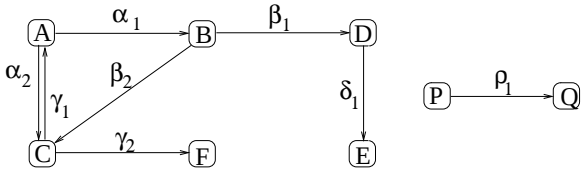
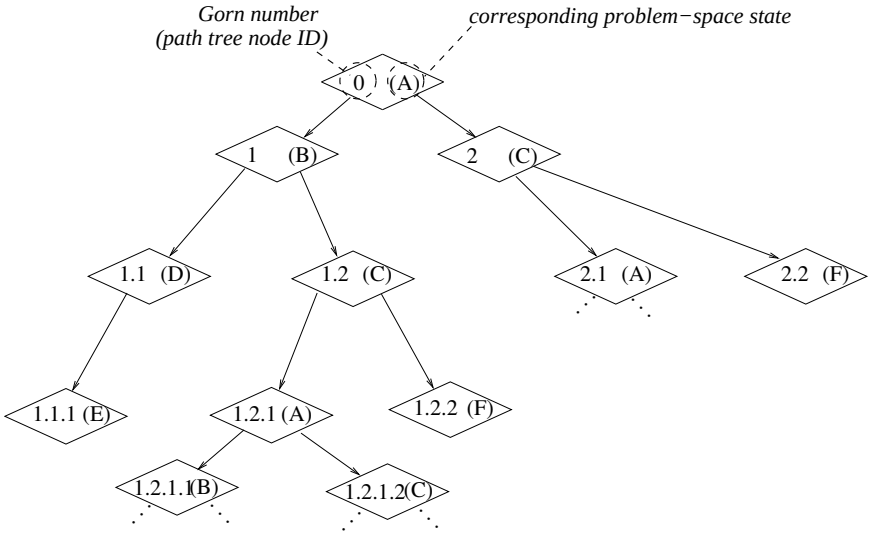
Computation, Information, and Intelligence (COMS/ENGRI/INFO/COGST 172), Fall 2005
9/5/05: **Lecture aid – Problem solving through systematic search**

Agenda: path trees, which explicitly represent all paths in a problem-space specification; depth-first and breadth-first search, which seek a solution within a (possibly infinite) path tree.
Announcements (update to office hours given last time) The regular weekly drop-in office-hours schedule kicks in this week (see "Course Staff and Weekly Office Hours") handout, with the following Labor Day (today, 9/5/05) exceptions: Prof. Lee's office hours will be 11:10-11:40 in 4152 Upson (this is a change to Friday's announcment), and there will be no office hours tonight.
Follow-ups to last time: I mis-stated the number of states in two specifications. Specification #1 has 192 states, not 240.[1] Specification #3 has 16 states, not 17.[2]

Small problem-space specification:



A is the initial state, and P is the only goal state.
The (top of the) corresponding path tree (using numerical order on action labels, which are omitted):



Trees: These consist of ≥ 1 *nodes* (a.k.a. *vertices*, singular *vertex*) and *directed edges* ("arrows") (a.k.a. *edges*) between some pairs of nodes. The *root* node has no edges into it; for every other node, there is exactly one way to get from the root to it following edges in the proper direction.
Gorn numbering of a tree: the root's Gorn number is 0. The i^{th} child (i starting from 1) of the node with Gorn number j is $j.i$, except we omit the leading "0." for convenience.

[1] For those interested in the combinatorics here (such as they are): I miscounted the number of classes meeting at nine, thus computing $5 \times 3 \times 4 \times 4 = 240$ rather than $4 \times 3 \times 4 \times 4 = 192$ schedules without conflicts.

[2] I had originally intended to make a pedagogical point by having the initial state be one of a different format ("[—]"), thus having an extra state ($1 + 2 \times 2 \times 2 \times 2$). I changed my mind but mentioned the old number.

Two search algorithms

The important differences between the two algorithms are in bold. We're using "visited" and "removed" for DFS and "touched" and "deleted" for BFS to facilitate notation.

Depth-first search:	*Breadth-first search*:
1. Mark node 0 visited (e.g., "v1").	1. Mark node 0 touched (e.g., "t1").
2. Choose the **deepest** visited (and non-removed) node n.	2. Choose the **largest-Gorn-numbered** touched (and non-deleted) node n.
(a) If n corresponds to a problem-space goal state, declare success and **stop**;	(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;	(b) otherwise, if n corresponds to a repeated problem-space state or is childless, delete it and all its descendants;
(c) otherwise, **mark n's least-Gorn-numbered unvisited child as visited.**	(c) otherwise, mark the **least-Gorn-numbered untouched node** as touched.
3. If the tree still has nodes, repeat step 2.	3. If the tree still has nodes, repeat step 2.
4. If the entire tree has been removed, declare failure.	4. If the entire tree has been deleted, declare failure.

The same path tree as on previous page