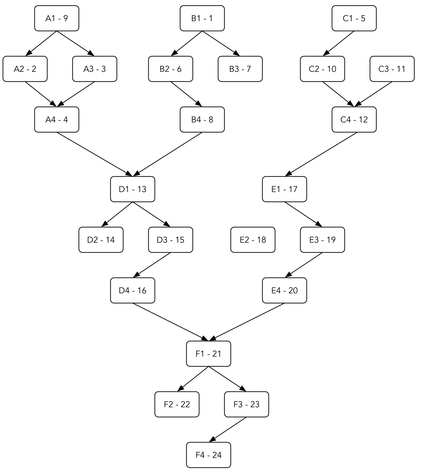
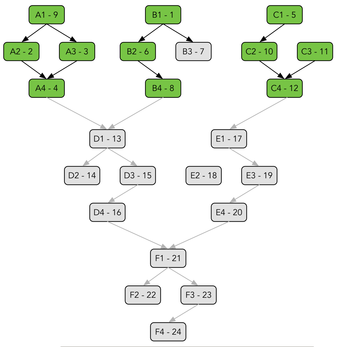
Universal Graph



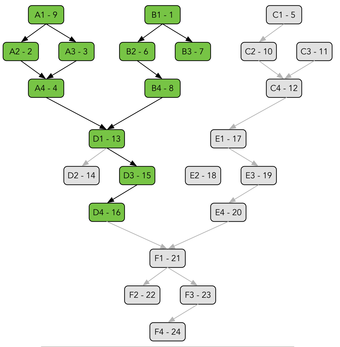
Legend

Semester 1 - instructor A



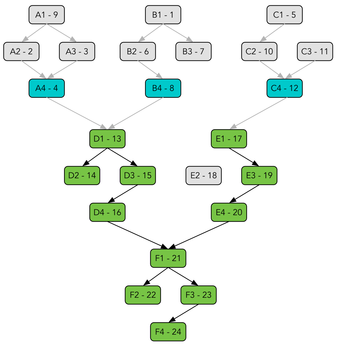
In syllabus above, all predicates are also part of the syllabus.
Sem 1 - Instructor A flow
A1-A2-A3-A4-B1-B2-B4-C1-C2-C3-C4

Semester 1 - instructor B



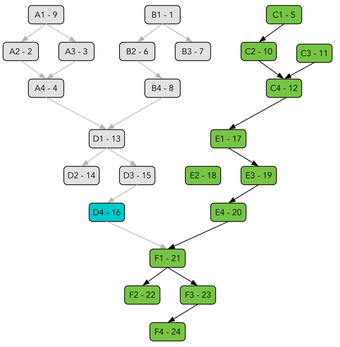
In syllabus above, all predicates are also part of the syllabus.
Sem 1 - Instructor B flow
A1-A2-A3-A4-B1-B2--B3-B4-D1-D3-D4

Semester 2 - instructor A



In syllabus above, all predicates are NOT part of the syllabus.
Ideal Sem 2 - Instructor A flow
D1-D2-D3-D4-E1-E3-E4-F1-F2-F3-F4
Special Conditions:
Fail at D1
Fail at E1

Semester 2 - instructor B



In syllabus above, all predicates are NOT part of the syllabus.
Ideal Sem 2 - Instructor B flow
C1-C2-C3-C4-E1-E2-E3-E4-F1-F2-F3-F4
Special Conditions:
Fail at F1

Please review the universal graph and sub-graph examples above

The universal graph indicates all concepts in a Topic/Course, lets say "Kinematics"
A subgraph indicates only the concepts that are part of the "syllabus" that an instructor uses for their college.

Write code to:

1) Define the Universal Graph shown above. (code should be able to take any universal graph)

2a) Enable selection of sub-graph as 'syllabus' (User Input)
OR
2b) Define sub-graph as 'syllabus'

Challenge 1
Write a simple recommender system:
Set configuration
consider_predicate_not_part_of_syllabus_depth=1 (if its 0, predicate nodes that are not part of syllabus will not be recommended)
traverse_depth_to_child_if_parent_efficacy_not_met=1 (if its 0, cannot recommend child node if parent node value=0. If value=2, recommender can go 2 levels from parent node when parent node=0)
success_criteria=0.7 (70% of syllabus nodes should be successful - have value =1)
i) The program should start (user input) at beginning node of 'syllabus' (use order number to determine the starting point only if there are more than 2 nodes at the same level)
ii) User can input 0 or 1. Recommend next node based on Input. (0-fail, 1-pass)
iii) Code should recommend predicate nodes that are not part of syllabus based on configuration setting above.
iv) Once 'success_criteria' is reached, show a message "successful" along with next recommendation.
v) Once all nodes are set to "1", show a message "You aced it" and exit the program.
vi) have a hot "key word" to exit the recommender at any time.

Challenge 2
Write a simple recommender system:
Set configuration
consider_predicate_not_part_of_syllabus_depth=1 (if its 0, predicate nodes that are not part of syllabus will not be recommended)
success_criteria=0.7 (70% of syllabus nodes should be successful)
i) accept JSON array [node_id,…] and convert to "syllabus" subgraph"
ii) accept JSON {node_id:0,…} (values can be zero or 1)
iii) Recommend that next node based on configuration setting.
iv) If entered JSON meets 'success_criteria' , show a message "successful" along with next recommendation.
v) If entered JSON has all nodes set to "1", show a message "You aced it" and exit the program.
vi) have a hot "key word" to exit the recommender at any time.