comp3220-introduction to Artificial intelligence

Renaldo Pringle - 620067065

Andrew Rhoden - 620034032

Leon-Neil Nash - 620056991

**DOCUMENTATION**

An eight-puzzle solver was created with the Depth- First Search (DFS) algorithm to be used to solve the puzzle. The code was completed in PROLOG which does this type of search natively. Given the type of search algorithm that was specified, the program may run at a lengthy period of time, depending on the combination of the puzzle that it is currently trying to solve. This time may seem as though the system has come to a stop but it is only allocating the necessary resources to find the solution. One disadvantage of the implementation used is that the program did not take into consideration how close it was to the goal state hence the solution may come out as not the optimal solution. Due to this fact, the DFS algorithm was sub-optimal and for the reason that it did not use heuristics. Other algorithms that utilize heuristic functions such as the A\* search algorithm or more so the Breadth First search (BFS) algorithm would be more optimal.

The implementation used, as mentioned above is DFS. In our code, all the possible moves or combinations of moves are first stated as “apply\_move(State, Move, NewState).” and classified based on the direction that they can be traversed (up,down,left,right). It then takes into consideration the state that it is currently in and checks which move is possible based on the list of possible moves stated as “find\_move(State, Move)”. When a move is found, the system recursively checks until it hits the base case which is when the current state becomes the goal state.

The tests used “solve” and the “find\_solution” queries.