Timothy Indrieri

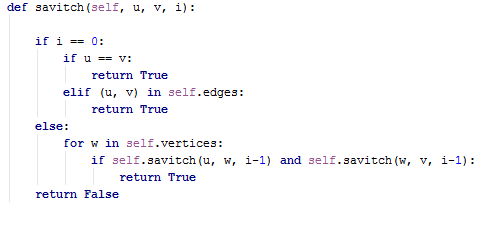
Brandon Artner

Edward Morkunas

Assignment 4

Consider two nodes, u and v. The reachability process looks to see if there is a path from u to v in a given (directed) graph G. This leads to the question, is there an efficient algorithm by which to verify this? An important solution to this question is a divide and conquer algorithm; Savitch’s algorithm which has a space complexity of O(log2 n) where n is the number of nodes in the graph G. The purpose of the algorithm is to find if node u and node v are reachable. Savitch’s algorithm uses the basic idea that whenever there is a path from u to v there exists some node in the middle of these such as node w. Since the intuition is that every path has a middle; node w suggests that the middle node will connect the two paths, forming a path from u to v with the following statement . This statement shows there must be some halfway point 2i-1 and 2i-1 evaluating this to 2i  for the total length of the path. The R represents the production path where G is the graph, u, v, are nodes, w is the midpoint and i is the length logarithmically of 2i .

**Our implementation of Savitch’s Algorithm:**



**Implementation:**

In designing Savitch’s algorithm we use a list to portray the system stack between each iteration. This list keeps track of the recursive calls that is checking for a path between the two nodes bounded by the length, i. When we find a path to the same node, we append a T onto the stack showing that there is in fact a path to that node, which is of length 0. The system stack is printed out multiple times throughout each call of the recursion. As the input graph becomes larger, more and more calls are printed out. This highlights the fact that the time efficiency of the code may not be very fast, however it does have excellent space complexity. If the desired length of the path is not 0 then we can make recursive calls to break down the problem into two parts where w becomes our midpoint. If there is a path from the start to w and a path from w to our end point then we return true. However, if there is not a path found then the algorithm will return false instead. Below are outputs for the sample run of Savitch’s algorithm. The grid size indicates the number of nodes in the grid with a graphical representation included. Following the implementation of the stack arranged for example R(1,4,2) showing u,v,i that is currently being pushed onto the stack. This implementation will pop only when there are two T values above that indicating truth values above them showing there is a valid connection between them thus the one below as well. Savitch’s algorithm helps in identifying reachability with a useful space complexity.

**Sample Run: n=1 Sample Run: n=2**

I 