



THE UNIVERSITY OF ZAMBIA

SCHOOL OF NATURAL SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

**COURSE : PROGRAMMING LANGUEGES AND
PARADIGMS**

LECTURER : DR. MUKALEBAIKALEBAI

GROUP ASSINGMENT

GROUP nine (9) MEMBERS :

1. FIDELIS MUSAMBA
2. CHELA MAIVUNE
3. COSSAM SOKO
4. MUSENGAH NYONDO
5. BLESSINGS MWANDIRA
6. MWAMBA MWILA

Q. Working with Context Free Grammars

Context free grammar is a formal grammar that is used to generate all the possible patterns of a string in a given formal language. Languages such as English are informal due to that they have many other features that context free grammar can not describe. Despite the fact that context free grammar can not describe the context of natural languages, they are still able to define the syntax and structure of these languages. It was due to this reason that context free grammars were introduced in the first place. Before you go straight into writing the code and the grammar rules you need to know some of the terms that are used in context free grammars. Three of the key ones you need to know is Terminal, non-terminal and start symbol. Terminal refers to the actual content and of the final sentence. These can letters or words, depending on the context in which one is used as the building blocks of the sentence. Non-terminal act as a sub language within the language defined by the grammar. These are also referred to as variables. We use these to create noun phrases, verb phrases, nouns, verbs, adjectives and so on. Start symbol is a non-terminal noun that represents the initial string that will be generated by the grammar

While writing grammar rule we start by defining the set of terminals and the start state. We have to keep in mind that start symbols are not part of the set of terminal, but instead they belong to the set of non-terminals. Bellow is an example of a grammar rules for a specified set of words.

T: ("Monkey", "the", "ate", "bananas")

S: Start State.

From the above data we can design the following rules

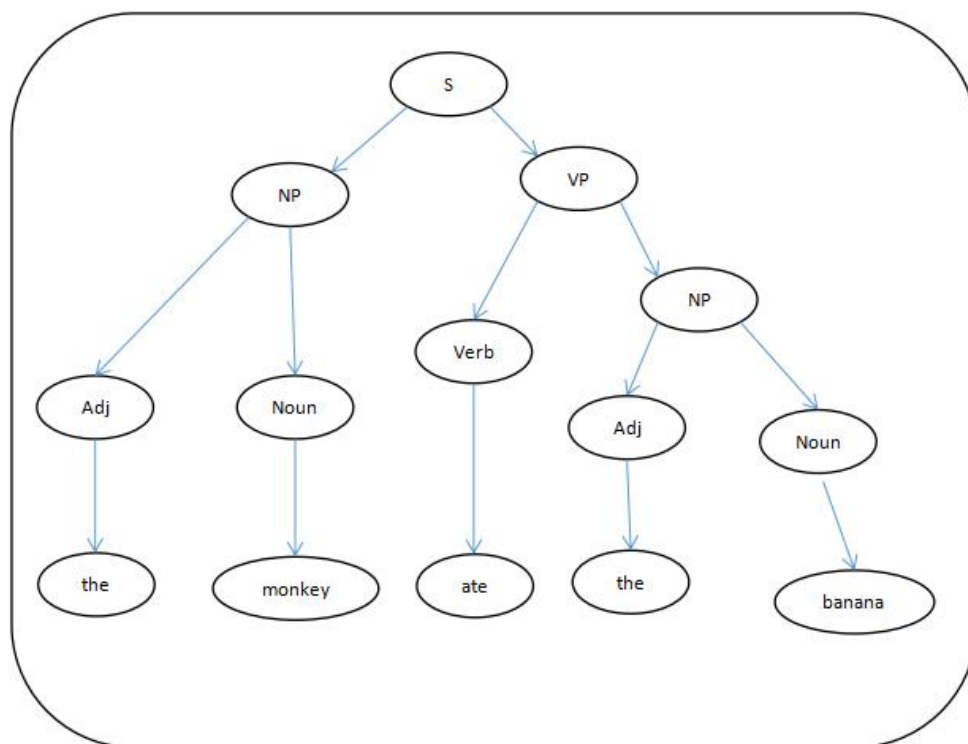
- $S \rightarrow \text{nounPhrase(NP) verbPhrase(VP)}$
- $NP \rightarrow \text{adj nounPhrase} \mid \text{adj noun}$
- $VP \rightarrow \text{verb nounPhrase}$

Adj \rightarrow the

Noun \rightarrow Monkey | banana

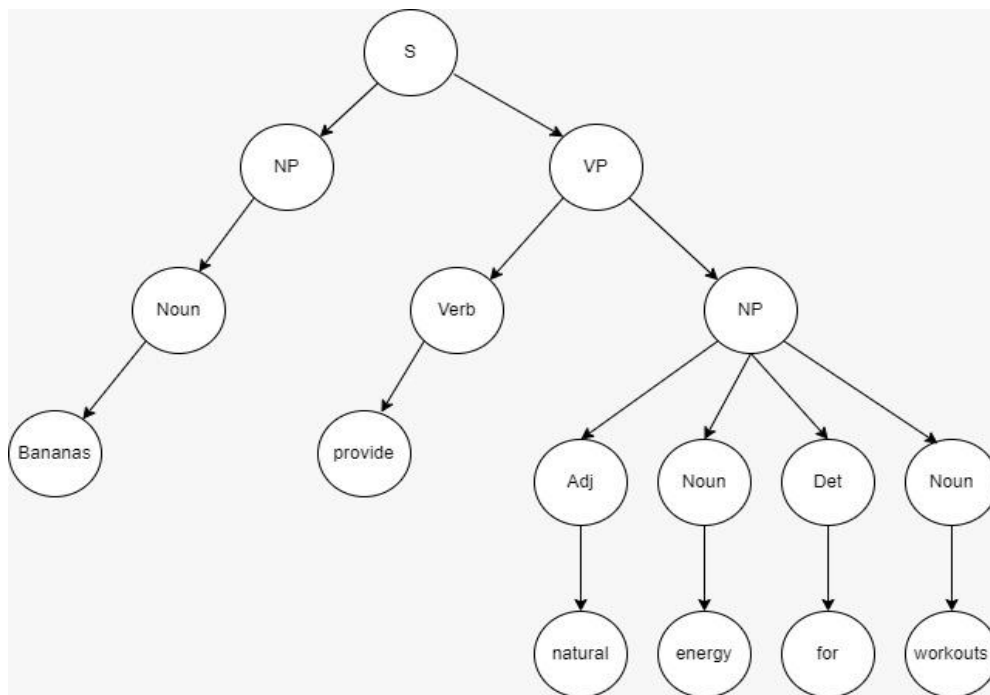
Verb --> ate

A better way to understand the above rules is to visualize them in the form of a tree structure. The tree will have S as the root with NP and VP as the child roots. From there we will proceed in the same way with NP and VP with the terminals been the leafs of the tree



Now from the above information we as a group we were tasked to make our own grammar rules and a code which will implement the rules and generate random sentences when run. The sentences do not fully make sense but we were able to try and make them as sensible as possible to a reader. We were able to come up with three grammar rules and tree structures of our own. The rules are in the code which is in the our Git Hub repository named " CSC-3301-GROUP-09-TeamManagementSystem ", branch titled "Assignment 2" . As for the tree structures they are shown bellow in this PDF document. Bellow are the three tree structures

From the above tree structures we were able to generate some random sentences which are one can generate from our interface created using an HTML created. Below we have screen shots of our interface and and some of the random sentences which we were able to generate.



RULES

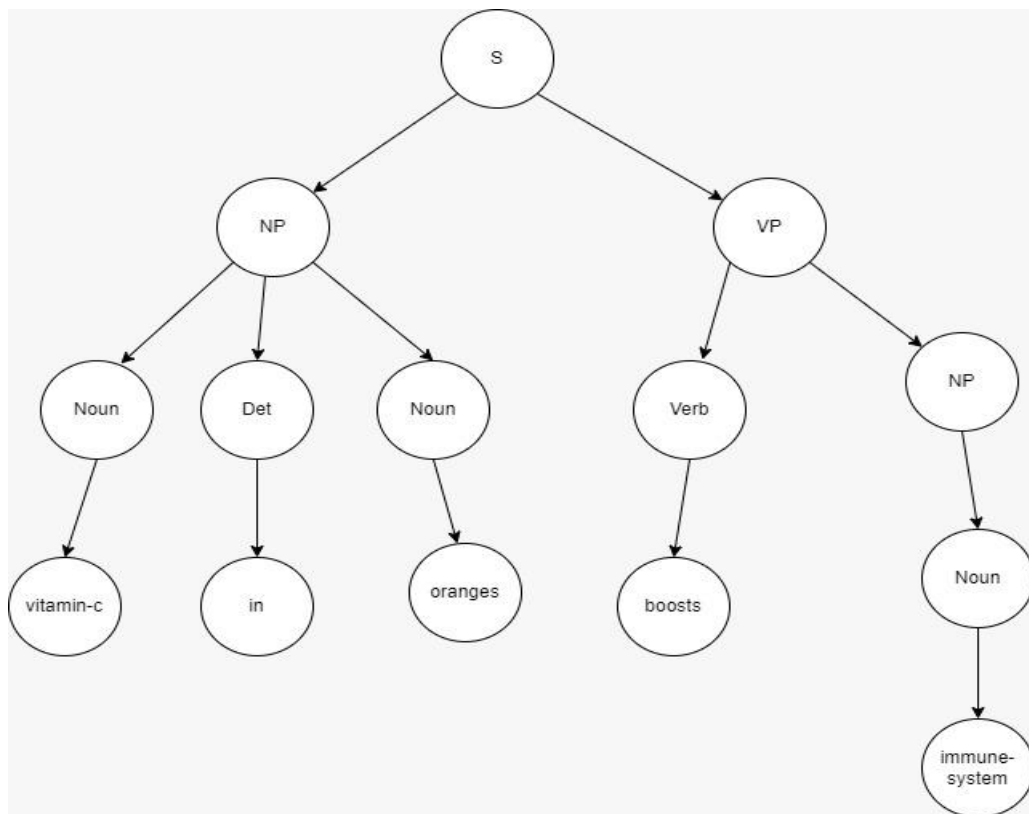
- S --> NP VP
- NP --> noun
- VP --> verb NP
- NP --> adj noun det noun

Noun --> Bananas , energy , workouts

Verb --> provide

Adj --> natural

Det --> for



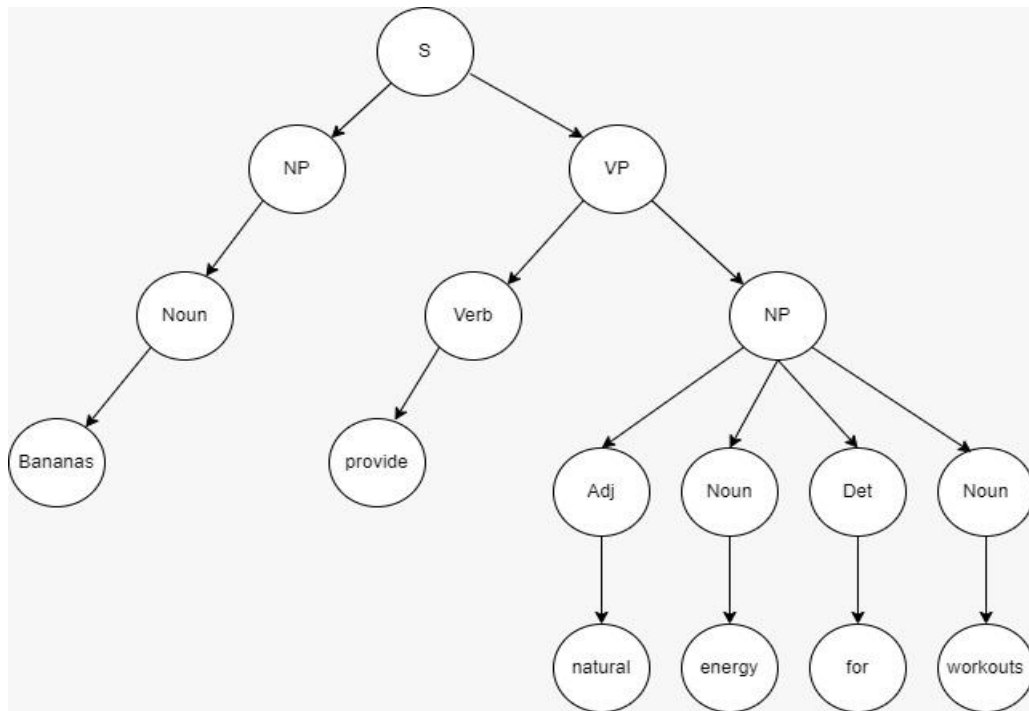
RULES:

- $S \rightarrow NP VP$
- $NP \rightarrow noun det noun$
- $VP \rightarrow verb NP$
- $NP \rightarrow noun$

Noun \rightarrow vitamin-c , oranges , immune system

Det \rightarrow in

Verb \rightarrow boosts



RULES:

- S --> NP VP
- NP --> noun
- VP --> verb NP
- NP --> adj noun det noun

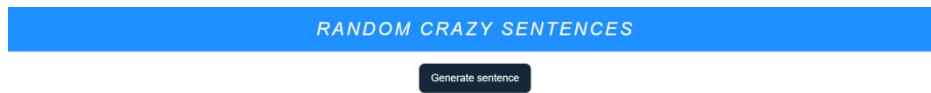
Noun --> banana , energy , workouts

Verb --> provide

Adj --> natural

Det --> for

For the user interface, we went for a simple design with has a header, the button to generate the sentences and the name for the group members put at the bottom of the page.



The figure above is a screenshot of our created user interface to our program. As for the screenshot bellow is of the screenshot of some of the sentences we are able to generate.

