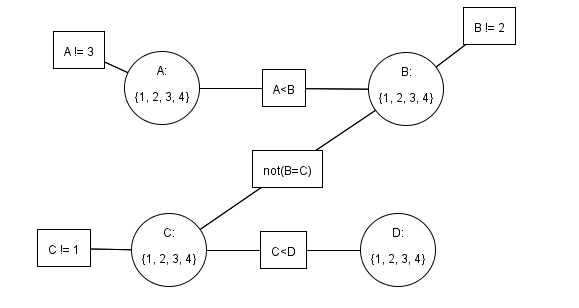
Peter Parianos

HW2

CIS467: Introduction to AI

Dr. Shirazi

Question 1:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



TDA : {(A, A !=3), (B,B!=2), (C,C!= 1), (C, C<D) , (D, C<D), (C, B != C), (B, B != C), (B,A < B), (A, A < B)}

**Step 1**: Move (A, A !=3) to notTDA and prune A =3

TDA : {(B,B!=2), (C,C!= 1), (C, C<D) , (D, C<D), (C, B != C), (B, B != C), (B,A < B), (A, A < B)}

NotTDA = {(A, A !=3)}

DA={1,2,4} DB={1,2,3,4} DC={1,2,3,4} DD={1,2,3,4}

**Step 2**: Move (B, B !=2) to notTDA and prune B =2

TDA : {(C,C!= 1), (C, C<D) , (D, C<D), (C, B != C), (B, B != C), (B,A < B), (A, A < B)}

NotTDA = {(A, A !=3),( B,B!=2)}

DA={1,2,4} DB={1,3,4} DC={1,2,3,4} DD={1,2,3,4}

**Step 3**: Move (C, C !=1) to notTDA and prune C =1

TDA : {(C, C<D) , (D, C<D), (C, B != C), (B, B != C), (B,A < B), (A, A < B)}

NotTDA = {(A, A !=3),( B,B!=2),( C,C!= 1)}

DA={1,2,4} DB={1,3,4} DC={2,3,4} DD={1,2,3,4}

**Step 4**: Move (C, C < D) to notTDA and prune C =4

TDA : { (D, C<D), (C, B != C), (B, B != C), (B,A < B), (A, A < B)}

NotTDA = { (A, A !=3),( B,B!=2),( C,C!= 1), (C, C<D) }

DA={1,2,4} DB={1,3,4} DC={2,3} DD={1,2,3,4}

**Step 5**: Move (D, C < D) to notTDA and prune D = 1,2

TDA : {(C, B != C), (B, B != C), (B,A < B), (A, A < B)}

NotTDA = { (A, A !=3),( B,B!=2),( C,C!= 1), (C, C<D), (D, C<D) }

DA={1,2,4} DB={1,3,4} DC={2,3} DD={3,4}

**Step 6**: Move (C, B != C) to notTDA and nothing is pruned

TDA: { (B,A < B), (A, A < B)}

NotTDA = { (A, A !=3),( B,B!=2),( C,C!= 1), (C, C<D), (D, C<D), (C, B != C) }

DA={1,2,4} DB={1,3,4} DC={2,3} DD={3,4}

**Step 7**: Move (B, B != C) to notTDA and nothing is pruned

TDA : { (B,A < B), (A, A < B)}

NotTDA = { (A, A !=3),( B,B!=2),( C,C!= 1), (C, C<D), (D, C<D), (C, B != C), (B, B != C) }

DA={1,2,4} DB={1,3,4} DC={2,3} DD={3,4}

**Step 8**: Move (B,A < B) to notTDA and B =1 is pruned and move (C, B != C) to TDA because for every arc related to B but not directly from B, needs t be reconsidered. If theoretical arc (Z, c’) where c’ != c, and c’ is a relation to B, this arc needs to be reconsidered.

TDA : { (A, A < B), (C, B != C) }

NotTDA = { (A, A !=3),( B,B!=2),( C,C!= 1), (C, C<D), (D, C<D), (C, B != C), (B,A < B) }

DA={1,2,4} DB={3,4} DC={2,3} DD={3,4}

**Step 9**: Move (A,A < B) to notTDA and A =4 is pruned

TDA : {(C, B != C) }

NotTDA = { (A, A !=3),( B,B!=2),( C,C!= 1), (C, C<D), (D, C<D), (C, B != C), (B,A < B),( A,A < B) }

DA={1,2} DB={3,4} DC={2,3} DD={3,4}

**Step 9**: Move (C, B != C) to notTDA and nothing is pruned

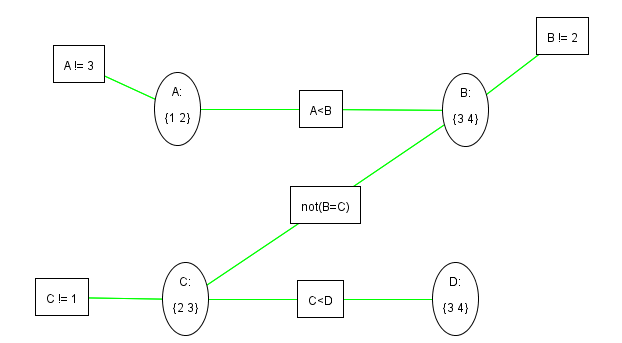
TDA:{}

notTDA : {(A, A !=3), (B,B!=2), (C,C!= 1), (C, C<D) , (D, C<D), (C, B != C), (B, B != C), (B,A < B), (A, A < B)}

DA={1,2} DB={3,4} DC={2,3} DD={3,4}

**Result of GAC:**

DA={1,2} DB={3,4} DC={2,3} DD={3,4}



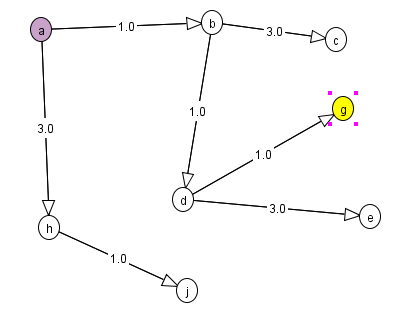
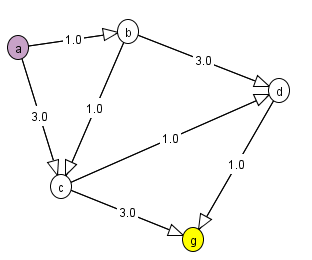
Question 2a: Question 2b:

|  |  |  |
| --- | --- | --- |
| p | q | (p **⇒** q) ꓥ (p↔q) |
| 0 | 0 | 1 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |

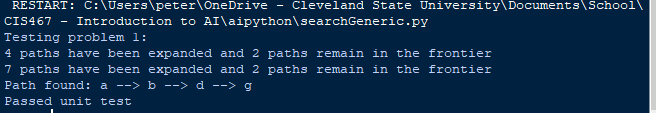
|  |  |  |  |
| --- | --- | --- | --- |
| p | q | r | (¬p **⇒** ( q ꓥ ¬r ) ­­­­­­­­) |
| 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |

**Question 3**

A Problem 1 Problem 2



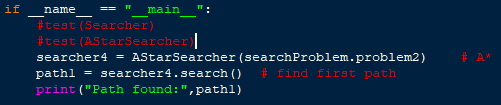
**B**, Outputs:

DFS: 

A\*:

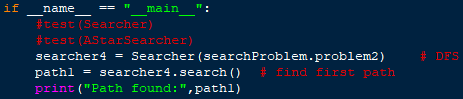


Those two tests were done with the test method. To do problem 2, I used this in searchGeneric.py:



Output: 

For DFS:

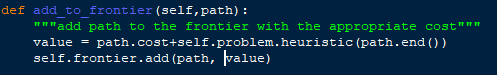


Output:



**C.**

The following function is AStarSearcher:

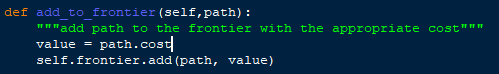


Contains a variable called value. Value is the same as  in the mathematical representation:

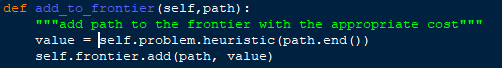
 h(p) is the heuristic function of path p, and cost(p) is the cost of the path p. We will have:

 For least-cost first searches, and only  for greedy best search.

So modify the above method for least-cost search:

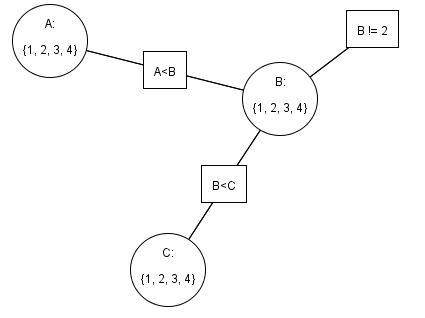
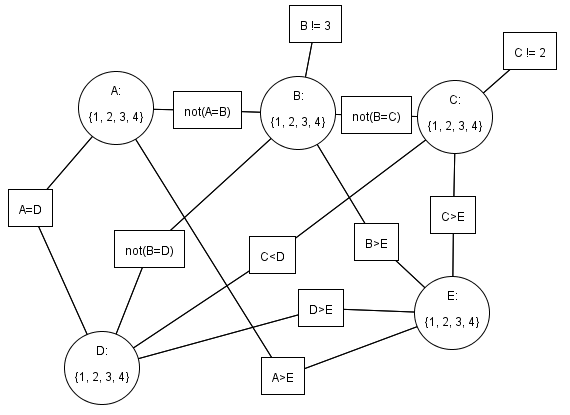


And for greedy best:

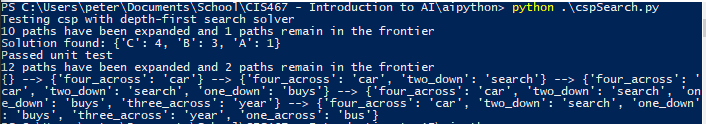


**Question 4**

CSP1: CSP2:

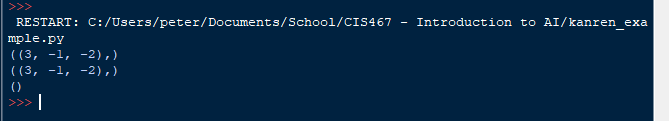


**B**



Question 5

Output:



After entering and running the code the script, the run function returns the values for *a,b* and *c* if the first expression in *la.eq\_assoccom()* matched the original function. As you can see from the output, *a* = 3, *b* = -1, and *c* = -2 matched for the original function. No values for *a, b,* and *c* could match *expression3* with *exression2.*

**Question 6**

Had to change this line:



To this line:



**Output:**

