesday 27 February

Due: Wednesday 7 March, 11:59 p.m.

Objectives:

- Understand **first-order** relations
- Practice using **logic programming** for inference

Collaboration: This laboratory assignment must be completed individually.

Problem 1: Expressing First-Order Logic

Complete AIMA Exercises 8.28(b,c,d,h,j,l). Write your answers in FOL (not Prolog). If you're not using LaTeX, you can copy-paste the Unicode symbols from https://en.wikipedia.org/wiki/List_of_logic_symbols.

Problem 2: Logic Programming

In the following questions, you will create a single Prolog program `grandpa.pl`. Write answers in the file in the order the questions are assigned; indicate which portion of the file is relevant to which question through the use of comments (using the `%` symbol).

Recall that variables in Prolog begin with a capital letter, while constants are in all lower case letters.

Part A

Primitive predicates are those used *only* as facts in our knowledge base, rather than as the head of some rule (though they may be part of the rule bodies). Let `male(X)` and `female(X)` be primitive unary predicates, with `child(C,P)` and `married(H,W)` as primitive binary predicates. Note that we mean `child` to be a biological relation. `married` and `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.[1] For example,

```
parent(P,C) :- child(C,P).
father(F,C) :- male(F), parent(F,C).
```

Along with the `parent` and `father` rules above, express the following familial relationships in a Prolog program. The bodies of your clauses should use as few terms as possible. Do not include any step-relations.

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

Part B

The song "I'm My Own Grandpa" (Latham/Jaffe) debuted in 1947. It was summarized by Niklaus Wirth (*Algorithms + Data Structures = Programs*, 1976) 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 stepdaughter 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 ($s2$).

[1] That is, at least those familial relationships that were thought of as "traditional" in one day and age; various countries, and provinces may have more modern definitions. To make your assignment easier, we'll stick to the traditional ones here.

Step I Step-relations should not have been reflected in your rules above. Add two new rules for `parent` that take these relations 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.

Step II There are many potential in-law relationss, 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 relation.

Step 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 facts using the same primitives be adjacent to each other.

Step IV Using Prolog, verify the conclusions drawn by the narrator:

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

Note: These should *not* go in your program, but would be queried interactively.

What to turn in

In addition to `references.txt`, your submission should include the following

- A PDF `exercises.pdf` of the logic sentences from Problem 1.
- Your completed `grandpa.pl` program file.
- A transcript `output.txt` of your program and results from Problem 2(B).IV.

How to Submit Your Prolog

Record your Prolog file and an interaction with the prolog interpreter to a PDF as follows:

1. To begin recording your session, type

   ```
   script output.txt
   ```

   Note that you may need to remove your username from the command line prompt before proceeding.
2. To echo your program to the transcript, type

   ```
   cat grandpa.pl
   ```

3. Start the Prolog interpreter, load your program, and use the interpreter to verify the conclusions given in Problem 2(B).IV.
4. Exit the Prolog interpreter by typing

   ```
   halt.
   ```

5. Stop the script session by typing `Ctrl-D`

Acknowledgments Problem 2 is adapted from P. Norvig, *Paradigms of Artificial Intelligence Programming*, Morgan-Kaufmann, 1992, Exercises 11.9 and 11.10.