50.021 – Artificial Intelligence

Kwan Hui

Week 02: Search

[The following notes are compiled from various sources such as textbooks, lecture materials, Web resources and are shared for academic purposes only, intended for use by students registered for a specific course. In the interest of brevity, every source is not cited. The compiler of these notes gratefully acknowledges all such sources. ]

Due: 9th Feb, 11:59pm

Submission: via eDimension

**Name: Sharryl Seto (1005523)**

**Class: Cl03**

# Breadth-First Search (BFS) and Depth-First Search (DFS)

**C**

**D**

**A**

**B**

**X**

For the above graph, A is the initial state and X is the goal state. Assuming that we insert nodes in terms of lowest alphabetical order first.

**TASK:** Answer the following questions:

1. Run BFS as a graph search, and list down the following: (i) the frontier/queue at every step; and (iii) the solution (if any).
2. FIFO Queue:

* A
* AB, AC
* AC, ABD (ABC is not added as C is in frontier/ to be explored)
* ABD (ACB/ACD is not added as B and D have been explored)
* ABDX

1. Solution: ABDX
2. Run DFS as a graph search, and list down the following: (i) the frontier/queue at every step; and (iii) the solution (if any).
3. LIFO Queue:

* A
* AB, AC
* AB, ACD (ACB not added as B is in frontier/ to be explored)
* AB, ACDX

1. Solution: ACDX
2. If BFS is run as a tree search (instead of a graph search), what additional nodes will be inserted? List down 3 such nodes.

Tree search means there is no check whether state is in frontier / to be explored as trees do not contain cycles while graphs may.

Additional nodes: ABC, ACB, ACD.

1. If DFS is run as a tree search (instead of a graph search), what additional nodes will be inserted? List down 3 such nodes.

Tree search means there is no check whether state is in frontier / to be explored as trees do not contain cycles while graphs may.

Additional nodes: ACB, ACBD, ACDB.

# More BFS/DFS

**C**

**E**

**D**

**A**

**B**

**F**

**I**

**H**

**X**

For the above graph, A is the initial state and X is the goal state. Assuming that we insert nodes in terms of lowest alphabetical order first.

**TASK:** Answer the following questions:

1. Run BFS as a graph search, and list down the following: (i) the frontier/queue at every step; and (iii) the solution (if any).
2. FIFO Queue:

* A
* AB, AC
* AC, ABD
* ABD, ACE, ACF
* ACE, ACF, ABDH, ABDX

1. Solution: ABDX
2. Run DFS as a graph search, and list down the following: (i) the frontier/queue at every step; (iii) the solution (if any).
3. LIFO Queue:

* A
* AB, AC
* AB, ACE, ACF
* AB, ACE, ACFI
* AB, ACE, ACFIH
* AB, ACE, ACFIHD, ACFIHX

1. Solution: ACFIHX

# Uniform Cost Search (UCS)

**C**

**E**

**D**

**A**

**B**

**F**

**I**

**H**

**X**

*2*

*1*

*5*

*50*

*15*

*0*

*3*

*5*

*5*

*20*

*30*

*10*

*3*

*3*

For the above graph, A is the initial state and X is the goal state. The path cost is written on the edges of the graph.

**TASK:** Run UCS as a graph search, and list down the following: (i) the frontier/queue at every step (including the path cost of each node); and (iii) the solution (if any).

Expand unexpanded node with the lowest path cost g(n).

1. Queue:

* A
* AC (3), AB (5)
* AB (5), ACF (3+5=8), ACE (3+150=153)
* ACF (3+5=8), ABD (5+50=55), ACE (153)
* ACFI (8+3=11), ACFE (8+5=13) (replaced ACE as has lower path cost), ABD (55)
* ACFE (13), ACFIH (11+3=14), ABD (55)
* ACFIH (14), ACFED (13+2=15), ABD (55)
* ACFED (15), ACFIHX (14+10=24), ABD (55)
* ACFEDX (15+1=16), ACFIHX (14+10=24), ABD (55)

1. Solution: ACFEDX