AI

EXPERIMENT NO-02

AIM:

Implement water jug problem using BFS or DFS (uniformed search)

Theory:

Problem statement →

In water jug problem in Artificial Intelligence we are provided with two jugs. One having the capacity to hold 3 gallons of water and other has capacity to 4 gallons. There is no other measuring equipment available and jugs also do not have any kind of marking on them. So the agents task here is to fill 4 gallon jug with 2 gallon of water by only these two jugs and no other material. Initially both our jugs are empty.

To solve this problem following set of rules were proposed

Prediction rules for solving the water jug problem. Here set x denotes the 4 gallon jug and y denotes 3 gallon jug

Sr NO	Initial state	condition	final state	Description of action
1	(x,y)	if $x<4$	$(4,y)$	fill 4 gallon jug full
2	(x,y)	if $y<3$	$(x,3)$	full 3 gallon jug full
3	(x,y)	if $x>0$	$(x-d,y)$	Pour some from 4 gallon
4	(x,y)	if $y>0$	$(x,y-d)$	Pour some from 3 gallon

5) (x,y)	if x>0	(0,y)	Empty 4 gallon
6) (x,y)	if y>0	(4,y-[4-x])	Pour some water from 3 gallon jug to full 4 gallon jug.
7) (x,y)	if (x+y)<7	(4,y-[4-x])	Pour some water from 3 gallon to full 4 gallon jug.
8) (x,y)	if (x+y<7)	(x-[3-y],y]	Pour some water from 4 gallon jug to full 3 gallon
9) (x,y)	if x+y<4	(x+y,0)	Pour all water from 3 gallon to 4 gallon jug
10) (x,y)	if (x+y)<3	(0,y+x)	Pour all water from 4 gallon jug to 3 gallon jug.

To solve Problem in minimum number of moves following set of rules in given respe sequences should be preffered.

Solution

Sr NO	4 gallon jug contents	3 gallon jug contents	Rule followed
1	0 gallon	0 gallon	Initial State
2	0 gallon	3 gallon	Rule no 2
3	3 gallon	0 gallon	Rule no 9
4	3 gallon	3 gallon	Rule no 2
5	4 gallon	2 gallon	Rule no 7
6	0 gallon	2 gallon	Rule no 5
7	2 gallon	0 gallon	Rule no 9

on reaching the 7 attempt we reach a state whis is our goal State. Therefore at this state we have solved our problem.

Conclusion: Thus we have successfully solved the Problem of water jug & implemented it using prolog language

SWISH File Edit Examples Help

Program

```
member(X,[X|_]).
member(X,[Y|Z]):-member(X,Z).
```

Singleton variables: [Y]

```

move(X,Y,_):-X=:=2,Y=:=0,write('done'),!.
move(X,Y,Z):-X<4,\+member((4,Y),Z),write("fill 4 jug"),nl,move(4,Y,[(4,Y)|Z]).
move(X,Y,Z):-Y<3,\+member((X,3),Z),write("fill 3 jug"),nl,move(X,3,[(X,3)|z]).
move(X,Y,Z):-X>0,\+member((0,Y),Z),write("pour 4 jug"),nl,move(0,Y,[(0,Y)|Z]).
move(X,Y,Z):-Y>0,\+member((X,0),Z),write("pour 3 jug"),nl,move(X,0,[(X,0)|Z]).
move(X,Y,Z):-P is X+Y,P>=4,Y>0,K is 4-X,M is Y-K,\+member((4,M),Z),write("pour from 3jug to 4jug"),nl,move(4,M,[(4,M)|Z]).
move(X,Y,Z):-P is X+Y,P>=3,X>0,K is 3-Y,M is X-K,\+member((M,3),Z),write("pour from 4jug to 3jug"),nl,move(M,3,[(M,3)|Z]).
move(X,Y,Z):-K is X+Y,K<4,Y>0,\+member((K,0),Z),write("pour from 3jug to 4jug"),nl,move(K,0,[(K,0)|Z]).
move(X,Y,Z):-K is X+Y,K<3,X>0,\+member((0,K),Z),write("pour from 4jug to 3jug"),nl,move(0,K,[(0,K)|Z]).
```

move(0,0,[(0,0)])

```
Singleton variables: [Y]
```

fill 4 jug
fill 3 jug
pour 4 jug
pour 3 jug
fill 4 jug
pour from 4jug to 3jug
pour 3 jug
pour from 4jug to 3jug
fill 4 jug
pour from 4jug to 3jug
pour 3 jug
done

true

Next | 10 | 100 | 1,000 | Stop

?- `move(0,0,[(0,0)]).`