

```

/* ----- */
/* ----- */

:- dynamic
    pom/1.

/* ----- */
/* ----- */

getVal([p(K,V)|_], K, V) :- !.
getVal([_|PS], K, V) :- getVal(PS,K,V).

getKeys([],_,[]).
getKeys([p(K,V)|PS], V, [K|KS]) :- !, getKeys(PS,V,KS).
getKeys([_|PS], V, KS) :- getKeys(PS,V,KS).

insOverIn([p(K,_)|PS],K, V, [p(K,V)|PS]) :- !.
insOverIn([P|PS], K, V, [P|NPS]) :- insOverIn(PS,K,V,NPS).
insOverIn([], K, V, [p(K,V)]).

assocList(AL,K,V,NAL) :-
    nonvar(K),
    nonvar(V),!,
    insOverIn(AL,K,V,NAL).
assocList(AL,K,V,AL) :-
    nonvar(K),
    var(V),!,
    getVal(AL,K,V).
assocList(AL,KS,V,AL) :-
    var(KS),
    nonvar(V),
    getKeys(AL,V,KS).

/* ----- */
/* ----- */

getPaths(S,E,Paths) :-
    retractall(pom(_)),
    setof(P,search(S,S,E,P),Paths).

search(_,E,E,[E]) :- !.
search(S,S,E,[S|P]) :-
    assertz(pom(S)),
    nextStep(S,N),
    not(pom(N)),
    search(S,N,E,P).
search(S,S,_,_) :-
    pom(S),
    retract(pom(S)),
    !, fail.
search(P,C,E,[C|T]) :-
    assertz(pom(C)),
    nextStep(C,N),
    not(pom(N)),
    testWay(P,C,N),
    search(C,N,E,T).
search(_,C,_,_) :-
    pom(C),
    retract(pom(C)),
    !, fail.

/* ----- */
/* jen pro test */

testWay(pos(X1,Y1),pos(X2,Y2),pos(X3,Y3)) :-
    (X1==X2,X2==X3);
    (Y1==Y2,Y2==Y3);
    (X1<X2,Y3>Y2);
    (Y1<Y2,X3<X2);
    (X2<X1,Y3<Y2);
    (Y2<Y1,X3>X2).

```

```

nextStep(pos(X,Y),pos(XX,Y)) :-
    XX is X + 1, check(XX,Y).
nextStep(pos(X,Y),pos(X,YY)) :-
    YY is Y + 1, check(X,YY).
nextStep(pos(X,Y),pos(XX,Y)) :-
    XX is X - 1, check(XX,Y).
nextStep(pos(X,Y),pos(X,YY)) :-
    YY is Y - 1, check(X,YY).

check(X,Y) :-
    X > 0, X =< 5, Y > 0, Y =< 5.

/* ----- */
/* ----- */

atMost([],_,N) :-
    N < 0, !, fail.
atMost([],_,N) :-
    N >= 0.
atMost([P|PS],AS,N) :-
    C =.. [P|AS],
    call(C), !,
    NN is N-1,
    atMost(PS,AS,NN).
atMost([_|PS],AS,N) :-
    atMost(PS,AS,N).

/* --- na testy --- */

gt(X,Y) :- X>Y.
ge(X,Y) :- X>=Y.
lt(X,Y) :- X<Y.
le(X,Y) :- X<=Y.
eq(X,Y) :- X==Y.
ne(X,Y) :- X\==Y.

/* ----- */
/* ----- */

/*

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

*/

/* fv(Where,FreeVars) */

fv(X,Y) :-
    fv(X,[],Y).

fv(lvar(N),B,[]) :-
    member(N,B),!.
fv(lvar(N),_,[N]).
fv(lapp(E1,E2),B,R) :-
    fv(E1,B,R1),
    fv(E2,B,R2),
    uni(R1,R2,R).
fv(labs(N,E),B,R) :-
    fv(E,[N|B],R).

uni([],X,X) :- !.
uni(X,[],X) :- !.
uni([H|T],X,Y) :-
    member(H,X),!,
    uni(T,X,Y).
uni([H|T],X,[H|Y]) :-
    uni(T,X,Y).

```

/ * _____ * /
 / * _____ * /