Jonathan Julian

Hw 1.4

#1

RULES:

is\_male(jonathan).

is\_male(jayden).

is\_male(bryan).

is\_male(jon).

is\_male(joe).

is\_male(joseph).

is\_male(braden).

is\_male(jacob).

is\_male(phillip).

is\_male(garret).

is\_male(natale).

is\_female(mary).

is\_female(samantha).

is\_female(onna).

is\_female(christina).

is\_female(kathy).

is\_female(judy).

is\_female(jodi).

is\_female(makayla).

is\_parent\_of(jon, jonathan).

is\_parent\_of(natale,joe).

is\_parent\_of(jon, makayla).

is\_parent\_of(christina, jonathan).

is\_parent\_of(christina, makayla).

is\_parent\_of(joe, jon).

is\_parent\_of(judy, jon).

is\_parent\_of(judy, joseph).

is\_parent\_of(joe, joseph).

is\_parent\_of(joseph, braden).

is\_parent\_of(joseph, jacob).

is\_parent\_of(joe, mary).

is\_parent\_of(judy, mary).

is\_parent\_of(mary, garret).

is\_parent\_of(mary, phillip).

is\_parent\_of(kathy, samantha).

is\_parent\_of(bryan, samantha).

is\_parent\_of(samantha, jayden).

is\_parent\_of(samantha, onna).

is\_parent\_of(jonathan, onna).

is\_parent\_of(jonathan, jayden).

mother(Mother, Child) :- is\_female(Mother) , is\_parent\_of(Mother, Child).

father(Father, Child) :- is\_male(Father), is\_parent\_of(Father, Child).

sibling1(Name1, Name2) :- is\_parent\_of(Parent, Name1), is\_parent\_of(Parent, Name2), Name1\=Name2.

brother1(Name1, Name2) :- is\_male(Name1), is\_parent\_of(Parent, Name1),is\_parent\_of(Parent, Name2), Name1\=Name2.

sister1(Name1, Name2) :- is\_female(Name1), is\_parent\_of(Parent, Name1),is\_parent\_of(Parent, Name2), Name1\=Name2.

sibling2(Name1, Name2) :- is\_parent\_of(Parent1, Name1), is\_parent\_of(Parent1, Name2), is\_parent\_of(Parent2, Name1), is\_parent\_of(Parent2, Name2), Parent2\=Parent1, Name1\=Name2.

brother2(Name1, Name2) :- is\_male(Name1),is\_parent\_of(Parent1, Name1), is\_parent\_of(Parent1, Name2), is\_parent\_of(Parent2, Name1), is\_parent\_of(Parent2, Name2), Parent2\=Parent1, Name1\=Name2.

sister2(Name1, Name2) :-is\_female(Name1), is\_parent\_of(Parent1, Name1), is\_parent\_of(Parent1, Name2), is\_parent\_of(Parent2, Name1), is\_parent\_of(Parent2, Name2), Parent2\=Parent1, Name1\=Name2.

cousin(Name1, Name2) :- is\_parent\_of(Grandparent1, Parent1), is\_parent\_of(Grandparent1, Parent2), is\_parent\_of(Parent1, Name1), is\_parent\_of(Parent2, Name2), Parent1\=Parent2, Name1\=Name2.

uncle(Uncle, N) :- is\_male(Uncle), is\_parent\_of(Grandparent, Uncle), is\_parent\_of(Grandparent, Parent), is\_parent\_of(Parent, N), Parent\=Uncle.

aunt(Aunt, N) :- is\_female(Aunt), is\_parent\_of(Grandparent, Aunt), is\_parent\_of(Grandparent, Parent), is\_parent\_of(Parent, N), Parent\=Aunt.

grandparent(Grandparent, Grandchild) :- is\_parent\_of(Grandparent, Parent), is\_parent\_of(Parent, Grandchild).

grandfather(Grandfather, Grandchild) :-is\_male(Grandfather),is\_parent\_of(Grandfather, Parent), is\_parent\_of(Parent, Grandchild).

grandmother(Grandmother, Grandchild) :-is\_female(Grandmother),is\_parent\_of(Grandmother, Parent), is\_parent\_of(Parent, Grandchild).

grandchild(Grandchild, Grandparent) :- is\_parent\_of(Parent, Grandchild), is\_parent\_of(Grandparent, Parent).

greatgrandparent(Greatgrandparent, Greatgrandchild):- grandparent(Greatgrandparent, Parent), is\_parent\_of(Parent, Greatgrandchild).

ancestor(Ancestor, Child) :- is\_parent\_of(Ancestor, Child).

ancestor(Ancestor, Child) :- is\_parent\_of(Ancestor, Child1), ancestor(Child1, Child).

OUTPUT:

?- mother(X,Y).

X = mary,

Y = garret .

?- father(X,Y).

X = jonathan,

Y = onna .

?- sibling1(X,Y).

X = jonathan,

Y = makayla .

?- brother1(X,Y).

X = jonathan,

Y = makayla .

?- sister1(X,Y).

X = mary,

Y = jon .

?- sibling2(X,Y).

X = jonathan,

Y = makayla .

?- brother2(X,Y).

X = jonathan,

Y = makayla .

?- sister2(X,Y).

X = mary,

Y = jon .

?- cousin(X,Y).

X = jonathan,

Y = braden .

?- uncle(X,Y).

X = jon,

Y = braden .

?- aunt(X,Y).

X = mary,

Y = jonathan .

?- grandparent(X,Y).

X = jon,

Y = onna .

?- grandmother(X,Y).

X = christina,

Y = onna .

?- grandfather(X,Y).

X = bryan,

Y = jayden .

?- grandchild(X,Y).

X = jonathan,

Y = joe .

?- greatgrandparent(X,Y).

X = natale,

Y = jonathan .

?- ancestor(X,Y).

X = jon,

Y = jonathan ;

X = natale,

Y = joe ;

X = jon,

Y = makayla ;

X = christina,

Y = jonathan ;

X = christina,

Y = makayla ;

X = joe,

Y = jon ;

X = judy,

Y = jon ;

X = judy,

Y = joseph ;

X = joe,

Y = joseph ;

X = joseph,

Y = braden ;

X = joseph,

Y = jacob ;

X = joe,

Y = mary ;

X = judy,

Y = mary ;

X = mary,

Y = garret ;

X = mary,

Y = phillip ;

X = kathy,

Y = samantha ;

X = bryan,

Y = samantha ;

X = samantha,

Y = jayden ;

X = samantha,

Y = onna ;

X = jonathan,

Y = onna ;

X = jonathan,

Y = jayden ;

X = jon,

Y = onna ;

X = jon,

Y = jayden ;

X = natale,

Y = jon ;

X = natale,

Y = joseph ;

X = natale,

Y = mary ;

X = natale,

Y = jonathan ;

X = natale,

Y = makayla ;

X = natale,

Y = onna ;

X = natale,

Y = jayden ;

X = natale,

Y = braden ;

X = natale,

Y = jacob ;

X = natale,

Y = garret ;

X = natale,

Y = phillip ;

X = christina,

Y = onna ;

X = christina,

Y = jayden ;

X = joe,

Y = jonathan ;

X = joe,

Y = makayla ;

X = joe,

Y = onna ;

X = joe,

Y = jayden ;

X = judy,

Y = jonathan .

#2

Facts/rules:

teaches(ta,csc210-03).

teaches(bierman,csc210-05).

teaches(humayoun,csc210-06).

teaches(ta,csc220-03).

teaches(song, csc220-05).

teaches(tomasevich, csc300-02).

teaches(yang, csc340-01).

teaches(souza, csc413-02).

teaches(song, csc648-03).

teaches(wall, csc510-02).

teaches(wall, csc510-01).

teaches(ta, csc340-03).

freshman(csc210-05).

freshman(csc220-05).

freshman(csc210-03).

freshman(csc210-06).

freshman(csc220-03).

sophomore(csc300-02).

sophomore(csc340-01).

sophomore(csc340-03).

junior(csc413-02).

junior(csc510-02).

junior(csc510-01).

senior(csc648-03).

morning(csc340-03).

morning(csc300-02).

morning(csc220-03).

morning(csc220-05).

morning(csc340-01).

afternoon(csc210-03).

afternoon(csc210-05).

afternoon(csc210-06).

afternoon(csc510-01).

afternoon(csc648-03).

afternoon(csc413-02).

afternoon(csc510-02).

afternoon(csc510-01).

tth(csc210-03).

tth(csc210-05).

tth(csc210-06).

tth(csc300-02).

tth(csc340-01).

tth(csc220-05).

tth(csc220-03).

tth(csc413-02).

tth(csc648-03).

mwf(csc340-03).

mwf(csc510-01).

mwf(csc510-02).

%RULES

teaches\_freshman(X) :- freshman(Y), teaches(X,Y).

teaches\_sophomore(X) :- sophomore(Y), teaches(X,Y).

teaches\_junior(X) :- junior(Y), teaches(X,Y).

teaches\_senior(X) :- senior(Y), teaches(X,Y).

teaches\_morning(X) :- morning(Y), teaches(X,Y).

teaches\_afternoon(X) :- afternoon(Y), teaches(X,Y).

teaches\_whole\_day(X) :- morning(Y1), teaches(X,Y1), afternoon(Y2), teaches(X,Y2).

works\_mwf(X) :- mwf(Y), teaches(X,Y).

works\_tth(X) :- tth(Y), teaches(X,Y).

teaches\_three\_classes(X):- teaches(X,Y1), teaches(X,Y2), teaches(X, Y3), Y3\=Y2, Y2\=Y1, Y3\=Y1.

OUTPUT:

?- teaches\_three\_classes(X).

X = ta ;

X = ta ;

X = ta ;

X = ta ;

X = ta ;

X = ta ;

false.

?- works\_tth(X).

X = ta ;

X = bierman ;

X = humayoun ;

X = tomasevich ;

X = yang ;

X = song ;

X = ta ;

X = souza ;

X = song.

?- works\_mwf(X).

X = ta ;

X = wall ;

X = wall ;

false.

?- teaches\_freshman(X).

X = bierman ;

X = song ;

X = ta ;

X = humayoun ;

X = ta ;

false.

?- teaches\_sophomore(X).

X = tomasevich ;

X = yang ;

X = ta.

?- teaches\_junior(X).

X = souza ;

X = wall ;

X = wall.

?- teaches\_senior(X).

X = song.

?- teaches\_morning(X).

X = ta ;

X = tomasevich ;

X = ta ;

X = song ;

X = yang ;

false.

?- teaches\_afternoon(X).

X = ta ;

X = bierman ;

X = humayoun ;

X = wall ;

X = song ;

X = souza ;

X = wall ;

X = wall.

?- teaches\_whole\_day(X).

X = ta ;

X = ta ;

X = song ;

false.