CSc600 – Homework #2: Prolog [redacted]

---

1. For your family (or any other real or hypothetical family) write a PROLOG program "family.pro" that is based on the following facts:

- is_male(NAME). e.g. is_male(tom).
- is_female(NAME). e.g. is_female(ann).
- is_parent_of(PARENT,CHILD). e.g. is_parent_of(ann,tom).

Add to these facts the following inference rules:

- mother(MOTHER, CHILD)
- father(FATHER, CHILD)

- sibling1(NAME1, NAME2) (1 parent in common)
- brother1(NAME1, NAME2) (1 parent in common)
- sister1(NAME1, NAME2) (1 parent in common)

- sibling2(NAME1, NAME2) (2 parents in common)
- brother2(NAME1, NAME2) (2 parents in common)
- sister2(NAME1, NAME2) (2 parents in common)

- cousin(NAME1, NAME2)
- uncle(UNCLE, CHILDNAME)
- aunt(AUNT, CHILDNAME)
- grandparent(GRANDPARENT, GRANDCHILD)
- grandmother(GRANDMOTHER, GRANDCHILD)
- grandfather(GRANDFATHER, GRANDCHILD)
- grandchild(GRANDCHILD, GRANDPARENT)

- greatgrandparent(GREATGRANDPARENT, GREATGRANDCHILD)

- ancestor(ANCESTOR, CHILDNAME)

Show the results of your program for each of inference rules.

Note: In various cultures there are different interpretations of family relationships. All such interpretations are equally acceptable, and you may select any one of them.

2. Write a PROLOG program that investigates family relationships using lists. The facts should be organized as follows:

```
m([first_male_name, second_male_name, ..., last_male_name]).
f([first_female_name, second_female_name, ..., last_female_name]).
family( [father, mother, [child_1, child_2,..., child_n]] ).

e.g.  m([tom, joe, peter, john]).
      f([ann, cathy, jane, kim]).
      family([john, ann, [tom,kim]]).
      family([joe, jane, [ ]]).
```

Write rules that define the following relations:

male(X)
female(X)
father, mother, parent
siblings1, siblings2
brother1, brother2
sister1, sister2
cousins
uncle, aunt
grandchild, grandson, granddaughter
greatgrandparent
ancestor

For each of these rules show an example of its use.

3. Write a PROLOG program that includes the following operations with lists:

membership testing (is an element member of a list?)
first element
last element
two adjacent elements
three adjacent elements
append list1 to list2 producing list3
delete element from a list
append element to a list
insert element in a list
compute the length of list
reverse a list
check whether a list is a palindrome
display a list

For each of these operations write your implementation of the operation and show an example of its use. If a predicate already exists (predefined in Prolog), modify its name (e.g. myappend or append1). Lists to be processed can be created by an auxiliary program, defined as facts, or entered from the keyboard.

4. Write a PROLOG program that solves the 8 queens problem (location of 8 queens on a chess board so that no queens have each other in check, i.e. are not located in the same row/column/diagonal).

A solution of this problem is presented in Chapter 22 of Webber's textbook which has slides posted on iLearn.

Notes:
(1) Each program must have examples of its output.
(2) [redacted]

[redacted]

[redacted]