Final Report

Brief Overview

The project focused on proving the correctness of breadth-first search. The result was a proof saying that if given an unweighted (directed) graph and two vertices v and w in it, if there exists a path from v to w then the implementation of BFS in this project finds a valid path from v to w and the path found is the shortest path between them. It also says that every path found by BFS is valid. By design, it only finds one path from every node w to the node it starts its search from.

The project also tried proving Dijkstra's algorithm in a manner very similar to proving BFS. The proof is almost fully complete as of the night of December 2nd and proven statements are similar to those of BFS's.

Representation and Implementation

The representation was chosen as follows. Each node in a graph is associated with an integer that identifies this particular node. Nodes are then combined with a list of nodes adjacent to it - this forms an adjacency list. Graphs themselves are lists of adjacency lists. Graphs are, in general, directed, but undirected graphs may be modelled as a special case of directed graphs.

BFS is implemented in a fairly traditional way. The implementation is similar to implementation of Dijkstra's algorithm and modelled practically as a special case of it, with the intent of reusing most of the proof for BFS for proving Dijkstra's algorithm.

The implementation of Dijkstra is also fairly traditional, the only major difference from BFS being the use of weighted edges in graphs.

Full representation and implementation details can be found in the documentation.

Proofs of BFS and Dijkstra's Algorithm

For proving BFS, we proceeded by functional induction on BFS and proved that a list of invariants hold during the execution of BFS. The complete description of those invariants can be found in the documentation.

The proof of Dijkstra's algorithm is very similar to the proof of BFS. The intent was to prove BFS and reuse almost the exact same proof for Dijkstra's algorithm, only by changing the fact that edges of the graph are now weighted.

`Admitteds` and Loopholes

We are proud to say that the BFS code contains no `admit`s and we are not aware of any loopholes. The proof of Dijkstra's algorithm contains some `admit`s, but is mostly complete.

Distribution of labor

The work on the project was mostly done in tight collaboration. Most major ideas were discussed through before coding them up. Andres was dealing more with laying out the overall structure of the proof, while Sandra was doing a lot of knocking off lemmas used in the code and substituting `admit`s with actual proofs. However, both authors of the project dealt with all kinds of necessary details every now and then.

License

The code created in this project is released to public domain (see code file).