Shenliang Wang

10/15/2017

CSC 600

SECTION #1

HW#2

Prolog#2

**Problem 1:family2.pl**

%FACTS

m([tatay, alex, edison, aj, adrian, michael, angelo, matt, lucas]).

f([nanay, emilie, nerissa, susanna, lerinna, melissa, edlin]). %Families

family([alex,emilie, [aj, susanna, lerinna, adrian]]). family([tatay, nanay, [alex, nerissa]]).

family([edison, nerissa, [ michael, melissa]]).

family([aj, edlin, [angelo, matt, lucas]]).

%RULES

male(X) :­ m(Male), member(X, Male). female(X) :­ f(Female), member(X, Female).

mother(Mother, Child) :­ family([\_, Mother, Y]), member(Child, Y).

father(Father, Child) :­ family([Father, \_, Y]), member(Child, Y).

parent(Parent, Child) :­ mother(Parent, Child). parent(Parent, Child) :­ father(Parent, Child).

%1 parent in commom.

siblings1(Child1, Child2) :­ mother(Mother, Child1), mother(Mother, Child2), father(X, Child1),

father(Y, Child2), X \= Y, Child1 \= Child2. siblings1(Child1, Child2) :­ father(Father, Child1), father(Father, Child2), mother(X,Child1), mother(Y, Child2), X\=Y, Child1\=Child2.

brother1(Child1, Child2) :­ male(Child2), siblings1(Child1, Child2).

sister1(Child1, Child2) :­ female(Child2), siblings1(Child1, Child2).

%2 parents in common.

siblings2(Child1, Child2) :­ mother(Mother, Child1), mother(Mother, Child2), father(Y, Child1), father(Y, Child2), Child1 \= Child2.

brother2(Child1, Child2) :­ male(Child2), siblings2(Child1, Child2).

sister2(Child1, Child2) :­ female(Child2), siblings2(Child1, Child2).

cousin(Name1, Name2) :­ parent(X, Name1), parent(Y, Name2), siblings2(X, Y), X \= Y.

uncle(Uncle, Child) :­ parent(X, Child), brother2(X, Uncle). aunt(Aunt, Child) :­ parent(X, Child), sister2(X, Aunt).

grandparent(GrandParent, Child) :­ parent(X, Child), parent(GrandParent, X).

grandmother(GrandMother, Child) :­ parent(X, Child), mother(GrandMother, X).

grandfather(Grandfather, Child) :­ parent(X, Child), father(Grandfather, X).

grandchild(Child, GrandParent) :­ grandparent(GrandParent, Child).

greatgrandparent(GreatGranParent, GreatChild) :­ parent(X, GreatChild), grandparent(GreatGranParent, X).

grandson(Child, Grandparent) :­ male(Child), grandchild(Child, Grandparent).

granddaughter(Child, Grandparent) :­ female(Child), grandchild(Child, Grandparent).

ancestor(Ancestor, Child) :­ parent(Ancestor, Child). ancestor(Ancestor, Child) :­ parent(Parent, Child), ancestor(Ancestor, Parent).

OUTPUT:

?­ male(X).

X = tatay ;

X = alex ;

X = edison ;

X = aj;

X = adrian ;

X = michael.

?­ female(X).

X = nanay ;

X = emilie ;

X = nerissa ;

X = susanna ;

X = lerinna ;

X = melissa ;

X = edlin.

?­ father(X,Y).

X = alex,

Y = aj;

X = alex,

Y = susanna ;

X = alex,

Y = lerinna ;

X = alex,

Y = adrian ;

X = tatay,

Y = alex ;

X = tatay,

Y = nerissa ;

X = edison,

Y = michael.

?­ mother(X,Y).

X = emilie,

Y = aj;

X = emilie,

Y = susanna ;

X = emilie,

Y = lerinna .

?­ siblings1(X,Y).

false.

?­ siblings2(X,Y).

X = aj,

Y = susanna ;

X = aj,

Y = lerinna ;

X = aj,

Y = adrian ;

X = susanna,

Y = aj;

X = susanna,

Y = lerinna .

?­ brothers1(adrian, alex).

Correct to: "brother1(adrian,alex)"? yes

false.

?­ brother2(adrian, alex).

false.

?­ brother2(adrian,aj).

true

?­ sister1(susanna,lerinna).

false.

?­ sister2(susanna,lerinna).

true

?­ cousins(X,Y).

Correct to: "cousin(X,Y)"? yes

X = michael,

Y = aj;

X = michael,

Y = susanna ;

X = michael,

Y = lerinna .

?­ uncle(X,Y).

X = alex,

Y = michael ;

X = alex,

Y = melissa ;

X = adrian,

Y = angelo .

?­ aunt(X,Y).

X = nerissa,

Y = aj;

X = nerissa,

Y = susanna ;

X = nerissa,

Y = lerinna.

?­ grandchild(X,Y).

X = michael,

Y = nanay ;

X = michael,

Y = tatay .

?­ grandson(X,Y).

X = aj,

Y = nanay ;

X = aj,

Y = tatay .

?­ granddaughter(X,Y).

X = susanna,

Y = nanay .

?­ greatgrandparent(X,Y).

X = nanay,

Y = angelo ;

X = tatay,

Y = angelo .

?­ ancestor(X,Y).

X = emilie,

Y = aj;

X = emilie,

Y = susanna .

**Problem 2: listProcessing.pl**

%RULES

is\_member(X, [X|T]).

is\_member(X, [\_|T]) :­ is\_member(X, T).

first\_element(X, [X|\_]).

last\_element(X, [X]).

last\_element(X, [\_|T]) :­ last\_element(X, T). two\_adjacent\_element(X,Y,[X,Y|\_]). two\_adjacent\_element(X,Y,[\_|T]) :­ adjacent\_element(X,Y,T). three\_adjacent\_element(X,Y,Z,L) :­ two\_adjacent\_element(X,Y,L), two\_adjacent\_element(Y,Z,L).

append\_list([], L2, L2).

append\_list([H|T], L2, [H|L3]) :­ append\_list(T,L2,L3). append\_element(L,E, NL) :­ append\_list(L, [E], NL).

delete\_element(X,[X|L], L).

delete\_element(X, [H|T], [H|L]) :­ delete\_element(X, T, L). insert\_element(X, L, NL) :­ delete\_element(X, NL, L).

length1([], 0).

length1([X|Y], Length) :­ length1(Y,Length1), Length is Length1+1.

reverse\_list([],[]).

reverse\_list([H|T], L) :­ reverse\_list(T, RL), append(RL, [H], L).

palindrome(L) :­ reverse\_list(L, R), R==L.

display\_list([]) :­ nl.

display\_list([H|T]) :­ write(H), nl, display\_list(T).

OUTPUT:

?­ member(1, [1,2,3,4]).

true .

?­ member(X, [1,2,3,4]).

X=1;

X=2;

X=3;

X=4.

?­ first\_element(X,[1,2,3,4]).

X=1.

?­ last\_element(X,[2,3,6]).

X=6

?­ two\_adjacent\_elements(X,Y,[1,2,3,4,5,6]).

Correct to: "two\_adjacent\_element(X,Y,[1,2,3,4,5,6])"? yes

X=1,

Y=2;

X=2,

Y=3;

X=3,

Y=4;

X=4,

Y=5;

X=5,

Y=6.

?­ three\_adjacent\_element(X,Y,Z,[1,2,3,4]).

X=1,

Y=2,

Z=3;

X=2, Y=3, Z=4; false.

?­ append\_list([1],[2,3],X).

X = [1, 2, 3].

?­ append\_element([1,2,3,4],5,X).

X = [1, 2, 3, 4, 5].

?­ insert\_element(1,[2,3,4], X).

X = [1, 2, 3, 4] .

?­ delete\_element(1, [2,1,3], X).

X = [2, 3] ;

false.

?­ length1([1,2,3,4],X).

X=4.

?­ reverse\_list([1,2,3,4],X).

X = [4, 3, 2, 1].

? palindrome([1,2,3,2,1]).

true.

?­ palindrome([1,2,3]).

false.

?­ display\_list([1,2,3,4,5]).

1

2

3

4

5

true.

**Problem 3: 8queens.pl**

%RULES

no\_check(\_, []).

no\_check(X/Y, [X1/Y1 | Rest]) :­Y =\= Y1, abs(Y1­Y) =\= abs(X1­X),

no\_check(X/Y, Rest).

legal([]).

legal([X/Y | Rest]) :­ legal(Rest), member(Y, [1,2,3,4,5,6,7,8]), no\_check(X/Y, Rest).

eightqueens(X) :­ X = [1/\_, 2/\_, 3/\_, 4/\_, 5/\_, 6/\_, 7/\_, 8/\_], legal(X).

OUTPUT:

?­ eightqueens(X).

X = [1/4, 2/2, 3/7, 4/3, 5/6, 6/8, 7/5, 8/1] ;

X = [1/5, 2/2, 3/4, 4/7, 5/3, 6/8, 7/6, 8/1] ;

X = [1/3, 2/5, 3/2, 4/8, 5/6, 6/4, 7/7, 8/1] ;

X = [1/3, 2/6, 3/4, 4/2, 5/8, 6/5, 7/7, 8/1] ;

X = [1/5, 2/7, 3/1, 4/3, 5/8, 6/6, 7/4, 8/2] ;

X = [1/4, 2/6, 3/8, 4/3, 5/1, 6/7, 7/5, 8/2] ;

X = [1/3, 2/6, 3/8, 4/1, 5/4, 6/7, 7/5, 8/2] ;

X = [1/5, 2/3, 3/8, 4/4, 5/7, 6/1, 7/6, 8/2] ;

X = [1/5, 2/7, 3/4, 4/1, 5/3, 6/8, 7/6, 8/2] ;

X = [1/4, 2/1, 3/5, 4/8, 5/6, 6/3, 7/7, 8/2] ;

X = [1/3, 2/6, 3/4, 4/1, 5/8, 6/5, 7/7, 8/2] ;

X = [1/4, 2/7, 3/5, 4/3, 5/1, 6/6, 7/8, 8/2] ;

X = [1/6, 2/4, 3/2, 4/8, 5/5, 6/7, 7/1, 8/3] ;

X = [1/6, 2/4, 3/7, 4/1, 5/8, 6/2, 7/5, 8/3] ;

X = [1/1, 2/7, 3/4, 4/6, 5/8, 6/2, 7/5, 8/3] ;

X = [1/6, 2/8, 3/2, 4/4, 5/1, 6/7, 7/5, 8/3] ;

X = [1/6, 2/2, 3/7, 4/1, 5/4, 6/8, 7/5, 8/3] ;

X = [1/4, 2/7, 3/1, 4/8, 5/5, 6/2, 7/6, 8/3] ;

X = [1/5, 2/8, 3/4, 4/1, 5/7, 6/2, 7/6, 8/3] ;

X = [1/4, 2/8, 3/1, 4/5, 5/7, 6/2, 7/6, 8/3] ;

X = [1/2, 2/7, 3/5, 4/8, 5/1, 6/4, 7/6, 8/3] ;

X = [1/1, 2/7, 3/5, 4/8, 5/2, 6/4, 7/6, 8/3] ;

X = [1/2, 2/5, 3/7, 4/4, 5/1, 6/8, 7/6, 8/3] ;

X = [1/4, 2/2, 3/7, 4/5, 5/1, 6/8, 7/6, 8/3] .