

The problem

Missionaries and Cannibals is a problem where you initially have 3 missionaries, 3 cannibals and a boat on one side of a river. The goal is to move all cannibals and missionaries to the other side of the river given some rules:

- If any side of the river has more cannibals than missionaries, you loose. The cannibals have a meat fest!
- The boat only moves if it has one or two passengers, you can choose who's on it. The boat doesn't move without anyone on it.

Start and finish states

The problem has an initial state where everyone is on the East side of the river and we reach a solution when everyone is on the West side of the river.

```
start([3,3,0,0,east]).
```

```
goal([0,0,3,3,west]).
```

Changing states

It's not hard to see that the state changes when the boat takes passengers across the river. There are 5 possible moves:

- 2 Cannibals
- 2 Missionaries
- 1 Cannibal
- 1 Missionary
- 1 Cannibal, 1 Missionary

Valid states

A new state is valid if:

- there are missionaries present on a given side of the river, there cannot be more cannibals than missionaries.
- the number of missionaries and cannibals on any given side has to be positive.

Legal States:

```
legal(CL,ML,CR,MR) :-  
    % is this state a legal one?  
    ML>=0, CL>=0, MR>=0, CR>=0,  
    (ML>=CL ; ML=0),  
    (MR>=CR ; MR=0).
```

Possible moves:

```
move([CL,ML,CR,MR,east],[CL,ML2,CR,MR2,west]):-  
  
    % Two missionaries cross east to west.  
    MR2 is MR+2,  
    ML2 is ML-2,  
    legal(CL,ML2,CR,MR2).
```

Solution found:

```
path([CL,ML,CR,MR,B],[CL,ML,CR,MR,B],_,MovesList):-  
    output(MovesList).
```

Printing Output:

```
output([]) :- nl.  
output([[A,B] | MovesList]) :-  
    output(MovesList),  
    write(B), write(' -----> '), write(A), nl,nl.
```

Output:

```
% mandc.pl compiled 0.02 sec, 1 clauses  
true.
```

```
4 ?- mandc([3,3,0,0,E],[0,0,3,3,W]).  
[3,3,0,0,east] -----> [1,3,2,0,west]  
[1,3,2,0,west] -----> [2,3,1,0,east]  
[2,3,1,0,east] -----> [0,3,3,0,west]  
[0,3,3,0,west] -----> [1,3,2,0,east]  
[1,3,2,0,east] -----> [1,1,2,2,west]  
[1,1,2,2,west] -----> [2,2,1,1,east]  
[2,2,1,1,east] -----> [2,0,1,3,west]  
[2,0,1,3,west] -----> [3,0,0,3,east]  
[3,0,0,3,east] -----> [1,0,2,3,west]  
[1,0,2,3,west] -----> [1,1,2,2,east]  
[1,1,2,2,east] -----> [0,0,3,3,west]
```

```
true
```