# The 8-5-3 Milk Jug Problem

Charlene Chu - November 23, 2017

## Problem


**Challenge.** Draw the state graph of this problem (no spilling allowed!).

## Solution

Assuming that there is no spilling allowed, here are all of the possible moves we can make. A move {*x*, *y*} means we are pouring milk from jug *x* to *y*, where jug 1, 2, 3 represents the 8-, 5-, 3-litre jug, respectively.

```
moves = Join[Subsets[{1, 2, 3}, {2, 2}], Reverse[Subsets[{1, 2, 3}, {2, 2}], 2]]
```

```
{{1, 2}, {1, 3}, {2, 3}, {2, 1}, {3, 1}, {3, 2}}
```

Here are all of the possible states. A state {*a*, *b*, *c*} means that there are *a*, *b*, *c* litres of milk in the 8-, 5-, 3-litre jug, respectively.

```
states =
 Solve[{a + b + c == 8, 0 ≤ a ≤ 8, 0 ≤ b ≤ 5, 0 ≤ c ≤ 3}, {a, b, c}, Integers] /. (_ → α_) :> α;
states // Short
```

```
{{0, 5, 3}, {1, 4, 3}, {1, 5, 2}, {2, 3, 3}, {2, 4, 2},
 {2, 5, 1}, <<13>>, {6, 1, 1}, {6, 2, 0}, {7, 0, 1}, {7, 1, 0}, {8, 0, 0}}
```

Given a move and the current state, *f*[move, state] returns the state after making a move.

```
f[move_, state_] := Module[{x, y, max = {8, 5, 3}, m, α, β, temp},
   x = move[[1]];
   y = move[[2]];
   m = max[[y]];
   α = state[[x]];
   β = state[[y]];
   temp = state;
   If[α ≠ 0 ∧ β ≠ m, If[α + β ≥ m,
     {temp[[x]], temp[[y]]} = {α + β - m, m}, {temp[[x]], temp[[y]]} = {0, α + β}], temp = Null];
   temp
  ];
```

We can create an edge from one state to another when we make a move.

```
createEdge[move_, state_] := state → f[move, state];
```

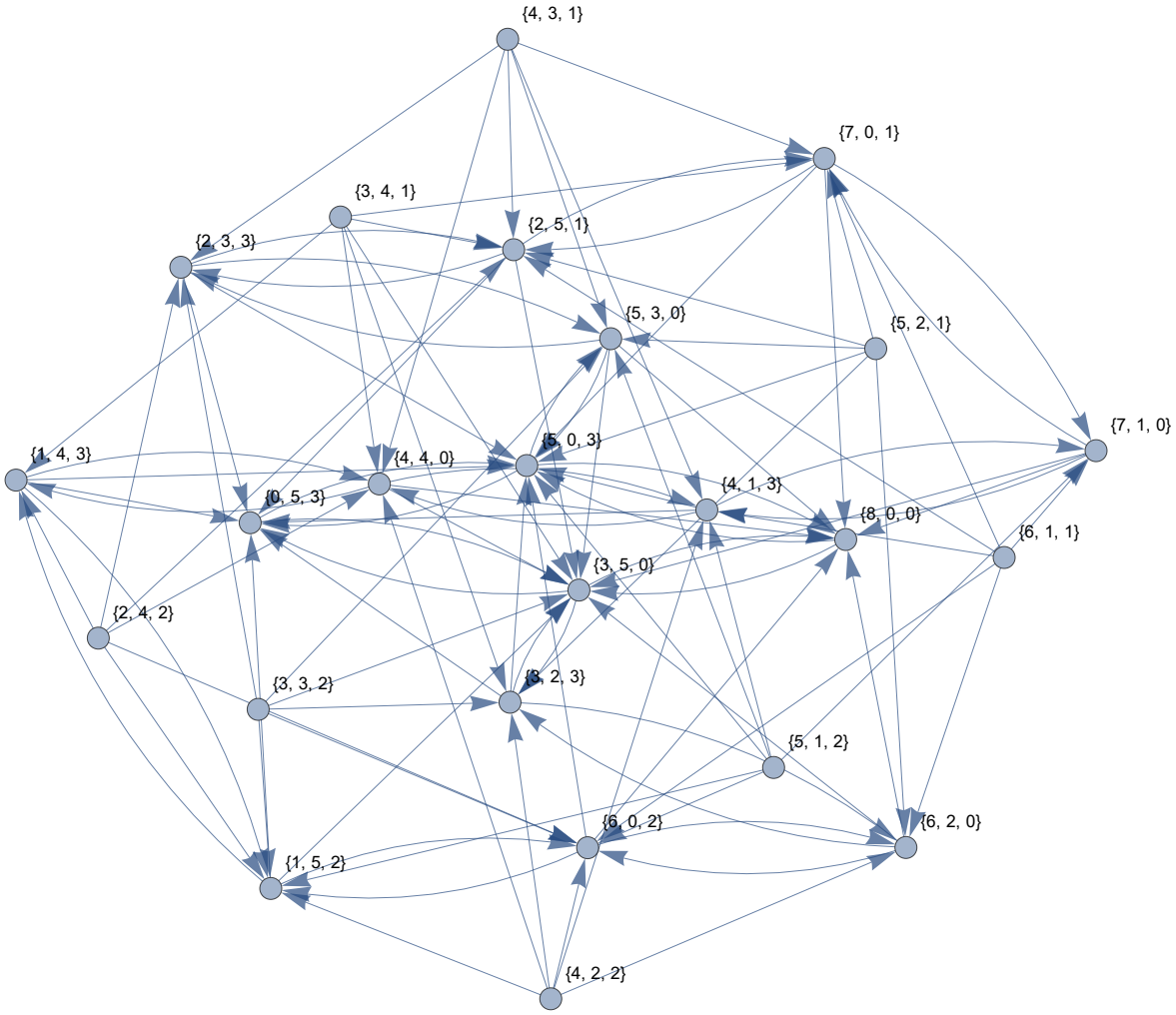
Here are all of the possible edges.

```
edges = Select[Flatten[Table[createEdge[move, state], {move, moves}, {state, states}]],
  ! MemberQ[#, Null] &];
edges // Short
```

```
{{1, 4, 3} → {0, 5, 3}, {2, 3, 3} → {0, 5, 3},
 {2, 4, 2} → {1, 5, 2}, <<101>>, {6, 1, 1} → {6, 2, 0}, {7, 0, 1} → {7, 1, 0}}
```

Here is the graph with all of the possible states.

```
graph = Graph[edges, VertexLabels → "Name", ImageSize → 620]
```




However, we are only interested in measuring 4 litres starting from the state {8, 0, 0}. So, consider the start state and the end states.

```
start = {{8, 0, 0}}
```

```
{{8, 0, 0}}
```

```
end = Select[states, MemberQ[#, 4] &]
```

```
{{1, 4, 3}, {2, 4, 2}, {3, 4, 1}, {4, 1, 3}, {4, 2, 2}, {4, 3, 1}, {4, 4, 0}}
```

Note that some of the end states cannot be achieved when we start at {8, 0, 0}. We can use FindShortestPath to determine which states we need and which states we don't.

```
paths = Join @@ Table[FindShortestPath[graph, s, e], {s, start}, {e, end}];
paths // Short
```

```
{{{8, 0, 0}, {3, 5, 0}, {3, 2, 3}, {6, 2, 0}, {6, 0, 2}, {1, 5, 2}, {1, 4, 3}},
 {}, <<3>>, {}, {{8, 0, 0}, {3, 5, 0}, <<5>>, {4, 4, 0}}}
```

Here, we can see that some paths are empty. This means that we cannot reach those end states. So,

let's remove them from the set of end states.

```
newEnd = Delete[end, Position[paths, {}]]
```

```
{{1, 4, 3}, {4, 1, 3}, {4, 4, 0}}
```

Since we know the shortest path between the start state to the end states, we can create edges between each node in the path.

```
newEdges = DeleteDuplicates[
  Flatten[Table[Table[path[[i]] → path[[i + 1]], {i, Length[path] - 1}], {path, paths}]]];
newEdges // Short
```

```
{{8, 0, 0} → {3, 5, 0}, {3, 5, 0} → {3, 2, 3},
 {3, 2, 3} → {6, 2, 0}, <<9>>, {7, 1, 0} → {4, 1, 3}, {1, 4, 3} → {4, 4, 0}}
```

Here is the graph for the 8-5-3 milk jug problem where the green node is a start state and the red nodes are end states.

```
startStyle = start /. {a_Integer, b_Integer, c_Integer} :> ({a, b, c} → Green);
endStyle = newEnd /. {a_Integer, b_Integer, c_Integer} :> ({a, b, c} → Red);
newGraph = Graph[newEdges, VertexLabels → "Name",
  VertexStyle → Join[startStyle, endStyle], ImageSize → 155]
```