Quiz 7

CSCE 580

March 30, 2017

Consider the state-space search problem in the figure below (Source: S. Edelkamp and S. Schroedl. *Heuristic Search: Theory and Applications*. Morgan-Kaufmann, 2012.), where heuristic estimates (the *h* function) are shown in parenthesis for each node, the start node is *a*, and the goal node is *g*.



Does Depth-First Search use *h*?

**Answer**: no.

Run Depth-First Search with multiple-path pruning by hand by filling out the table below. The order of operators is up, left, right, and then down.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Selection | OPEN | CLOSED |  |
| 1 | {} | [a] | {} |  |
| 2 | *a* | [*b,d,c*] | {*a*} |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  | Since the goal node *g* is generated, DFS stops |

**Answer**:

Note that multiple-path pruning is not equivalent to cycle check. (Although it seems to be so in this simple case.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Selection | OPEN | CLOSED |  |
| 1 | {} | [(*a,nil)*] | {} |  |
| 2 | *a* | [(*b,a),(d,a),(c,a)*] | {*a*} |  |
| 3 | *b* | *[(e,b),(f,b),(d,c)]* | *{a,b}* |  |
| 4 | *e* | *(f,d,c]* | *{a,b,e}* |  |
| 5 | *f* | *[d,c]* | *{a,b,e,f}* |  |
| 6 | *d* | *[(g,d),c]* | *{a,b,e,f,d}* | Since *g* is generated, DFS stops |

Here is pseudocode for the DFS algorithm. (Source: Judea Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1984.) For the exercise, ignore step 4; ignore the “clean up” parts of steps 4 and 7; push nodes into OPEN in the order requested; perform a cycle check.

Depth-First Search

I. Put the start node on OPEN.

2. If OPEN is empty, exit with failure; otherwise continue.

3. Remove the topmost node from OPEN and put it on CLOSED. Call this node n.

4. If the depth of n is equal to the depth bound, clean up CLOSED and go to step 2; otherwise continue.

5. Expand n, generating all of its successors. Put these successors (in no particular order) on top of OPEN and provide for each a pointer back to n.

6. If any of these successors is a goal node, exit with the solution obtained by tracing back through its pointers; otherwise continue.

7. If any of these successors is a dead end, remove it from OPEN and clean up CLOSED.

8. Go to step 2.