ATC-Assignment 3

In this assignment, you need to implement a tool to determine **Context-Free Language (CFL)** reachability in a given graph.

Problem Statement:

Given a directed graph G = (V, E) and a context-free grammar (CFG) in Chomsky normal form, the tool should determine whether there is a path from a given source node to a target node such that the sequence of edge labels along the path forms a string derivable from the CFG. We will refer to this tool as CFL-Reachability Solver.

CFL-Reachability Solver should process the graph and the grammar to determine reachability based on the context-free language defined by the CFG.

Input Details

The input consists of:

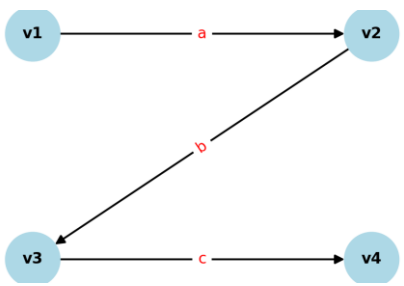
1. A directed graph
2. A context-free grammar (CFG) in Chomsky Normal Form (CNF), defining the language of valid paths.
3. Source node and a target node.

The input is read from a file Input.txt. A sample input file is:

1 # number of inputs

S -> A B; A -> a; B -> BC| b; C -> c **# CFG**

V1V2:a V2V3:b V3V4:c **#Graph**



V1 **#Source node**

V4 **#Target node**

Output Details

For each input, the tool should output YES if there exists a path from source node to target node such that the sequence of edge labels belongs to the language of the CFG; otherwise, output NO.

A sample output file for the above input might be:

YES

Code and Implementation Details

Your implementation should primarily be in main.py and helpers.py, specifically **TODO part** where it is available.

Evaluation

The repository has a file named “Output.txt” this file will contain the expected decisions on input test strings in “Input.txt”. Evaluation is based on the number of matches with the decisions in “Output.txt”. Evaluation also considers the quality of the submitted code in terms of readability of the code with proper variable names and comments as needed.

Submission Details

Each team for this assignment must comprise **exactly two students**, and only one submission per team is allowed.

Submit only the following files:

- Main.py and helpers.py
- Any additional files required for implementation.

"Upload the files as a ZIP file named `SRN1_SRN2_A3.zip`, where `SRN1` and `SRN2` are the last 5 digits of SRN of team members. Ensure that all required files are included in the ZIP before submission."

Setup

Use python programming language to implement the above tool.

Please reach out in case of any issues with implementation.