Topics: Problem spaces and problem solving.

Example problem space

The states are A, B, C, D, E, F, R, and S, with A being the initial state. For the purposes of this example only, we leave the goal state unspecified. The operators are as follows:

\[
\begin{align*}
\alpha_1 &: A \rightarrow B \\
\gamma_1 &: C \rightarrow A \\
\alpha_2 &: A \rightarrow C \\
\gamma_2 &: C \rightarrow F \\
\beta_1 &: B \rightarrow D \\
\delta_1 &: D \rightarrow E \\
\beta_2 &: B \rightarrow C \\
\rho_1 &: R \rightarrow S
\end{align*}
\]

Path tree (goal states not indicated, operator labels omitted):

Depth-first search:

1. Mark node 0 visited.
2. Choose the deepest visited node \( n \).
   (a) If \( n \) corresponds to a problem-space goal state, declare success and stop;
   (b) otherwise, if \( n \) corresponds to a repeated problem-space state or is childless, remove it and all its descendants;
   (c) otherwise, mark \( n \)'s least-Gorn-numbered unvisited child as visited.
3. If the tree still has nodes, repeat step 2.
4. If the entire tree has been removed, declare failure.

Breadth-first search:

1. Mark node 0 touched.
2. Choose the largest-Gorn-numbered touched node \( n \).
   (a) If \( n \) corresponds to a problem-space goal state, declare success and stop;
   (b) otherwise, if \( n \) corresponds to a repeated problem-space state or is childless, delete it and all its descendants;
   (c) otherwise, mark the least-Gorn-numbered untouched node as touched.
3. If the tree still has nodes, repeat step 2.
4. If the entire tree has been deleted, declare failure.

Note: we’re using “visited” and “removed” for DFS and “touched” and “deleted” for BFS to facilitate lecture notation.