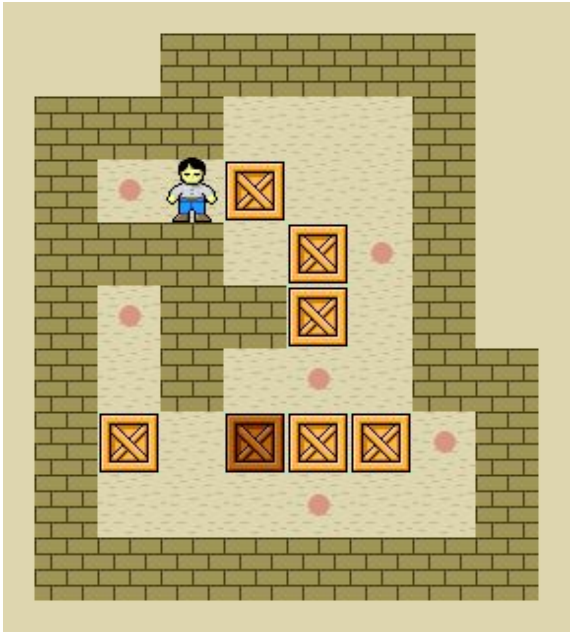
A Tabled Prolog Program for Solving Sokoban

Autores: Neng-Fa Zhou, Agostino Dovier
Apresentação: Vinícius Marcondes Vieira

Problema: Sokoban

- Levar cada caixa para a devida posição
- Menor quantidade de movimentos
- Evitar repetições e loops
- Deadlocks



Algoritmo

- B-Prolog
- Utiliza tabelamento para armazenar caminhos
- Simplificação do problema a problema de menor caminho
- Objetivos se dividem em dois
- Alto uso de memória
- Velocidade

```
:-table plan_sokoban(+,+,-,min).
plan_sokoban(_SokobanLoc,BoxLocs,Plan,Len):-
    goal_reached(BoxLocs),!,
    Plan=[],Len=0.
plan_sokoban(SokobanLoc,BoxLocs,[push(BoxLoc,Dir,DestLoc)|Plan],Len):-
    select(BoxLoc,BoxLocs,BoxLocs1),
    neib(PrevNeibLoc,BoxLoc,Dir),
    \+ member(PrevNeibLoc,BoxLocs1),
    neib(BoxLoc,NextNeibLoc,Dir),
    good_dest(NextNeibLoc,BoxLocs1),
    reachable_by_sokoban(SokobanLoc,PrevNeibLoc,BoxLocs),
    choose_dest(BoxLoc,NextNeibLoc,Dir,DestLoc,NewSokobanLoc,BoxLocs1),
    insert_ordered(DestLoc,BoxLocs1,NewBoxLocs),
    plan_sokoban(NewSokobanLoc,NewBoxLocs,Plan,Len1),
    Len is Len1+1.

:-table reachable_by_sokoban/3.
reachable_by_sokoban(Loc,Loc,_BoxLocs).
reachable_by_sokoban(Loc1,Loc2,BoxLocs):-
    neib(Loc1,Loc3,_),
    \+ member(Loc3,BoxLocs),
    reachable_by_sokoban(Loc3,Loc2,BoxLocs).

good_dest(Loc,BoxLocs):-
    \+ member(Loc,BoxLocs),
    (corner(Loc)->storage(Loc);true),
    foreach(BoxLoc in BoxLocs, \+ stuck(BoxLoc,Loc)).

choose_dest(Loc,NextLoc,_Dir,Dest,NewSokobanLoc,_BoxLocs):-
    Dest=NextLoc, NewSokobanLoc=BoxLoc.
choose_dest(Loc,NextLoc,Dir,Dest,NewSokobanLoc,BoxLocs):-
    neib(NextLoc,NextNextLoc,Dir),
    good_dest(NextNextLoc,BoxLocs),
    choose_dest(NextLoc,NextNextLoc,Dir,Dest,NewSokobanLoc,BoxLocs).
```

```
:-table neib/3.
neib(Loc1,Loc2,up):-
    top(Loc1,Loc2).
neib(Loc1,Loc2,down):-
    top(Loc2,Loc1).
neib(Loc1,Loc2,right):-
    right(Loc1,Loc2).
neib(Loc1,Loc2,left):-
    right(Loc2,Loc1).

goal_reached([]).
goal_reached([Loc|Locs]):-
    storage(Loc),
    goal_reached(Locs).

:-table corner/1.
corner(X):-
    \+ top(X,_),
    (\+ right(_,X); \+ right(X,_)),!.
corner(X):-
    \+ top(_,X),
    (\+ right(_,X); \+ right(X,_)),!.

:-table stuck/2.
stuck(X,Y):-
    (right(X,Y);right(Y,X)),
    (\+ storage(X); \+ storage(Y)),
    (\+ top(X,_), \+ top(Y,_);
     \+ top(_,X), \+ top(_,Y)),!.
stuck(X,Y):-
    (top(X,Y);top(Y,X)),
    (\+ storage(X); \+ storage(Y)),
    (\+ right(X,_), \+ right(Y,_);
     \+ right(_,X), \+ right(_,Y)),!.
```

Resultados

- Passou em 11 de 15 testes
- Falta de memória
- Pode ser melhorado aplicando conceitos mais avançados de programação
- Analisa todas as possíveis soluções, inclusive as não ótimas

COMPETITION RESULTS (CPU TIME, SECONDS).

Instance	BPSolver	Clasp
1-sokoban-optimization-0-0.asp	0.58	0.06
13-sokoban-optimization-0-0.asp	0.06	0.74
18-sokoban-optimization-0-0.asp	0.00	9.80
20-sokoban-optimization-0-0.asp	33.57	13.24
24-sokoban-optimization-0-0.asp	2.66	3.52
27-sokoban-optimization-0-0.asp	0.78	1.16
29-sokoban-optimization-0-0.asp	0.78	2.92
33-sokoban-optimization-0-0.asp	1.96	26.74
37-sokoban-optimization-0-0.asp	0.38	8.52
4-sokoban-optimization-0-0.asp	Mem Out	0.62
43-sokoban-optimization-0-0.asp	Mem Out	35.67
45-sokoban-optimization-0-0.asp	Mem Out	9.30
47-sokoban-optimization-0-0.asp	Mem Out	18.66
5-sokoban-optimization-0-0.asp	0.00	0.16
9-sokoban-optimization-0-0.asp	0.00	2.12

Referência

N. -F. Zhou and A. Dovier, "A Tabled Prolog Program for Solving Sokoban," 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence, 2011, pp. 896-897, doi: 10.1109/ICTAI.2011.145.

FIM