CISC 404/604 homework assignment 11.

1. (10 points) Put the following axioms into a file and load them into BinProlog.

parent(dad(X),X).
parent(mom(X),X).
grandparent(X,Y) :- parent(X,Z),parent(Z,Y).

Now ask the queries (one at a time).

?- grandparent(X,jim).

?- grandParent(X,Y).

How many answers can you get to these queries? What are they? Make a script file showing BinProlog finding the answers and turn it in.

2. (10 points) Define a procedure (predicate) intersect(X,Y,Z) such that if X and Y are lists, Z will be a list of the items that appear on both lists. Thus, if you make the query

?- intersect([3,6,2,5],[1,9,4,5,6],X).

It will respond with X = [6, 5]. This can be done with one fact and two rules. You will need to use the built-in function member(X,Y) in one of the rules, where member(X,Y) means that X is a member of the list Y. Make a script file in which you demonstrate your program on three pairs of lists of numbers and turn it in.

3. (10 points) Write a Prolog program isort(N,X,Y) which succeeds when N is a number, X is a list of numbers in increasing order, and Y is the list X with N inserted in it in the proper place so that the numbers in Y are also in increasing order. A number may appear in the sorted list more than once. Thus, after loading your definition of isort into BinProlog, you should get this behavior:

?- isort(7,[2,4,7,7,9],X).
X = [2,4,7,7,7,9]
?- isort(2,[-3,0,1,1,4,5],X).
X = [-3, 0, 1, 1, 2, 4, 5]
?- isort(3,X,[1,2,3,4]).
X = [1,2,4]

This time you will need two facts and one rule. The rule will need to use the cut operator. The order of the facts and rule is

important Listing both facts before the rule will not work. Make a script file in which you demonstrate your program on examples like the above and turn it in.

4. (10 points) The Tower of Hanoi is a classic problem in computer science. There are three piles. Two of the piles are initially empty. One pile is a tower of discs, all of different size and stacked so each is smaller than the disk it is sitting on. The problem is to move the discs one at a time between the piles so that at all times, the discs in each pile are stacked according to size, and at the end, the initial pile has been moved to one of the other piles, which has been designated in advance. To solve this problem , you will write a couple of rules and/or facts for a predicate transfer(N,X,Y,Z), which means "transfer a pile of N discs from pile X to pile Z, using Y for temporary storage if needed." The basic idea of the solution is this: to move the top N discs from pile X to pile Z, you must first move the top N-1 discs from X to Y, then move the top disc of pile X to pile Z, then move the top N-1 discs from Y to Z. One of the rules or facts will deal with the special case of moving just one disc from one pile to another pile, or the special case of moving no disc from one pile to another pile. The other rule will use the "X is ..." expression that Prolog uses for evaluation of arithmetic expressions. The rule will also need to call the predicate move disc(X,Y) which is defined by the rule

move disc(X,Y) :- write([move, top, disc, from, X, to, Y]), nl.

Make a script in which you demonstrate your program by making the query transfer(3,a,b,c) and turn it in.

5. (20 points) Modify the Prolog program in project2.pl (in ~chester/prolog on Strauss) so that it finds the shortest path between two cities. You will probably need to use the asserta predicate.