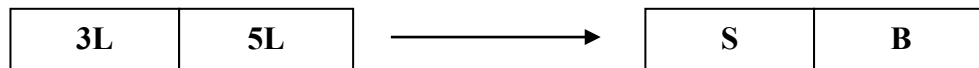


COS30019 - Introduction to Artificial Intelligence
Tutorial Problems Week 3

Task 1: Consider this problem: We have one 3 litre jug, one 5 litre jug and an unlimited supply of water. The goal is to get exactly one litre of water into either jug. Either jug can be emptied or filled, or poured into the other.

For this problem give:

(a) An appropriate data structure for representing a state would be a pair of integers (int S, int B); $3 \geq S \geq 0$, $5 \geq B \geq 0$. S represents the amount of water in the small 3 litre jug and B the amount of water in the big 5 litre jug.



(b) The initial state **(0, 0)**

(c) The goal test **(S, 1) or (1, B)**

(d) A specification of the operators which includes the preconditions (**Precond**) that must be satisfied before the operator can be used and a **new state** generated.

F3 – Fill 3L jug

 [Precond: $S < 3$]

 [Effects: $S' = 3$]

F5 – Fill 5L jug

 [Precond: $B < 5$]

 [Effects: $B' = 5$]

E3 – Empty 3L jug

 [Precond: $S > 0$]

 [Effects: $S' = 0$]

E5 – Empty 5L jug

 [Precond: $B > 0$]

 [Effects: $B' = 0$]

P35 – Pour water from 3L jug into 5L jug

 [Precond: $S > 0$ and $B < 5$]

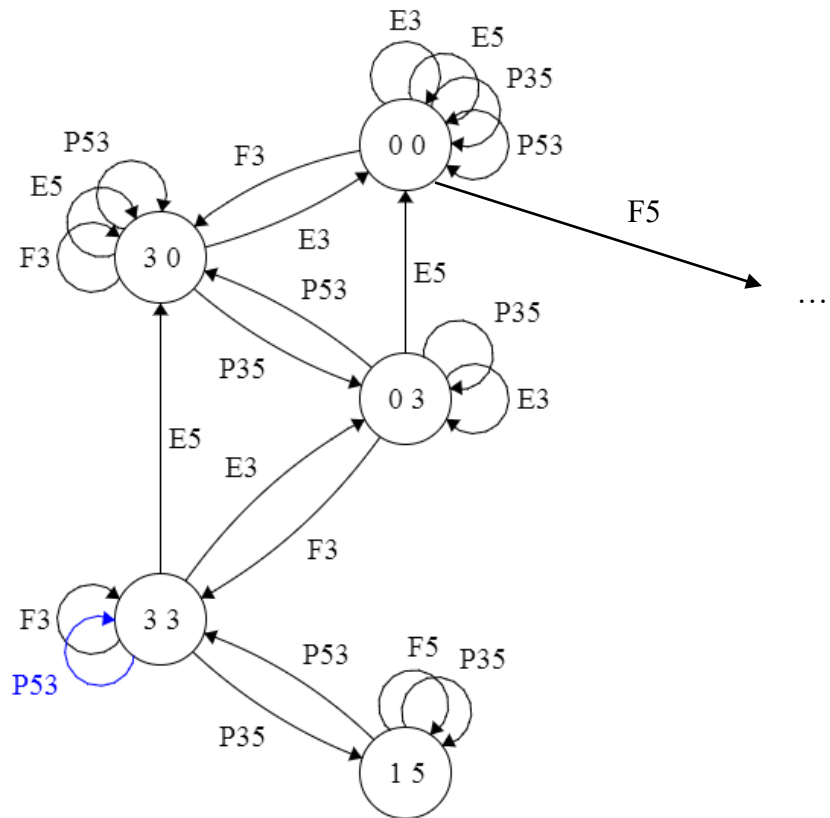
 [Effects: if ($S \geq 5-B$) then ($S' = S - (5 - B)$, $B' = 5$); else ($S' = 0$, $B' = S + B$)]

P53 – Pour from 5L jug into 3L jug

 [Precond: $B > 0$ and $S < 3$]

 [Effects: if ($B \geq 3-S$) then ($S' = 3$, $B' = B - (3 - S)$); else ($S' = S + B$, $B' = 0$)]

(e) Draw the full state space



Generally, we can also ignore the actions that do not generate a new state (see Question (d))

(f) What is the solution to the problem?

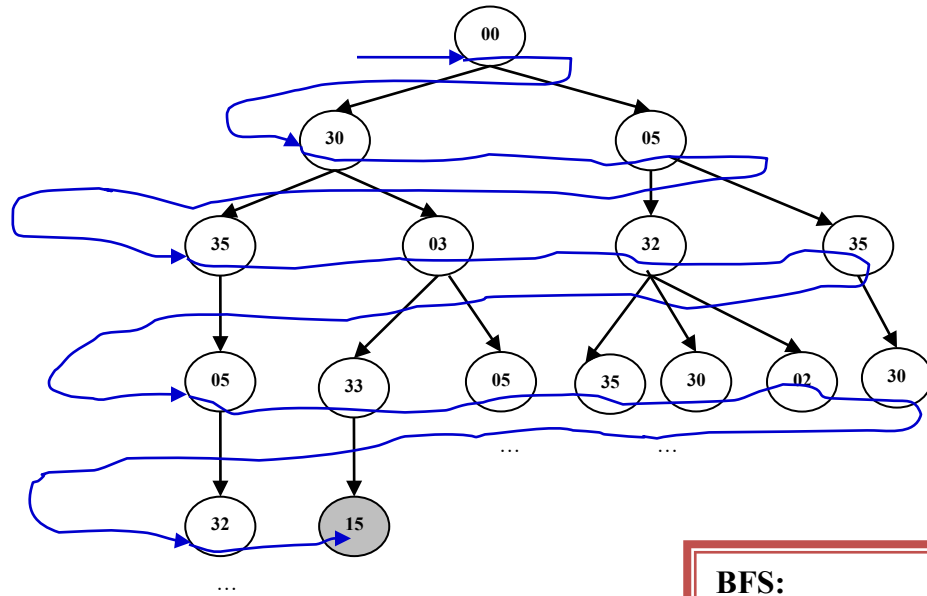
F3, P35, F3, P35 (there are other solutions, but this one is optimal)

Task 2: In the previous exercise, a representation for states and the full state space were developed. For the same problem, apply search strategies and note:

- The order in which nodes are created in memory
- The nodes that are not created in memory at all for the following search strategies:

State – (0, 0) or (0, 3), Node can contain the same state several times (repeated state or duplicate state)

- a) **Breadth first search with no checking for duplicate states**
(after that the solution is found, no node is created)

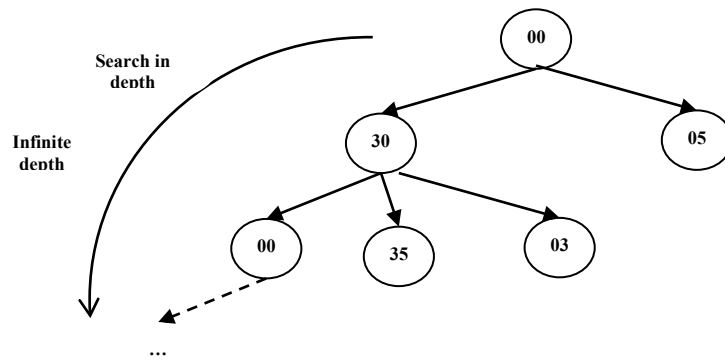


BFS:

Complete (it can scan through all the nodes exhaustively)

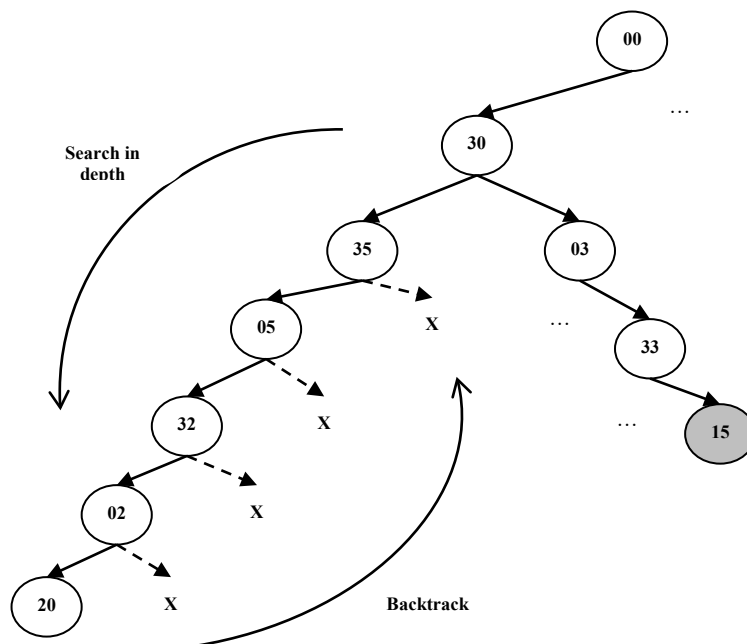
Time and memory consuming

c) Depth first search with no checking for duplicate states



Please make sure that you are able to draw a blue line (with arrows) to indicate the order of expansion for the above tree (and also the tree below).

d) Depth first search with checking for duplicate states



DFS:

Not optimal (we don't know if there are other solutions and if they are better than the found one)

Complete (if we check for repeated states)

