

Term block - 7

Problem Statement :-

Design an algorithm for to implement breadth first search and develop a program for the same.

Theory :-

The breadth first search algorithm is used to search a tree as graph structure for a node that meets a set of criteria.

Algorithm :-

Breadth first Search()

1. $open \leftarrow (start, NIL)$

$closed \leftarrow ()$

while not Nil($open$)

do not pair \leftarrow Head($open$)

$node \leftarrow$ Head($notpair$)

if GoalTest($node$) = True

then return ReconstructPath($notpair, closed$)

else $closed \leftarrow$ cons($notpair, closed$)

$children \leftarrow$ moveGen($node$)

$nodes \leftarrow$ RemoveSeen($children, open, closed$)

$new \leftarrow$ makePairs($nodes, node$)

$open \leftarrow$ Append(Tail($open$), new)

return "No solution found",

Program

BFS in Prolog

s(a,b)

s(a,c)

s(b,d)

s(b,e)

s(e,f)

s(c,g)

s(d,h)

s(e,i)

s(e,j)

s(f,k)

goal(t)

goal(j)

Solve(Start, Solution):-

breadthfirst([Start], Solution)

breadthfirst([[Node]path], [Node|path]):-goal(Node)

breadthfirst([path|paths], Solution):-

contant(path, newpaths), write(newpaths), nl, conc(paths

newpaths, paths!), breadthfirst([path], Solution).

entend([Node(path], newpaths):-

bagot([NewNode, Nodepath], (S[Node, NewNode], not(member(NewNode, [Node[path]]))), newpaths ? 1 -- entend(, []).

conc([], L, L)

conc([X|L1], L2, [X|L3]) :- write('conc'), write(X), write(L3),

write(" "), write(L1), write(" "), write(L2), conc(L1, L2, L3)

output:

	A	B	C	D	E	F
A	0	1	1	1	0	0
B	1	0	0	1	1	0
C	1	0	0	1	0	1
D	1	1	1	0	1	1
E	0	1	0	1	0	1
F	0	0	1	1	1	0

BFS traversal: B A D E C F

Conclusion:

In this termwork we learnt about - Breadth first Search algorithm & implemented code in Prolog