```
:-dynamic pos/1, minim/1, minp/1.

/* ------------------------------------------------------------ */
/* 7 */

projection([],_,[]):-!.
projection(_,[],[]):-!.
projection(D,[H|T],R) :-
  sel(D,H,R1),projection(D,T,RS), appRes(R1,RS,R).

sel([],_,[]).
sel([R|RS],C,[S|SS]) :-
  ind(R,C,S), sel(RS,C,SS).

ind([H|_],1,[H]) :- !.
ind([_|T],N,R) :-
  NN is N-1, ind(T,NN,R).

appRes(R,[],R).
appRes([R1|RS1],[R2|RS2],[R|RS]) :-
  appd(R1,R2,R), appRes(RS1,RS2,RS).

appd([],R,R).
appd([H|T],Y,[H|R]) :-
  appd(T,Y,R).


/* ------------------------------------------------------------ */
/* 5 */

/*

lvar(NAME).      promenna
lapp(E1,E2).     aplikace
labs(NAME,E).    abstrakce

*/

/* subst(What,Where,Var,Res) */

subst(What,lvar(N),N,What) :- !.
subst(What,lvar(N),_,lvar(N)) :- !.
subst(What,lapp(E1,E2),Var,lapp(NE1,NE2)) :-
  subst(What,E1,Var,NE1),
  subst(What,E2,Var,NE2).
subst(What,labs(Name,E),Name,labs(Name,E)) :- !.
subst(What,labs(Name,E),Var,labs(Name,NE)) :-
  subst(What,E,Var,NE).


/* ------------------------------------------------------------ */
/* 8 */

search(From,To,Shrt) :-
  retractall(pos(_)),
  retractall(minim(_)),
  retractall(minp(_)),
  assertz(minim(-1)),
  assertz(minp([])),
  bagof(L,findp(From,To,1,[]),_),
  minp(Shrt).

findp(F,F,X,CP) :-
  (minim(-1),retract(minim(-1));
   minim(MV),X<MV,retract(minim(MV))),
  app(CP,[F],MP),
  retract(minp(_)),
  assertz(minp(MP)),
  assertz(minim(X)),!.
findp(F,F,_,_) :- !, fail.
findp(F,T,N,P) :-
  assertz(pos(F)),
  nextStep(F,NF),
  not(pos(NF)),
  NN is N+1,
  app(P,[F],NMP),
  findp(NF,T,NN,NMP).
findp(F,_,_,_) :-
  pos(F),
  retract(pos(F)),
  !, fail.


app([],L,L).
app([H|T],YS,[H|TT]) :-
  app(T,YS,TT).

/* toto nebylo treba */

nextStep(p(X,Y),p(XX,YY)) :-
  move(I,J),
  XX is X+I,
  YY is Y+J,
  XX > 0, YY > 0,
  XX < 4, YY < 4.

move(1,0).
move(0,1).
move(-1,0).
move(0,-1).


/* ------------------------------------------------------------ */
/* 5 */

tryeval(E,Y,V,F,T) :-
  nonvar(V),!,
  V >= F,
  V =< T,
  Y is E.

tryeval(E,Y,V,F,T) :-
  var(V),!,
  gen(F,T,V),
  Y is E.

gen(F,_,F).
gen(F,T,W) :-
  gen(F,T,V),
  W is V+1,
  (W =< T; W > T, !,fail).


/* ------------------------------------------------------------ */
/* 5 */


propsub(Su,Set) :-
  subs(Su,Set),
  len(Su,Lu),
  len(Set,Le),
  Lu < Le.

len([],0).
len([_|T],N) :-
  len(T,NN),
  N is NN+1.

subs([],_).
subs([H|T],L) :-
  elem(H,L),
  subs(T,L).

elem(H,[H|_]) :- !.
elem(H,[_|T]) :-
  elem(H,T).




/* ------------------------------------------------------------ */
/* ------------------------------------------------------------ */
```