L03 Search

William Mak

January 05 2015

1 Search

1. We need a way to determine which nodes are reachable from current node.
2. commonly called a successor() function
3. goal test.
   (a) Check graph for loops
   (b) we don't go into any of these loops.
   (c) therefore we have a tree.
4. Our search algorithm then is straight forward
   (a) **select a node** to expand, we'll keep track of a list of active nodes.
      i. DFS
         A. **implemented:** FIFO Queue
         B. **runtime:** $O(b^s)$
      ii. BFS
         A. **implemented:** Stack + Recursion
         B. **runtime:** $O(b^m)$
      iii. UCS Uniform Cost Search.
         A. expand cheapest node first. A less special BFS.
         B. **implemented:** Priority Queue + Heap.
   (b) Check for solution.
   (c) **Add** children of newly expanded node to <u>active node list</u> in some **specified order**.

2 Heuristic Search

1. Get an estimate for how close some node is from goal.
2. Manhattan Distance, Euclidian Distance
3. come up with a good heuristic **(Good Exam Question)**

4. A* Search
    (a) Heuristic Search
    (b) Compute 'cost' of a node as $f(n) = g(n) + h(n)$
    (c) $g(n) =$ Cost to get to n from s
    (d) $h(n) =$ Heurstic estimate to goal.
5. Properties of an admissible heuristic: $h(n) \leq h^{*}(n)$ where $h * (n)$ is the true cost to get to n from S