

```
:- [lab5lmc4].
```

```
/* Notez suma ordinala cu + si ridicarea la putere cu ^. Sa construim cateva poseturi  
(P,OrdP), introducand multimea suport P si relatia de succesiune, apoi obtinand din acestea  
relatia de ordine OrdP cu predicatul orddinsucc: */
```

```
% A = L2^2+L2: suma ordinala dintre romb si lantul cu doua elemente:
```

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

```
% B = "V rasturnat":
```

```
posetB(B,OrdB) :- B=[u,v,1], orddinsucc([(u,1),(v,1)],B,OrdB).
```

```
% L2: lantul cu doua elemente:
```

```
l2([0,1],OrdL2) :- orddinsucc([(0,1)],[0,1],OrdL2).
```

```
% P = L2+"V":
```

```
posetP(P,OrdP) :- P=[a,b,c,d], orddinsucc([(a,b),(b,c),(b,d)],P,OrdP).
```

```
% L3: lantul cu trei elemente:
```

```
l3(L3,OrdL3) :- L3=[0,a,1], orddinsucc([(0,a),(a,1)],L3,OrdL3).
```

% L2^2: rombul:

romb(R,OrdR) :- R=[\emptyset ,a,b,1], orddinsucc([(\emptyset ,a),(\emptyset ,b),(a,1),(b,1)],R,OrdR).

% L2+L2^2+L2:

latL(L,OrdL) :- L=[\emptyset ,a,x,y,b,1],
 orddinsucc([(\emptyset ,a),(a,x),(a,y),(x,b),(y,b),(b,1)],L,OrdL).

% L2^3: cubul:

cub(C,OrdC) :- C=[\emptyset ,a,b,c,x,y,z,1],
orddinsucc([(\emptyset ,a),(\emptyset ,b),(\emptyset ,c),(a,x),(a,y),(b,x),(b,z),(c,y),(c,z),(x,1),(y,1),(z,1)],C,OrdC)
.

% Diamantul: M3:

m3(M3,OrdM3) :- M3=[\emptyset ,a,b,c,1],
 orddinsucc([(\emptyset ,a),(\emptyset ,b),(\emptyset ,c),(a,1),(b,1),(c,1)],M3,OrdM3).

% Pentagonul: N5:

n5(N5,OrdN5) :- N5=[\emptyset ,x,y,z,1],
 orddinsucc([(\emptyset ,x),(x,1),(\emptyset ,y),(y,z),(z,1)],N5,OrdN5).

% Hexagonul:

```

hexa(M,Ord) :- M=[0,u,x,y,z,1],
               orddinsucc([(0,u),(u,x),(x,1),(0,y),(y,z),(z,1)],M,Ord).

/* Predicat care testeaza daca o functie F:A->B pastreaza relatiile binare  $R \leq A^2$  si  $S \leq B^2$ ,
mai precis daca duce pe R in S, adica are proprietatea ca, pentru orice x,y in A, daca (x,y)
in R, atunci (F(x),F(y)) in S: */

pastrel(F,R,S) :- not((member((X,Y),R), member((X,FX),F), member((Y,FY),F),
                       not(member((FX,FY),S)))).

% Functiile crescatoare F:P->Q intre doua poseturi (P,OrdP) si (Q,OrdQ):

fctcresc(F,P,OrdP,Q,OrdQ) :- functie(F,P,Q), pastrel(F,OrdP,OrdQ).

% Functiile strict crescatoare F:P->Q intre doua poseturi (P,OrdP) si (Q,OrdQ):

fctstrcresc(F,P,OrdP,Q,OrdQ) :- functie(F,P,Q), ordstrdinord(OrdP,OrdStrP),
                                ordstrdinord(OrdQ,OrdStrQ), pastrel(F,OrdStrP,OrdStrQ).

/* Interogati:
?- posetB(B,OrdB), l2(L2,OrdL2), fctcresc(F,B,OrdB,L2,OrdL2).
?- posetB(B,OrdB), l2(L2,OrdL2), fctstrcresc(F,B,OrdB,L2,OrdL2).
?- posetA(A,OrdA), posetB(B,OrdB), fctstrcresc(F,A,OrdA,B,OrdB).
si dati ;/Next pentru a obtine toate solutiile.
*/

% Multimea functiilor crescatoare injective de la posetul A la posetul B:

```

```
fctlecrescinjAlaB(LF) :- posetA(A,OrdA), posetB(B,OrdB),
    setof(F, (fctcresc(F,A,OrdA,B,OrdB), injectiv(F)), LF), !.
fctlecrescinjAlaB([]).
```

% Multimea functiilor crescatoare surjective de la posetul A la posetul B:

```
fctlecrescsurjAlaB(LF) :- posetA(A,OrdA), posetB(B,OrdB),
    setof(F, (fctcresc(F,A,OrdA,B,OrdB), surjectiv(F,B)), LF), !.
fctlecrescsurjAlaB([]).
```

```
/* Interogati:
?- fctlecrescinjAlaB(LF).
?- fctlecrescsurjAlaB(LF).
*/
```

```
/* Determinarea minorantilor, respectiv a majorantilor M ai unei submultimi S a unui poset
(P,OrdP): */
```

```
minoreaza(M,S,Ord) :- not((member(X,S), not(member((M,X),Ord)))).
```

```
minorant(M,S,P,OrdP) :- member(M,P), minoreaza(M,S,OrdP).
```

```
minorantii(S,P,OrdP,LM) :- setof(M, minorant(M,S,P,OrdP), LM), !.
minorantii(_,_,_,[]).
```

```
majoreaza(M,S,Ord) :- not((member(X,S), not(member((X,M),Ord)))).
```

```
majorant(M,S,P,OrdP) :- member(M,P), majoreaza(M,S,OrdP).
```

```
majorantii(S,P,OrdP,LM) :- setof(M, majorant(M,S,P,OrdP), LM), !.  
majorantii(_,_,_,[]).
```

```
/* Determinarea minimului, respectiv a maximului M unei multimi S raportat la ordinea Ord;  
ca si in cazul predicatelor minoreaza si majoreaza, Ord poate fi o ordine pe S sau pe o  
multime care include pe S; de fapt, pentru ca aceste predicate sa functioneze, este  
suficient ca Ord sa fie o lista care include o relatie de ordine pe S: */
```

```
min(S,Ord,M) :- minorant(M,S,S,Ord).
```

```
max(S,Ord,M) :- majorant(M,S,S,Ord).
```

```
/* Determinarea infimumului, respectiv a supremumului M al unei submultimi S a unui poset  
(P,OrdP): */
```

```
inf(S,P,OrdP,M) :- minorantii(S,P,OrdP,LM), max(LM,OrdP,M).
```

```
sup(S,P,OrdP,M) :- majorantii(S,P,OrdP,LM), min(LM,OrdP,M).
```

```
/* Determinarea elementelor minimale, respectiv a elementelor maximale M ale unei submultimi  
S a unui poset (P,Ord), apoi a listei LM a fiecarora dintre acestea: */
```

```
elemminimal(M,S,Ord) :- ordstrdinord(Ord,OrdStr), nueminstrict(M,S,OrdStr).
```

```

nueminstrict(M,S,OrdStr) :- member(M,S),
    not((member(X,S), member((X,M),OrdStr))).

elemmaximal(M,S,Ord) :- ordstrdinord(Ord,OrdStr), nuemajstrict(M,S,OrdStr).

nuemajstrict(M,S,OrdStr) :- member(M,S),
    not((member(X,S), member((M,X),OrdStr))).

elementelemiminale(S,Ord,LM) :- ordstrdinord(Ord,OrdStr),
    elementeminimale(S,OrdStr,LM).

elementeminimale(S,OrdStr,LM) :- setof(M, nueminstrict(M,S,OrdStr), LM), !.
elementeminimale(_,_,[]).

elementelemmaximale(S,Ord,LM) :- ordstrdinord(Ord,OrdStr),
    elementemaximale(S,OrdStr,LM).

elementemaximale(S,OrdStr,LM) :- setof(M, nuemajstrict(M,S,OrdStr), LM), !.
elementemaximale(_,_,[]).

% variante:

elemmin(M,S,Ord) :- member(M,S), not((member(X,S), member((X,M),Ord), X\=M)).

elemmax(M,S,Ord) :- member(M,S), not((member(X,S), member((M,X),Ord), X\=M)).

elemminimale(S,Ord,LM) :- setof(M, elemmin(M,S,Ord), LM), !.

```

```
elemminimale(_,_,[]).
```

```
elemmaximale(S,Ord,LM) :- setof(M, elemmax(M,S,Ord), LM), !.  
elemmaximale(_,_,[]).
```

```
/* Interogati:
```

```
?- posetA(A,OrdA), minorantii([a,b],A,OrdA,Minorantii), majorantii([a,b],A,OrdA,Majorantii).  
?- posetA(A,OrdA), inf([a,b],A,OrdA,Inf), sup([a,b],A,OrdA,Sup).  
?- posetA(A,OrdA), min([a,b],OrdA,Min).  
?- posetA(A,OrdA), max([a,b],OrdA,Max).  
?- posetA(A,OrdA), min([0,a,b,c],OrdA,Min), max([0,a,b,c],OrdA,Max).  
?- posetA(A,OrdA), min(A,OrdA,Min), max(A,OrdA,Max).  
?- posetB(B,OrdB), min(B,OrdB,Min).  
?- posetB(B,OrdB), max(B,OrdB,Max).  
?- posetB(B,OrdB), max(B,OrdB,Max), write(Max), min(B,OrdB,Min).  
?- posetP(P,OrdP), elemminimale(P,OrdP,MinP), elementelemminimale(P,OrdP,MinimaleP),  
elemmaximale(P,OrdP,MaxP), elementelemmaximale(P,OrdP,MaximaleP),  
elemminimale([c,d],OrdP,Mincd), elementelemminimale([a,b],OrdP,Minimaleab),  
elemmaximale([c,d],OrdP,Maxcd), elementelemmaximale([a,b],OrdP,Maximaleab).  
*/
```

```
% Multimea functiilor crescatoare de la  $L_2^2$  la  $L_2$ :
```

```
fctlecrescRomblaL2(LF) :- romb(R,OrdR), l2(L2,OrdL2),  
                        setof(F, fctcresc(F,R,OrdR,L2,OrdL2), LF), !.  
fctlecrescRomblaL2([]).
```

```

/* Interogati:
?- fctlecreascRomblaL2(LF), afislista(LF), length(LF,NrFct).
*/

/* Sa determinam daca un poset (L,OrdL) este latice (Ore), respectiv latice marginita,
respectiv latice marginita complementata: */

latice(L,OrdL) :- not((member(X,L), member(Y,L),
                    not((inf([X,Y],L,OrdL,_), sup([X,Y],L,OrdL,_)))))).

latmarg(L,OrdL) :- latice(L,OrdL), min(L,OrdL,_), max(L,OrdL,_).

latmargcomplem(L,OrdL) :- latice(L,OrdL), min(L,OrdL,Zero), max(L,OrdL,Unu),
                    not((member(X,L), not((member(Y,L),
                    inf([X,Y],L,OrdL,Zero), sup([X,Y],L,OrdL,Unu)))))).

% Alta varianta, cu determinarea complementilor fiecarui element complementat:

complem(X,L,OrdL,Y) :- min(L,OrdL,Zero), max(L,OrdL,Unu), member(Y,L),
                    inf([X,Y],L,OrdL,Zero), sup([X,Y],L,OrdL,Unu).

laticemargcomplem(L,OrdL) :- latmarg(L,OrdL),
                    not((member(X,L), not(complem(X,L,OrdL,_))))).

% Sau, fara calculul lui 0 si 1 repetat:

complem(X,L,OrdL,Zero,Unu,Y) :- member(Y,L),

```



```
inf([X,Y],L,OrdL,Zero), sup([X,Y],L,OrdL,Unu).
```

```
laticemarg(L,OrdL,Zero,Unu) :- lattice(L,OrdL),  
    min(L,OrdL,Zero), max(L,OrdL,Unu).
```

```
laticemargincomplem(L,OrdL) :- laticemarg(L,OrdL,Zero,Unu),  
    not((member(X,L), not(complem(X,L,OrdL,Zero,Unu,_))))).
```

```
/* Interogati:
```

```
?- latL(L,OrdL), lattice(L,OrdL).  
?- latL(L,OrdL), latmarg(L,OrdL).  
?- latL(L,OrdL), latmargcomplem(L,OrdL).  
?- posetA(A,OrdA), latmarg(A,OrdA).  
?- posetA(A,OrdA), latmargcomplem(A,OrdA).  
?- posetB(B,OrdB), lattice(B,OrdB).  
?- l2(L2,OrdL2), latmargcomplem(L2,OrdL2).  
?- l3(L3,OrdL3), latmarg(L3,OrdL3).  
?- l3(L3,OrdL3), latmargcomplem(L3,OrdL3).  
?- romb(R,OrdR), latmargcomplem(R,OrdR).  
?- m3(M3,OrdM3), latmargcomplem(M3,OrdM3), write(OrdM3).  
?- n5(N5,OrdN5), latmargcomplem(N5,OrdN5), write(OrdN5).  
?- l3(L3,OrdL3), complem(0,L3,OrdL3,Y).  
?- l3(L3,OrdL3), complem(1,L3,OrdL3,Y).  
?- l3(L3,OrdL3), complem(a,L3,OrdL3,Y).  
?- m3(M3,OrdM3), complem(0,M3,OrdM3,Y).  
?- m3(M3,OrdM3), complem(1,M3,OrdM3,Y).  
?- m3(M3,OrdM3), complem(a,M3,OrdM3,Y).
```

```

?- m3(M3,OrdM3), complem(b,M3,OrdM3,Y).
?- m3(M3,OrdM3), complem(c,M3,OrdM3,Y).
?- n5(N5,OrdN5), complem(0,N5,OrdN5,Y).
?- n5(N5,OrdN5), complem(1,N5,OrdN5,Y).
?- n5(N5,OrdN5), complem(x,N5,OrdN5,Y).
?- n5(N5,OrdN5), complem(y,N5,OrdN5,Y).
?- n5(N5,OrdN5), complem(z,N5,OrdN5,Y).
?- l3(L3,OrdL3), laticemarg(L3,OrdL3,Zero,Unu).
?- m3(M3,OrdM3), laticemarg(M3,OrdM3,Zero,Unu).
?- n5(N5,OrdN5), laticemarg(N5,OrdN5,Zero,Unu).
?- l3(L3,OrdL3), laticemargcomplem(L3,OrdL3).
?- l3(L3,OrdL3), laticemargincomplem(L3,OrdL3).
?- m3(M3,OrdM3), laticemargcomplem(M3,OrdM3), laticemargincomplem(M3,OrdM3).
?- n5(N5,OrdN5), laticemargcomplem(N5,OrdN5), laticemargincomplem(N5,OrdN5).
*/

```

```

/* Determinarea sublaticilor, respectiv a sublaticilor marginite S, ale unei latici,
respectiv latici marginite date prin laticia sa (Ore) subiacenta (L,OrdL): */

```

```

sublat(S,L,OrdL) :- sublista(S,L), not((member(X,S), member(Y,S),
      inf([X,Y],L,OrdL,XsiY), sup([X,Y],L,OrdL,XsauY),
      not((member(XsiY,S), member(XsauY,S)))))).

```

```

sublatmarg(S,L,OrdL) :- sublat(S,L,OrdL), inch0si1(S,L,OrdL).

```

```

inch0si1(S,L,OrdL) :- min(L,OrdL,Zero), max(L,OrdL,Unu),
      member(Zero,S), member(Unu,S).

```

```
sublaticemarg(S,L,OrdL,Zero,Unu) :- sublat(S,L,OrdL),  
    min(L,OrdL,Zero), max(L,OrdL,Unu), member(Zero,S), member(Unu,S).
```

```
% Sublaticile si sublaticile marginite ale lui  $L_2+L_2^2+L_2$ :
```

```
sublaticileL(LS) :- latL(L,OrdL), setof(S, sublat(S,L,OrdL), LS), !.  
sublaticileL([]).
```

```
sublaticilemargL(LS) :- latL(L,OrdL), setof(S, sublatmarg(S,L,OrdL), LS), !.  
sublaticilemargL([]).
```

```
sublaticilemarginiteL(LS) :- latL(L,OrdL),  
    setof(S, sublaticemarg(S,L,OrdL,_,_), LS), !.  
sublaticilemarginiteL([]).
```

```
% Sa determinam daca un poset (P,OrdP) este lant:
```

```
lant(P,OrdP) :- not((member(X,P), member(Y,P),  
    not((member((X,Y),OrdP) ; member((Y,X),OrdP))))).
```

```
/* Sa determinam sublaticile si sublaticile marginite liniar ordonate si pe cele neliniar  
ordonate ale lui  $L_2+L_2^2+L_2$  si sa demonstram ca toate subposeturile sale liniar ordonate  
sunt sublatici: */
```

```
sublatlinord(S,L,OrdL) :- sublat(S,L,OrdL), lant(S,OrdL).
```

```
sublaticilinordL(LS) :- latL(L,OrdL), setof(S, sublatlinord(S,L,OrdL), LS), !.  
sublaticilinordL([]).
```

```
sublatmarglinord(S,L,OrdL) :- sublatmarg(S,L,OrdL), lant(S,OrdL).
```

```
sublaticimarglinordL(LS) :- latL(L,OrdL),  
    setof(S, sublatmarglinord(S,L,OrdL), LS), !.  
sublaticimarglinordL([]).
```

```
sublatnelinord(S,L,OrdL) :- sublat(S,L,OrdL), not(lant(S,OrdL)).
```

```
sublaticinelinordL(LS) :- latL(L,OrdL),  
    setof(S, sublatnelinord(S,L,OrdL), LS), !.  
sublaticinelinordL([]).
```

```
sublatmargnelinord(S,L,OrdL) :- sublatmarg(S,L,OrdL), not(lant(S,OrdL)).
```

```
sublaticimargnelinordL(LS) :- latL(L,OrdL),  
    setof(S, sublatmargnelinord(S,L,OrdL), LS), !.  
sublaticimargnelinordL([]).
```

```
sublant(S,L,OrdL) :- sublista(S,L), lant(S,OrdL).
```

```
sublanturil(LS) :- latL(L,OrdL), setof(S, sublant(S,L,OrdL), LS), !.  
sublanturil([]).
```

```
sublantmarg(S,L,OrdL) :- sublista(S,L), lant(S,OrdL), inch0si1(S,L,OrdL).
```

```
sublanturimargL(LS) :- latL(L,OrdL), setof(S, sublantmarg(S,L,OrdL), LS), !.  
sublanturimargL([]).
```

```
toatelantsublat(L,OrdL) :- not((sublant(S,L,OrdL), not(sublat(S,L,OrdL))))).
```

```
toatelantsublatL :- latL(L,OrdL), toatelantsublat(L,OrdL).
```

```
/* Interogati:
```

```
?- l3(L3,OrdL3), lant(L3,OrdL3).
```

```
?- m3(M3,OrdM3), lant(M3,OrdM3).
```

```
?- sublaticileL(LS), afislista(LS), length(LS,NrSublat).
```

```
?- sublaticilemargL(LS), afislista(LS), length(LS,NrSublatmarg).
```

```
?- sublaticilemarginiteL(LS), afislista(LS), length(LS,NrSublatmarg).
```

```
?- sublaticilinordL(LS), afislista(LS), length(LS,NrSublant).
```

```
?- sublanturil(LS), afislista(LS), length(LS,NrSublant).
```

```
?- sublaticimarglinordL(LS), length(LS,NrSublantmarg).
```

```
?- sublanturimargL(LS), afislista(LS), length(LS,NrSublantmarg).
```

```
?- sublaticinelinordL(LS), length(LS,NrSublatnelinord).
```

```
?- sublaticimargnelinordL(LS), length(LS,NrSublatmargnelinord).
```

```
?- toatelantsublatL.
```

```
*/
```

```
/* Inchiderea la complement a unei submultimi S a unei latici marginite complementate  
(L,OrdL,Zero,Unu), varianta: */
```

```
inchcomplem(S,L,OrdL) :- min(L,OrdL,Zero), max(L,OrdL,Unu),
```

```
not((member(X,S), complen(X,L,OrdL,Zero,Unu,Y), not(member(Y,S))))).
```

```
inchcomplen(S,L,OrdL,Zero,Unu) :- not((member(X,S),  
    complen(X,L,OrdL,Zero,Unu,Y), not(member(Y,S))))).
```

```
/* Determinarea subalgebrelor booleene ale unei algebre Boole (B,OrdB), variante  
echivalente: */
```

```
subalgebrabool(S,B,OrdB) :- sublaticemarg(S,B,OrdB,Zero,Unu),  
    inchcomplen(S,B,OrdB,Zero,Unu).
```

```
subalgbool(S,B,OrdB) :- sublatmarg(S,B,OrdB), inchcomplen(S,B,OrdB).
```

```
subalgBoole(S,B,OrdB) :- sublat(S,B,OrdB), S\=[], inchcomplen(S,B,OrdB).
```

```
/* Lista subalgebrelor booleene ale cubului, cu fiecare dintre predicatele anterioare: */
```

```
subalgebreboolCub(LS) :- cub(B,OrdB), setof(S, subalgebrabool(S,B,OrdB), LS), !.  
subalgebreboolCub([]).
```

```
subalgboolCub(LS) :- cub(B,OrdB), setof(S, subalgbool(S,B,OrdB), LS), !.  
subalgboolCub([]).
```

```
subalgBooleCub(LS) :- cub(B,OrdB), setof(S, subalgBoole(S,B,OrdB), LS), !.  
subalgBooleCub([]).
```

```
/* Interogati:
```

```
?- subalgebreboolCub(LS), afislista(LS).
?- subalgboolCub(LS), afislista(LS).
?- subalgBooleCub(LS), afislista(LS).
*/
```

```
morflat(F,L,OrdL,M,OrdM) :- functie(F,L,M), pastrdisjconj(F,L,OrdL,M,OrdM).
```

```
pastrdisjconj(F,L,OrdL,M,OrdM) :- not((member(X,L), member(Y,L),
    member((X,FX),F), member((Y,FY),F),
    inf([X,Y],L,OrdL,XsiY), sup([X,Y],L,OrdL,XsauY),
    inf([FX,FY],M,OrdM,FXsiFY), sup([FX,FY],M,OrdM,FXsauFY),
    not((member((XsiY,FXsiFY),F), member((XsauY,FXsauFY),F))))).
```

```
morflatmarg(F,L,OrdL,M,OrdM) :- morflat(F,L,OrdL,M,OrdM),
    pastr0si1(F,L,OrdL,M,OrdM).
```

```
pastr0si1(F,L,OrdL,M,OrdM) :- min(L,OrdL,ZeroL), max(L,OrdL,UnuL),
    min(M,OrdM,ZeroM), max(M,OrdM,UnuM),
    auxpastr0si1(F,ZeroL,UnuL,ZeroM,UnuM).
```

```
auxpastr0si1(F,ZeroL,UnuL,ZeroM,UnuM) :- member((ZeroL,ZeroM),F),
    member((UnuL,UnuM),F).
```

```
pastrcomplem(F,L,OrdL,M,OrdM) :-
    min(L,OrdL,ZeroL), max(L,OrdL,UnuL), min(M,OrdM,ZeroM), max(M,OrdM,UnuM),
    pastrcomplem(F,L,OrdL,ZeroL,UnuL,M,OrdM,ZeroM,UnuM).
```

```
pastrcomplem(F,L,OrdL,ZeroL,UnuL,M,OrdM,ZeroM,UnuM) :- not((member(X,L),
    member((X,FX),F), complem(X,L,OrdL,ZeroL,UnuL,Y), member((Y,FY),F),
    not(complem(FX,M,OrdM,ZeroM,UnuM,FY)))).
```

```
pastreazacomplem(F,L,OrdL,M,OrdM) :-
    min(L,OrdL,ZeroL), max(L,OrdL,UnuL), min(M,OrdM,ZeroM), max(M,OrdM,UnuM),
    pastreazacomplem(F,L,OrdL,ZeroL,UnuL,M,OrdM,ZeroM,UnuM).
```

```
pastreazacomplem(F,L,OrdL,ZeroL,UnuL,M,OrdM,ZeroM,UnuM) :- not((member(X,L),
    member((X,FX),F), complem(X,L,OrdL,ZeroL,UnuL,Y),
    complem(FX,M,OrdM,ZeroM,UnuM,FY), not(member((Y,FY),F)))).
```

```
/* Morfismele de latici marginite intre algebre Boole sunt morfisme booleene, asadar
urmatoarele predicate intorc aceleasi solutii F ca morflatmarg: */
```

```
morfbool(F,L,OrdL,M,OrdM) :- morflatmarg(F,L,OrdL,M,OrdM),
    pastrcomplem(F,L,OrdL,M,OrdM).
```

```
morfbooleene(F,L,OrdL,M,OrdM) :- morflatmarg(F,L,OrdL,M,OrdM),
    pastreazacomplem(F,L,OrdL,M,OrdM).
```

```
/* Sa determinam morfismele de latici marginite, i.e. morfismele booleene, de la romb (L2^2)
la L2, apoi endomorfismele booleene ale rombului, apoi automorfismele booleene ale rombului,
apoi morfismele de latici marginite de la pentagon la diamant, apoi morfismele de latici
marginite de la hexagon la pentagon:
?- romb(R,OrdR), l2(L2,OrdL2), setof(F, morflatmarg(F,R,OrdR,L2,OrdL2), LF), afislista(LF),
length(LF,Cate).
```



```

?- romb(R,OrdR), l2(L2,OrdL2), setof(F, morfbool(F,R,OrdR,L2,OrdL2), LF), afislista(LF),
length(LF,Cate).
?- romb(R,OrdR), l2(L2,OrdL2), setof(F, morfbooleene(F,R,OrdR,L2,OrdL2), LF), afislista(LF),
length(LF,Cate).
?- romb(R,OrdR), setof(F, morflatmarg(F,R,OrdR,R,OrdR), LF), afislista(LF), length(LF,Cate).
?- romb(R,OrdR), setof(F, morfbool(F,R,OrdR,R,OrdR), LF), afislista(LF), length(LF,Cate).
?- romb(R,OrdR), setof(F, morfbooleene(F,R,OrdR,R,OrdR), LF), afislista(LF),
length(LF,Cate).
?- romb(R,OrdR), setof(F, (morflatmarg(F,R,OrdR,R,OrdR), injectiv(F), surjectiv(F,R)), LF),
afislista(LF), length(LF,Cate).
?- romb(R,OrdR), setof(F, (morfbool(F,R,OrdR,R,OrdR), injectiv(F), surjectiv(F,R)), LF),
afislista(LF), length(LF,Cate).
?- romb(R,OrdR), setof(F, (morfbooleene(F,R,OrdR,R,OrdR), injectiv(F), surjectiv(F,R)), LF),
afislista(LF), length(LF,Cate).
?- n5(N5,OrdN5), m3(M3,OrdM3), setof(F, morflatmarg(F,N5,OrdN5,M3,OrdM3), LF),
afislista(LF), length(LF,Nr).
?- hexa(M,Ord), n5(N5,OrdN5), setof(F, morflatmarg(F,M,Ord,N5,OrdN5), LF), afislista(LF),
length(LF,Nr).
*/

```

% Determinarea automorfismelor booleene ale cubului, variante echivalente:

```

automBooleCub(LF) :- cub(C,OrdC),
    setof(F, (morflatmarg(F,C,OrdC,C,OrdC), injectiv(F), surjectiv(F,C)), LF), !.
automBooleCub([]).

```

% predicatul anterior este mult prea lent; mai avantajos:

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

```
automboolCub(LF) :- cub(C,OrdC), setof(F, P^(permutare(C,P), fctbij(F,C,P),
morflatmarg(F,C,OrdC,C,OrdC)), LF), !.
automboolCub([]).
```

% si mai avantajos:

```
autombooleeneCub(LF) :- cub(C,OrdC), setof(F, P^(permutare(C,P), fctbij(F,C,P),
pastr0si1(F,C,OrdC,C,OrdC), pastrdisjconj(F,C,OrdC,C,OrdC)), LF), !.
autombooleeneCub([]).
```

% inca mai avantajos:

```
automorfbooleeneCub(LF) :- cub(C,OrdC), Cminus01=[a,b,c,x,y,z],
    setof(F, (P,G)^(permutare(Cminus01,P), fctbij(G,Cminus01,P),
    append([(0,0),(1,1)],G,F), pastrdisjconj(F,C,OrdC,C,OrdC)), LF), !.
automorfbooleeneCub([]).
```

% varianta echivalenta:

```
automorfismebooleeneCub(LF) :- cub(C,OrdC), Cminus01=[a,b,c,x,y,z],
    setof(F, (P,Q)^(permutare(Cminus01,P), append([0|P],[1],Q),
    fctbij(F,C,Q), pastrdisjconj(F,C,OrdC,C,OrdC)), LF), !.
automorfismebooleeneCub([]).
```

% Determinarea morfismelor booleene de la cub la romb, variante echivalente:

```
morfBooleCublaRomb(LF) :- cub(C,OrdC), romb(R,OrdR),  
    setof(F, morflatmarg(F,C,OrdC,R,OrdR), LF), !.  
morfBooleCublaRomb([]).
```

% si acest predicat calculeaza foarte lent; varianta mai avantajoasa:

```
morfboolCublaRomb(LF) :- cub(C,OrdC), romb(R,OrdR),  
    Cminus01=[a,b,c,x,y,z], setof(F, G^(functie(G,Cminus01,R),  
    append([(0,0),(1,1)],G,F), pastrdisjconj(F,C,OrdC,R,OrdR)), LF), !.  
morfboolCublaRomb([]).
```

/* Interogati:

```
?- romb(C,OrdC), setof(F, P^(permutare(C,P), fctbij(F,C,P), pastr0si1(F,C,OrdC,C,OrdC),  
pastrdisjconj(F,C,OrdC,C,OrdC)), LF), afislista(LF), length(LF,Nr).
```

```
?- automboolCub(LF), afislista(LF), length(LF,Nr).
```

```
?- autombooleeneCub(LF), afislista(LF), length(LF,Nr).
```

```
?- automorfbooleeneCub(LF), afislista(LF), length(LF,Nr).
```

```
?- automorfismebooleeneCub(LF), afislista(LF), length(LF,Nr).
```

```
?- morfboolCublaRomb(LF), afislista(LF), length(LF,Nr).
```

Teste (interogari intermediare):

```
?- Cminus01=[a,b,c,x,y,z], Rminus01=[a,b], setof(G, functie(G,Cminus01,Rminus01), L),  
afislista(L), length(L,Nr).
```

```
?- Cminus01=[a,b,c,x,y,z], Rminus01=[a,b], functie(G,Cminus01,Rminus01),  
append([(0,0),(1,1)],G,F).
```

```
?- cub(C,OrdC), romb(R,OrdR), Cminus01=[a,b,c,x,y,z], functie(G,Cminus01,R),
append([(0,0),(1,1)],G,F), write(F), nl, pastrdisjconj(F,C,OrdC,R,OrdR).
```

```
?- cub(C,OrdC), romb(R,OrdR),
pastrdisjconj([(0,0),(a,0),(b,0),(c,1),(x,0),(y,1),(z,1),(1,1)],C,OrdC,R,OrdR).
```

Pentru a testa output-ul predicatului morfBooleCublaRomb cu solutia matematica a acestui exercitiu, putem proceda astfel:

```
?- morfboolCublaRomb(LF), afislistamorf(LF), length(LF,Nr).
```

```
*/
```

```
afislistamorf(LF) :- cub(C,_), scrieIrand(C), auxafislistamorf(LF).
```

```
scrieIrand([]) :- nl,
```

```
write('-----'), nl.
```

```
scrieIrand([H|T]) :- write('f('), write(H), write(')'), tab(3), scrieIrand(T).
```

```
scriefct([]) :- nl.
```

```
scriefct([(_,FX)|F]) :- auxscriefct(FX), tab(1), scriefct(F).
```

```
auxscriefct(0) :- write('(0,0)').
```

```
auxscriefct(a) :- write('(0,1)').
```

```
auxscriefct(b) :- write('(1,0)').
```

```
auxscriefct(1) :- write('(1,1)').
```

```
auxafislistamorf([]).
```

```
auxafislistamorf([F|LF]) :- transffct(F,H), scriefct(H), auxafislistamorf(LF).
```

```
transffct([(0,0),(1,1)|G],H) :- append([(0,0)|G],[(1,1)],H).
```