

Connor Fitzpatrick

Professor Rivas

Formal Languages and Computability

February 2, 2020

DFA Project Proposal- Word Search

For my project in Formal Languages and Computability, I am going to design a DFA for a program that can solve word search puzzles. To summarize, a word search contains a two dimensional array of letters, most of which being random. The puzzle includes a list of words that can be found in the 2-dimensional array. The entirety of the word will be either be to the right, left, up down, up-right, up-left, down-right, down-left of a words first letter.

One efficient way to solve such a puzzle is to look for the first letter of a word, and scan the array row by row until you come across that letter. When you find that letter, search in each possible direction to see if there is a letter that matches the second letter of the word you are searching for. Keep searching in this direction until either the searched letters match the word that is being searched for or a limit of the array has been reached (in other words you have reached the end of a row or column). Once that word is found, it will be crossed off the list. This process will be repeated for every single word until it is found. One challenge in designing a DFA for this would be that each word has a different size. One thought on resolving that matter would be to include some sort of recursive element to the DFA and possibly the program itself.

A program can certainly be designed to solve this puzzle faster than a human. For this proposed project, I think I would first attempt to store the list of words in a hash table. Each

letter of each word in the list of words to search for would represent a single node. The assortment of words would most likely be a two dimensional array, as it could be easily scanned and searched in any direction from all positions. If a letter matches one of the first letters of a desired word, then it will be stored in a temporary string, which is added to as long as a match can be maintained.