**CYKParser on GPUs**

Akshay Satam, Prateek Roy

**1. Problem description:**

We are planning to implement serial CYK parsing algorithms (mentioned in paper[1]) and also both the parallel approaches of CYK Parsing algorithm using GPUs (mentioned in paper[1]) in order to reduce the computation time. The CKY dynamic programming is used to find the most likely parse tree for a given sentence of length n in O(G n^3) time. While often ignored, the grammar constant G typically dominates the runtime in practice. Because of these costly computations, serially generating the parse-tree takes a huge amount of time.

**2. Original goal(s) / deliverable(s):**

• Project Implementation:

An implementation of the CYK serial algorithm based on the pseudocode (Fig 3 in paper[1])  
An implementation of the CYK Parser parallel algorithm on GPU, based on the approach discussed in the paper[1]  (thread based mapping, block based mapping (Fig. 6 and 7)).

• Project analysis:

Analyzing the thread based parallel, block based parallel and the serial algorithms mentioned in the paper.

**3. Progress so far:**

• We have implemented the serial algorithm for CYK parsing (Fig 3 and Fig 4 in paper[1]) using unary and binary relaxation.

• We are able to generate the best parse tree based on the scores obtained from CFG. We have used the examples of CFG from the references mentioned below.

• We can also return the best N parse trees if required but for now, we are just returning the best parse tree.

• We have gone through the techniques to parallelize the CYK parser on GPU and are trying to get deeper insights into them.

• We have used dummy scores for the grammar rules as we could not find the scores for the rules in the search we did so far. For the final implementation, we will use the scores given by the experts.

**4. Challenges so far:**

• Understanding some NLP concepts like Context Free Grammar, Top Down Parsing, Bottom-up parsing

• Gaining firm understanding of the serial CYK parsing algorithm (Fig 3 and Fig 4 in paper[1])

• Coming up with a structure to maintain scores for grammar rules in CYK algorithm

• Efficiently calculating the scores of the parse trees generated by the CYK algorithm

**5. References (if any)**

https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/37628.pdf[1]

http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/pcfgs.pdf

http://www.cse.unsw.edu.au/~cs9414/Intro/notes/nlp/grampars.html

https://gist.github.com/xiaohan2012/b9e3ab0ac5d23362bf33