CKY Algorithm

The **CKY algorithm** (Cocke–Kasami–Younger) is a well-known **bottom-up parsing algorithm** used in Natural Language Processing (NLP) to determine whether a given sentence can be generated by a **context-free grammar (CFG)**, and to build a **parse tree** for that sentence if possible.

Key Features of the CKY Algorithm:

1. **Applies to CNF (Chomsky Normal Form)**:
    - The CFG must be transformed into CNF, where all production rules are of the form:
        - A → BC (two non-terminals)
        - A → a (a single terminal)
2. **Dynamic Programming Approach**:
    - CKY uses a 2D **parse table** (or matrix) to store intermediate results.
    - Each cell `(i, j)` of the table contains the set of non-terminals that can derive the substring of the input sentence from position `i` to `j`.
3. **Bottom-up Parsing**:
    - It starts with the smallest substrings (single words) and builds up to longer substrings.

How CKY Algorithm Works:

Input:

- A sentence (sequence of tokens)
- A CFG in Chomsky Normal Form

Steps:

1. Initialize a table of size `n x n`, where `n` is the number of words in the sentence.
2. For each word in the input sentence, find all non-terminals that generate it and fill the diagonal of the table.
3. For increasing spans of length (from 2 to n):
    - For every possible partition of the span:
        - Check if any production `A → B C` exists such that `B` is in the left span and `C` is in the right span.
        - If yes, add `A` to the current cell.
4. After filling the table, check if the start symbol (`S`) exists in the top-right cell (cell covering the whole sentence).

Applications in NLP:

- **Syntactic Parsing**: To check the grammatical correctness of a sentence.
- **Syntax-based Machine Translation**: In phrase-based translation models.
- **Speech Recognition**: For decoding grammatical structures.

Example Use Case:

Given a sentence: `"the cat chases the dog"`
And a grammar in CNF, CKY will determine whether this sentence can be generated by the grammar and also help build the parse tree.

Introduction

https://www.youtube.com/watch?v=SFQ-owZaU_s&t=776s

Numerical-1

https://www.youtube.com/watch?v=kTklFRpbjRU&t=33s

Numerical 2-

https://www.youtube.com/watch?v=2fiDAk4qC9o