Assignment #10

CSC301 – -- **Due Thursday, 11/30, in Sakai**

In order to complete these exercises you will need to install prolog from the CSC301 Prolog webpage.

Write a Prolog program by creating a file with you favorite **program** editor that contains the following facts:

- here the predicate parent(X,Y) means X is the parent of Y
- you can also download the following database from the course website

```
female(pam).
female(liz).
female(ann).
female(pat).
male(tom).
male(bob).
male(jim).
parent(pam,bob).
parent(tom,bob).
parent(tom,liz).
parent(bob,ann).
parent(bob,pat).
parent(pat,jim).
```

(a) Load this file into Prolog, usually this is done with the *consult file predicate*:

?- consult('<filename>').

On Windows you can load the fact database with the menu point File→Consult. More hints on how to load and run Prolog programs on the course Prolog website. Once you have loaded the program pose the following queries:

?- female(ann).

?- female(jim).

?- parent(X,bob).

?- parent(tom,X).

?- parent(X,ann),parent(X,pat).

What are the answers to these queries? Beware, for some queries here might be more than one answer. To get all the answers type a ';' and carriage return at the question mark.

(b) Now, using the parent predicate formulate the following Prolog queries:

1. Who is Pat's parent?
2. Does Liz have a child?

3. Who is Pat's grandparent?

(c) Given the above facts, extend the program by writing rules defining the following predicates:

sister(X,Y) -- X is the sister of Y.

son(X,Y) -- X is the son of Y.

father(X,Y) -- X is the father of Y.

grandmother(X,Y) -- X is the grandmother of Y.

ancestor(X,Y) -- X is an ancestor of Y.

(Hint: this predicate might come in handy: different(X,Y):- not(X=Y). Some predicate definitions might be **recursive**.)

Demonstrate that your program works by posing the following queries:

4. ?- sister(X,pat).
5. ?- sister(X,Y).
6. ?- son(jim,X).
7. ?- father(X,bob).
8. ?- grandmother(X,ann).
9. ?- ancestor(X,jim).

Hand in the source code of your prolog program and a proof of the program execution.