```
/* Fie multimile A,B,C arbitrare, fixate.
   Fie x arbitrar, fixat.
   Notam cu variabilele A,B,C (am putea folosi _a,_b,_c respectiv) proprietatile:
A: x apartine lui A
B: x apartine lui B
C: x apartine lui C
   Sa demonstram idempotenta reuniunii: AUA=A.
   Avem de demonstrat: x apartine lui AUA <=> x apartine A, adica:
(x apartine lui A sau x apartine lui A) <=> x apartine lui A, adica:
A ; A <=> A, adica:
pentru orice A membru al multimii [false,true], A ; A <=> A, adica A ; A are aceeasi valoare de
adevar ca si A.
   La fel pentru idempotenta intersectiei: A^A=A, unde am notat cu ^ intersectia de multimi.
   Aici, in plain text, voi mai nota cu:
0 multimea vida,
\ diferenta intre multimi,
/\ diferenta simetrica intre multimi,
x produsul cartezian de multimi,
<= incluziunea nestricta intre multimi (i.e. inclus sau egal cu),
< incluziunea stricta intre multimi (inclus strict in),
>= incluziunea nestricta in sens invers intre multimi (include sau e egal cu),
> incluziunea stricta in sens invers intre multimi (include strict pe). */

:- op(300,xfx,xor).

implica(P,Q) :- not(P); Q.
echiv(P,Q) :- implica(P,Q), implica(Q,P).
implicstr(P,Q) :- implica(P,Q), not(echiv(P,Q)).
P xor Q :- P,not(Q) ; Q,not(P).

idempreun(A) :- echiv(A;A, A).
```

```
idempinters(A) :- echiv((A,A), A).

demidempreun :- not((member(A,[false,true]), write(A), nl, not(idempreun(A))))).
demidempinters :- not((member(A,[false,true]), write(A), nl, not(idempinters(A))))).

/* Sa demonstram comutativitatea reuniunii, a intersectiei si a diferentei simetrice:
AUB=BUA, A^B=B^A si A/\B=B/\A: */

comutreun(A,B) :- echiv(A;B, B;A).
comutinters(A,B) :- echiv((A,B), (B,A)).
comutdifsim(A,B) :- echiv(A xor B, B xor A).

demcomutreun :- not((member(A,[false,true]), member(B,[false,true]), write((A,B)), nl,
not(comutreun(A,B)))).
demcomutinters :- not((member(A,[false,true]), member(B,[false,true]), write((A,B)), nl,
        not(comutinters(A,B)))).
demcomutdifsim :- not((member(A,[false,true]), member(B,[false,true]), write((A,B)), nl,
        not(comutdifsim(A,B)))).

/* Sa demonstram asociativitatea reuniunii, a intersectiei si a diferentei simetrice:
AU(BUC)=(AUB)UC, A^(B^C)=(A^B)^C si A/\(B/\C)=(A/\B)/\C: */

asocreun(A,B,C) :- echiv(A;(B;C), (A;B);C).
asocinters(A,B,C) :- echiv((A,(B,C)), ((A,B),C)).
asocdifsim(A,B,C) :- echiv(A xor (B xor C), (A xor B) xor C).

demasocreun :- not((member(A,[false,true]), member(B,[false,true]),
                member(C,[false,true]), write((A,B,C)), nl, not(asocreun(A,B,C)))).
demasocinters :- not((member(A,[false,true]), member(B,[false,true]),
                member(C,[false,true]), write((A,B,C)), nl, not(asocinters(A,B,C)))).
demasocdifsim :- not((member(A,[false,true]), member(B,[false,true]),
```

```
                member(C,[false,true]), write((A,B,C)), nl, not(asocdifsim(A,B,C)))).

/* Observam ca: A<B <=> (A<=B si non(A=B)) <=>
<=> implicstr(x apartine lui A, x apartine lui B)
<=> implicstr(A,B), cu notatiile de mai sus pentru variabilele A,B.
*/

/* Sa demonstram ca reuniunea (binara sau, mai general, nevida - vom vedea) isi include termenii,
iar intersectia (binara sau, mai general, nevida - vom vedea) e inclusa in termenii sai: A<=AUB si
A^B<=A: */

reunincltерm(A,B) :- implica(A, A;B).
intersinclinterm(A,B) :- implica((A,B), A).

demreunincltерm :- not((member(A,[false,true]), member(B,[false,true]), write((A,B)), nl,
        not(reunincltерm(A,B))))).
demintersinclinterm :- not((member(A,[false,true]), member(B,[false,true]), write((A,B)),
        nl, not(intersinclinterm(A,B))))).

% Sa demonstram ca multimea vida e inclusa (nestrict) in orice multime: 0<=A:

vidainclorice(A) :- implica(false,A).

demvidainclorice :- not((member(A,[false,true]), write(A), nl, not(vidainclorice(A))))).

/* Sa demonstram ca orice multime e inclusa (nestrict) in ea insasi si nu e inclusa strict in ea
insasi: A<=A si non(A<A): */

incleainsasi(A) :- implica(A,A).
noninclstreainsasi(A) :- not(implicstr(A,A)).
```

```prolog
demincleainsasi :- not((member(A,[false,true]), write(A), nl, not(incleainsasi(A)))).
demnoninclstreainsasi :- not((member(A,[false,true]), write(A), nl,
                          not(noninclstreainsasi(A)))).

% Sa demonstram ca singura parte a multimii vide este multimea vida: A<=0 <=> A=0:

sgpartevide(A) :- echiv(implica(A,false), echiv(A,false)).

demsgpartevide :- not((member(A,[false,true]), write(A), nl, not(sgpartevide(A)))).

% Sa demonstram tranzitivitatea incluziunii (nestricte): (A<=B si B<=C) => A<=C:

tranzincl(A,B,C) :- implica((implica(A,B), implica(B,C)), implica(A,C)).

demtranzincl :- not((member(A,[false,true]), member(B,[false,true]),
                member(C,[false,true]), write((A,B,C)), nl, not(tranzincl(A,B,C)))).

/* Sa demonstram ca intersectand in ambii membri ai unei incluziuni (nestricte) cu o multime se
pastreaza sensul incluziunii: A<=B => A^C<=B^C: */

intersambiim(A,B,C) :- implica(implica(A,B), implica((A,C), (B,C))).

demintersambiim :- not((member(A,[false,true]), member(B,[false,true]),
                member(C,[false,true]), write((A,B,C)), nl, not(intersambiim(A,B,C)))).
```