

```
1  :- dynamic p/1.
2  :- dynamic q/1.
3  :- dynamic r/1.
4  r(X) :- q(X).
5  q(X) :- p(X).
6  p(t).
7
8  :- dynamic qs/2.
9  :- dynamic part/4.
10 :- dynamic app/3.
11 qs([],[]).
12 qs([X|Xs],Ys) :-
13     part(X,Xs,Littles,Bigs),
14     qs(Littles,Ls),
15     qs(Bigs,Bs),
16     app(Ls,[X|Bs],Ys).
17 part(_,[],[],[]).
18 part(X,[Y|Xs],[Y|Ls],Bs) :- X > Y, part(X,Xs,Ls,Bs).
19 part(X,[Y|Xs],Ls,[Y|Bs]) :- X <= Y, part(X,Xs,Ls,Bs).
20
21 app([],Ys,Ys).
22 app([X|Xs],Ys,[X|Zs]) :- app(Xs,Ys,Zs).
23
24 solve(Pred,N):-
25     setval(step_count,0),
26     (solve_bt(Pred) ->
27         write('Success. '),nl;
28         write('Failed. '),nl),
29     getval(step_count,N).
30
31 solve_bt(true) :- !.
32 solve_bt(A) :- arithmetic(A),incval(step_count), !,A.
33 solve_bt((A,B)) :- !,solve_bt(A), solve_bt(B).
34 solve_bt(H) :-
35     clause(H,Body),
36     incval(step_count),
37     solve_bt(Body).
38
39 arithmetic(_<_).
40 arithmetic(_>_).
41 arithmetic(_<=_).
42 arithmetic(_>=_).
43 arithmetic(_:=_).
44 arithmetic(_=\=_).
```

```
1 propagate(Cons,Doms,{NewDomains}):-
2     arrayify(Cons,Constraints),
3     arrayify(Doms,VarsDomains),
4     var_dom(VarsDomains,Variables,Domains),
5     domains(Variables,Domains,Constraints,NewDs),
6     (
7         fromto( (Variables,NewDs,NewDomains),
8                 ([V|RestVar],[D|RestDom],[V:D,RestDomains]),
9                 (RestVar,RestDom,RestDomains),
10                ([VLast],[DomLast],VLast:DomLast)) do true
11     ),!.
12
13 domains(Variables,Domains,Constraints,Ranges) :-
14     findall(A,all_solns(Variables,Domains,Constraints,A),Solutions),
15     (
16         fromto( (Solutions,Ranges),
17                 (Solns,[Range|RestRans]),
18                 (RestSolns,RestRans),
19                 ([[]|_],[[]])
20             ) do minmax(Solns,Range),cutcolumn(Solns,RestSolns)
21     ).
22
23 all_solns(Variables,Domains,Constraints,A) :-
24     length(Domains,N),length(A,N),
25     (
26         fromto( (Variables,Domains,A),
27                 ([X|RestV],[D|RestD],[E|RestA]),
28                 (RestV,RestD,RestA),
29                 ([],[],[[]])
30             ) do member(E,D), X is E
31     ),(
32         foreach(Con,Constraints) do Con
33     ).
34
35 minmax(List,(Min..Max)) :-
36     findall(A,member([A|_],List),D),
37     min(D,Min),
38     max(D,Max).
39
40 cutcolumn(List,Result) :-
41     fromto( (List,Result),
42             ([[_|RestVals]|RestList],[RestVals|RestResult]),
43             (RestList,RestResult),
44             ([],[[]])) do true.
45
46 arrayify({G},Res) :-!, arrayify(G,Res).
47 arrayify((X,Y),[X|Rest]) :-!, arrayify(Y,Rest).
```

```
48 arrayify(Y,[Y]).
49
50 consec(N,N,[N]).
51 consec(N,M,[N|Rest]) :- N<M,N1 is N+1,consec(N1,M,Rest),!.
52
53 var_dom([],[],[]).
54 var_dom([Var:Start..End|Tail],[Var|VarRest],[Dom|DomRest]) :-
55     consec(Start,End,Dom),
56     var_dom(Tail,VarRest,DomRest).
```

```
1  solve(true,_) :- !.
2  solve((A,B),Callstack) :- !,
3      solve(A,Callstack),
4      solve(B,Callstack).
5  solve(H,Callstack) :-!,
6      clause(H,Body),
7      (strict_member(H,Callstack) ->
8          write('Infinite loop detected, execution aborted'),nl;
9          solve(Body,[H|Callstack])
10     ).
11
12
13  arithmetic(_<_).
14  arithmetic(_>_).
15  arithmetic(_=<_).
16  arithmetic(_>=_).
17  arithmetic(_:=:_).
18  arithmetic(_=\=_).
19
20  strict_member(H,[H1]) :- H==H1.
21  strict_member(H,[H1|_]) :- H==H1,! .
22  strict_member(H,[_|T]) :- strict_member(H,T).
23
24  :- dynamic p/2.
25  :- dynamic q/1.
26  :- dynamic r/1.
27
28  p(X,[X]).
29  p(X,[H|T]) :- p(X,T).
30
31
```