```
:- [lab7lmc2].

%%%%%%%%% Exercitiul 1 / prima lista de subiecte:

l2xl3(A,OrdA) :- A=[0,a,b,c,d,1],
        orddinsucc([(0,a),(0,b),(a,c),(b,c),(b,d),(c,1),(d,1)],A,OrdA).

l2plusl2xl2(A,OrdA) :- A=[0,u,v,w,1],
        orddinsucc([(0,u),(u,v),(u,w),(v,1),(w,1)],A,OrdA).

functiistrdescresc(P,OrdP,Q,OrdQ,LF) :- ordstrdinord(OrdP,OrdStrP),
        ordstrdinord(OrdQ,OrdStrQ), invrel(OrdStrQ,InvOrdStrQ),
        setof(F, (functie(F,P,Q), pastrrel(F,OrdStrP,InvOrdStrQ)), LF), !.
functiistrdescresc(_,_,_,_,[]).

fctL2xL3laL2plusL2xL2(LF) :- l2xl3(P,OrdP), l2plusl2xl2(Q,OrdQ),
        functiistrdescresc(P,OrdP,Q,OrdQ,LF).

% varianta 1 pentru al doilea predicat:

functiineinj([]).
functiineinj([F|LF]) :- /* write(F), nl,*/ not(injectiv(F)), functiineinj(LF).

niciunainj :- fctL2xL3laL2plusL2xL2(LF), functiineinj(LF).

% varianta 2 pentru al doilea predicat:
```

```
functiineinjective(LF) :- not((member(F,LF), /* write(F), nl, */ injectiv(F))).

niciunainjectiva :- fctL2xL3laL2plusL2xL2(LF), functiineinjective(LF).

%%%%%%%%% Exercitiul 2 / prima lista de subiecte:

/* NU MAI SCRETI URMATOARELE DACA LE-ATI SCRIS IN REZOLVAREA MATEMATICA:
Fie h:V->L2 a.i. h |= Sigma U Delta. <=> h|=Sigma si h|=Delta.
Fie f : L2={0,1} -> {false,true} izomorfismul boolean: f(0)=false, f(1)=true.
Conform TCT:
Sigma |- alfa V (beta -> gama) <=> Sigma |= alfa V (beta -> gama) =>
        h |= alfa V (beta -> gama) <=> h~(alfa V (beta -> gama)) = 1
        <=> f(h~(alfa V (beta -> gama))) = true
        <=> f(h~(alfa)) V [f(h~(beta)) -> f(h~(gama))] = 1;
Delta |- gama -> alfa <=> Delta |= gama -> alfa => h |= gama -> alfa <=>
        h~(gama -> alfa) = 1 <=> f(h~(gama -> alfa)) = true <=>
        f(h~(gama)) -> f(h~(alfa)) = true.
Avem de demonstrat ca Sigma U Delta |- beta -> alfa, i.e., conform TCT:
Sigma U Delta |= beta -> alfa, adica e suficient sa demonstram ca:
h |= beta -> alfa. <=> h~(beta -> alfa) = 1 <=> f(h~(beta -> alfa)) = true
<=> f(h~(beta)) -> f(h~(alfa)) = true.
SCRIETI DOAR ACEST LUCRU: Consideram variabilele Prolog:
Alfa = f(h~(alfa)), Beta = f(h~(beta)), Gama = f(h~(gama)).
FACULTATIV DE SCRIS: Atunci urmatoarele predicate returneaza valorile booleene:
ipoteza1(Alfa,Beta,Gama) = f(h~(alfa)) V [f(h~(beta)) -> f(h~(gama))]
ipoteza2(Alfa,Gama) = f(h~(gama)) -> f(h~(alfa))
concluzia(Alfa,Beta) = f(h~(beta)) -> f(h~(alfa))
```

```
*/

ipoteza1(Alfa,Beta,Gama) :- Alfa ; implica(Beta,Gama).

ipoteza2(Alfa,Gama) :- implica(Gama,Alfa).

concluzia(Alfa,Beta) :- implica(Beta,Alfa).

regded :- not((listaValBool([Alfa,Beta,Gama]), ipoteza1(Alfa,Beta,Gama),
        ipoteza2(Alfa,Gama), not(concluzia(Alfa,Beta)))).

%%%%%%%%% Exercitiul 3 / prima lista de subiecte:

multA([a,b,c,d]).

succA([(a,b),(b,d)]).

posetA(A,OrdA) :- multA(A), succA(SuccA),
        orddinsucc(SuccA,A,OrdA).

detR(R) :- posetA(_,OrdA), invrel(OrdA,R).

fctbij(F,A,B) :- permutare(B,P), constrbij(F,A,P).

constrbij([],[],[]).
constrbij([(H,K)|F],[H|T],[K|U]) :- constrbij(F,T,U).
```

```
izomposeturi(F,P,OrdP,Q,OrdQ) :- fctbij(F,P,Q), pastrrel(F,OrdP,OrdQ),
        invrel(F,G), pastrrel(G,OrdQ,OrdP).

detF(F) :- posetA(A,OrdA), invrel(OrdA,R), izomposeturi(F,A,OrdA,A,R).

verifAsatepsilon :- multA(A), detF(F), detR(R),
        not((member(X,A), member(Y,A), /* write((X,Y)), nl, */
        member((X,FX),F), member((Y,FY),F), member((FY,FFY),F),
        not(implica(member((X,FY),R), not(member((FX,FFY),R))))))).

/* A |= (oricare x)(oricare y)(p(x,y)) <=>
A |= non[(exista x)(exista y)(non p(x,y))]
*/

% Facultativ, verificari:

verifAsatisfepsilon :- multA(A), detF(F), detR(R),
        not((member(X,A), member(Y,A),
        member((X,FX),F), member((Y,FY),F), member((FY,FFY),F),
        not((implica(member((X,FY),R), not(member((FX,FFY),R))),
        write((X,Y)), nl)))).

%%%

afisperechisatimplic :- multA(A), detF(F), detR(R),
        not((member(X,A), member(Y,A),
        member((X,FX),F), member((Y,FY),F), member((FY,FFY),F),
```

```
        not(afisperechesatimplic(X,Y,R,FX,FY,FFY))).

afisperechesatimplic(X,Y,R,FX,FY,FFY) :-
   implica(member((X,FY),R), not(member((FX,FFY),R))), !, write((X,Y)), nl.
afisperechesatimplic(_,_,_,_,_,_).

%%%

afisperechinusatimplic :- multA(A), detF(F), detR(R),
        not((member(X,A), member(Y,A),
        member((X,FX),F), member((Y,FY),F), member((FY,FFY),F),
        not(afisperechenusatimplic(X,Y,R,FX,FY,FFY)))).

afisperechenusatimplic(X,_,R,FX,FY,FFY) :-
   implica(member((X,FY),R), not(member((FX,FFY),R))), !.
afisperechenusatimplic(X,Y,_,_,_,_) :- write((X,Y)), nl.

%%%

afisperechi :- multA(A), detF(F), detR(R),
        not((member(X,A), member(Y,A),
        member((X,FX),F), member((Y,FY),F), member((FY,FFY),F),
        not(afispereche(X,Y,R,FX,FY,FFY)))).

afispereche(X,Y,R,FX,FY,FFY) :-
   implica(member((X,FY),R), not(member((FX,FFY),R))), !,
        write((X,Y)), write(' satisface implicatia'), nl.
```

```
afispereche(X,Y,_,_,_,_) :-
        write((X,Y)), write(' nu satisface implicatia'), nl.

%%%%%%%%% Exercitiul 4 / a doua lista de subiecte:

l2xl2plusl2(A,OrdA) :- A=[0,v,w,u,1],
        orddinsucc([(0,v),(0,w),(v,u),(w,u),(u,1)],A,OrdA).

functiistrcresc(P,OrdP,Q,OrdQ,LF) :-
        ordstrdinord(OrdP,OrdStrP), ordstrdinord(OrdQ,OrdStrQ),
        setof(F, (functie(F,P,Q), pastrrel(F,OrdStrP,OrdStrQ)), LF), !.
functiistrcresc(_,_,_,_,[]).

fctL2xL3laL2xL2plusL2(LF) :- l2xl3(P,OrdP), l2xl2plusl2(Q,OrdQ),
        functiistrcresc(P,OrdP,Q,OrdQ,LF).

toatepastrsucc :- fctL2xL3laL2xL2plusL2(LF),
        l2xl3(_,OrdP), l2xl2plusl2(_,OrdQ),
        succdinord(OrdP,SuccP), succdinord(OrdQ,SuccQ),
        toatepastrrel(LF,SuccP,SuccQ).

toatepastrrel(LF,SuccP,SuccQ) :- not((member(F,LF), /* write(F), nl, */
                not(pastrrel(F,SuccP,SuccQ)))).

%%%%%%%%% Exercitiul 5 / a doua lista de subiecte:

% Cu aceleasi notatii ca in Exercitiul 1 / prima lista de subiecte:
```

```
ip1(Alfa,Beta,Gama) :- implica(Alfa, Beta;Gama).

ip2(Alfa,Beta) :- implica(Beta, not(Alfa)).

ip3(Alfa,Beta,Gama) :- implica(not(Alfa), (not(Beta),Gama)).

cl(Beta,Gama) :- not(Beta),Gama.

ded :- not((listaValBool([Alfa,Beta,Gama]), ip1(Alfa,Beta,Gama),
        ip2(Alfa,Beta), ip3(Alfa,Beta,Gama), not(cl(Beta,Gama)))).

%%%%%%%%% Exercitiul 6 / a doua lista de subiecte:

multimeaA([a,b,c,d]).

succesiuneaA([(a,b),(c,d)]).

posetulA(A,OrdA) :- multimeaA(A), succesiuneaA(SuccA),
        orddinsucc(SuccA,A,OrdA).

detf(F) :- posetulA(A,OrdA), invrel(OrdA,InvOrdA),
        izomposeturi(F,A,OrdA,A,InvOrdA),
        member((a,Fa),F), member((b,Fb),F),
        not(member(Fa,[a,b])), not(member(Fb,[a,b])).

detrelR(R) :- multimeaA(A), succesiuneaA(SuccA), eqgen(SuccA,A,R).
```

```
verificAsatepsilon :- multimeaA(A), detf(F), detrelR(R),
        not((member(X,A), not((member(Y,A),
        member((X,FX),F), member((Y,FY),F), member((FY,FFY),F),
        implica(member((X,Y),R), member((FX,FFY),R))))))).

/* A |= (oricare x)(exista y)(p(x,y)) <=>
A |= non[(exista x)[non[(exista y)(p(x,y))]]
*/
```