COMS4995 PFP Project Proposal: Word Hunt Solver

Allison Liu (al4130)

Fall 2022

1 Background

Word Hunt is a game where a player is given a 4x4 board of tiles corresponding to letters, and the goal is to create as many words as possible from that board. The restrictions are that consecutive letters in the word must be adjacent to each other on the board (right, left, up, down, diagonal), and the player may not use the same tile twice in the same word.

A board of $n \times n$ would have a maximum of $(n^2) \times (n^2)!$ possible sequences. The solver would blow up very easily since the basic version of Word Hunt is 4×4 board, which would result in trillions of paths.

2 Approach

Given a board and a dictionary file, the general approach to solve this game is to start at a given tile and do a depth first search in accordance to the restrictions of the game. At every depth, the word should be checked against a dictionary to verify its validity as a word. If it is a word, add it to the output list. After going through each tile and its respective paths, print out the output list, which is the list of all possible words that the board can create.

This is a very naive approach to the problem. This approach could be improved by parsing through the given dictionary and storing the words in a trie instead of a list. This way, we would now be able to determine if the path reaches a point where that word prefix does not result in any new words, and we would be able to prune the rest of the paths from that prefix. Although, depending on the size of the dictionary file, converting the file from a list to a trie may take quite a bit of time, this would still only cost constant time and space.

For this project, I could focus on doing the depth first searches of each tile in parallel. Each tile would have its own tree of paths; therefore, the searches can be done distinctly and parallelly. Each individual search could be done parallelly as well. The search path has to branch 8 times at first, and those 8 subtrees can be traversed parallelly.

Another way to parallelize this approach is to bound the searches to a subset of squares, which may help simplify the issue of revisiting a previous tile. However, this approach may change the fundamental sequential approach to the problem.

3 Objective

The final project will return a list of all possible words that can be created given a dictionary and a game board. I will first implement the approach sequentially in a basic sense. From there, I will focus on parallelizing the depth first search. Then, as time permits, I will implement the conversion of the dictionary into a trie and the feature of pruning the excessive paths on the board.

4 References

Marlow, S. (2013). Parallel and concurrent programming in Haskell. O'Reilly.