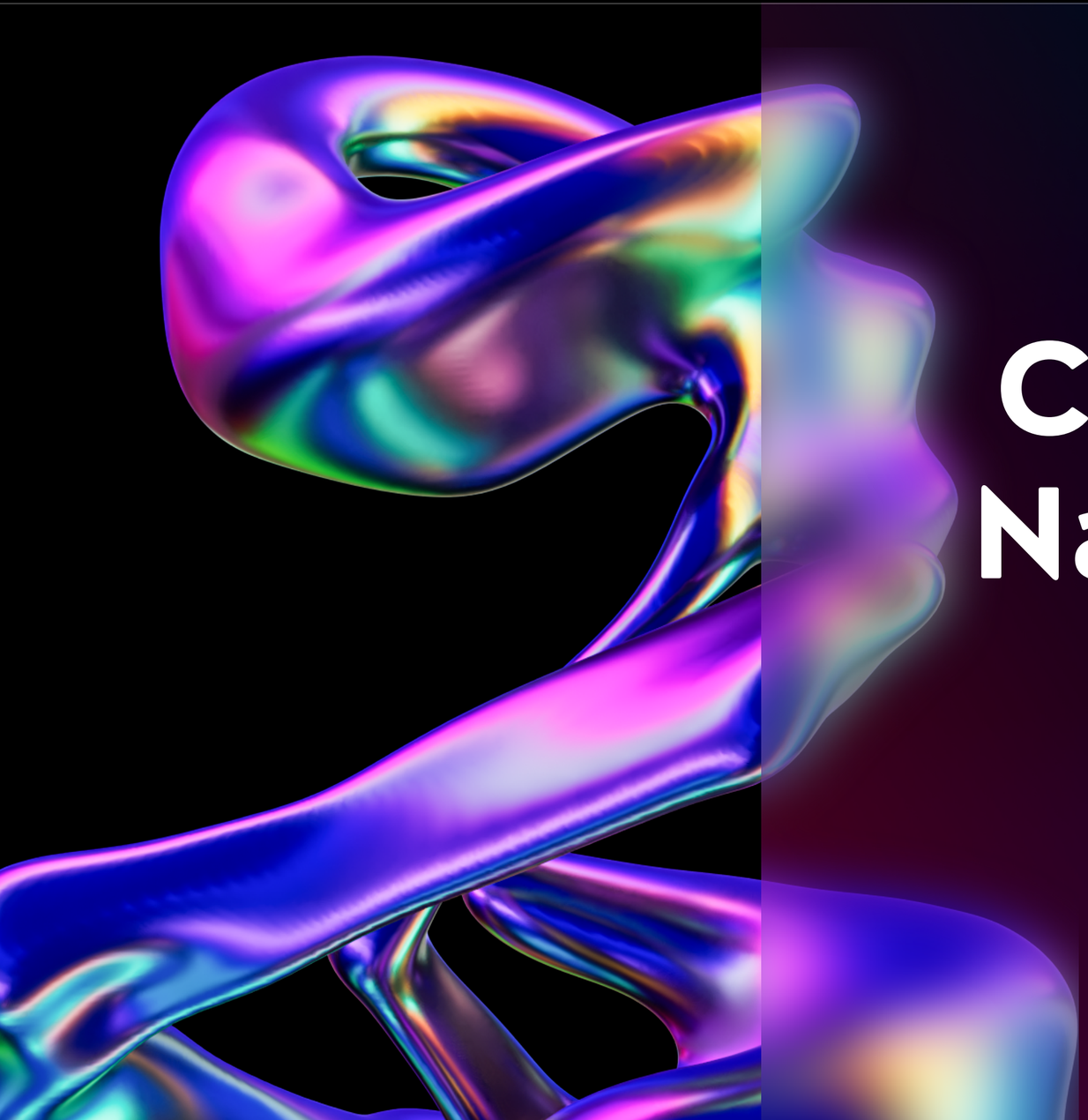


Linguistic Compilation and Natural Language Processing

Dr. Joseph Costantine

Project Objective

Implement a Lexical and Syntax analyzer

Implement a semantic analyzer

Natural Language Processing

Lexical

Noun	Adverb
Adjective	Pronoun
Verb	Determinant / Adposition

```cpp
#include <iostream>
#include <map>
#include <string>

using namespace std;

class Dictionary
{
public:
    map<string, string> AdjectiveDictionary;
    map<string, string> DeterminerDictionary;
    map<string, string> PronounDictionary;
    map<string, string> AdpositionDictionary;
    map<string, string> VerbDictionary;
    map<string, string> NounDictionary;
    bool isAdjective(string);
    bool isDeterminer(string);
    bool isPronoun(string);
    bool isVerb(string);
    bool isNoun(string);
    bool isAdposition(string);
    Dictionary()
    {
        AdjectiveDictionary["adorable"] = "Adjective";
        AdjectiveDictionary["adventurous"] = "Adjective";
        AdjectiveDictionary["aggressive"] = "Adjective";
        AdjectiveDictionary["agreeable"] = "Adjective";
```

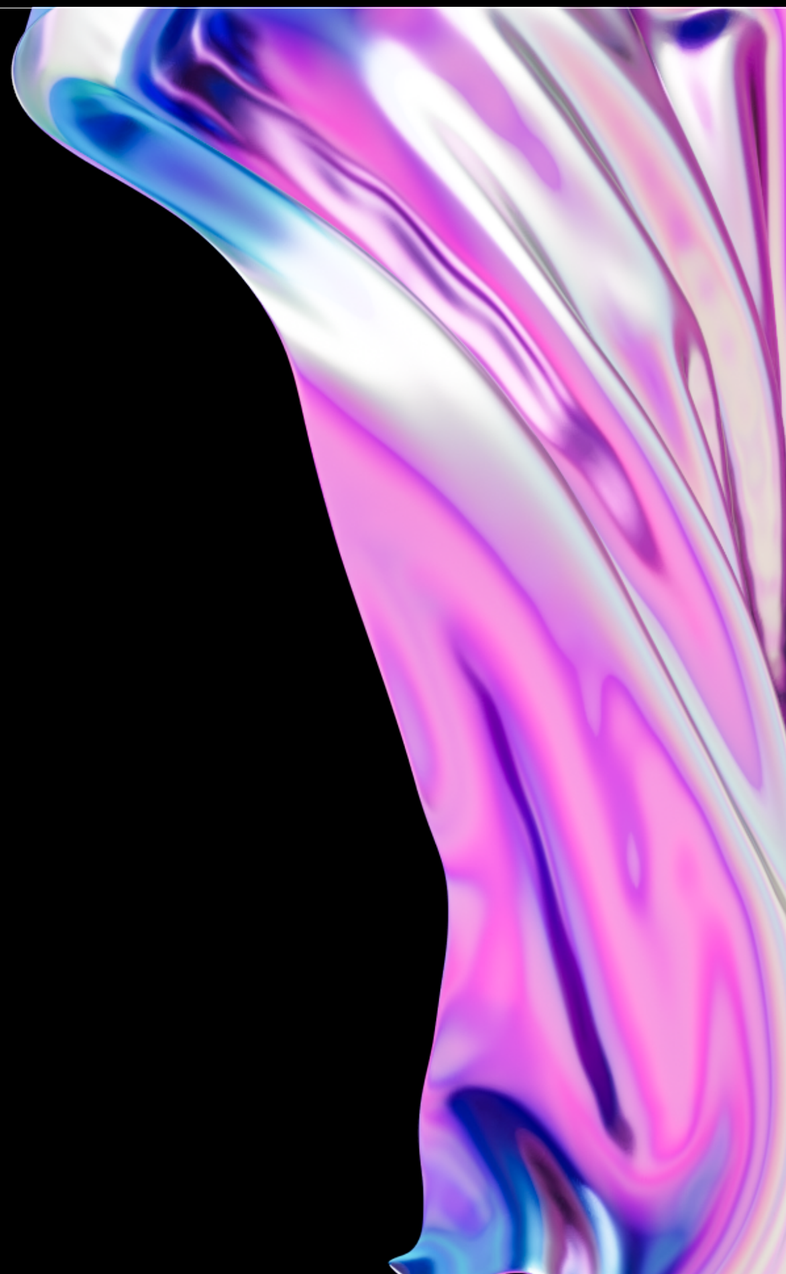
The Grammar ,
Syntax - Phrase Structure Rules

S -> NP VP

NP -> (Det) (Adj) N (PP)

PP -> P NP

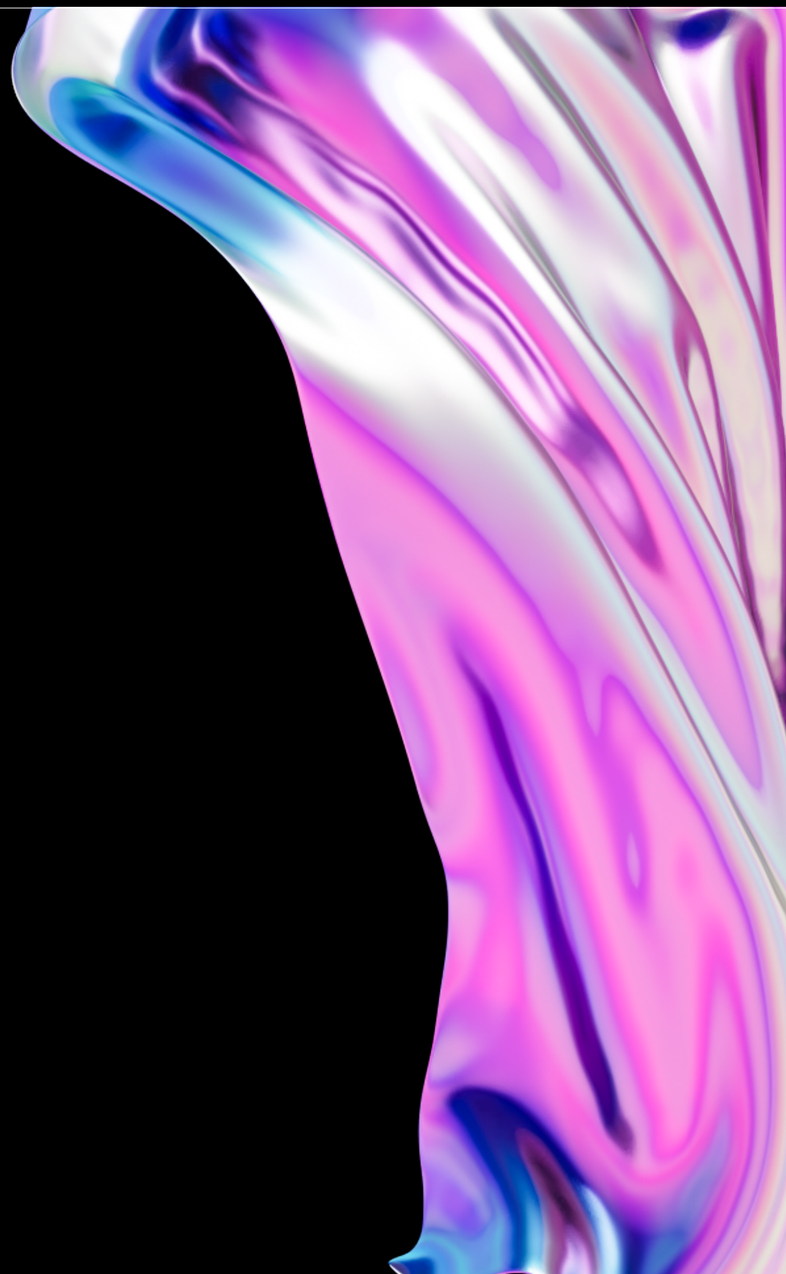
VP -> V (NP) (PP) (Adv)



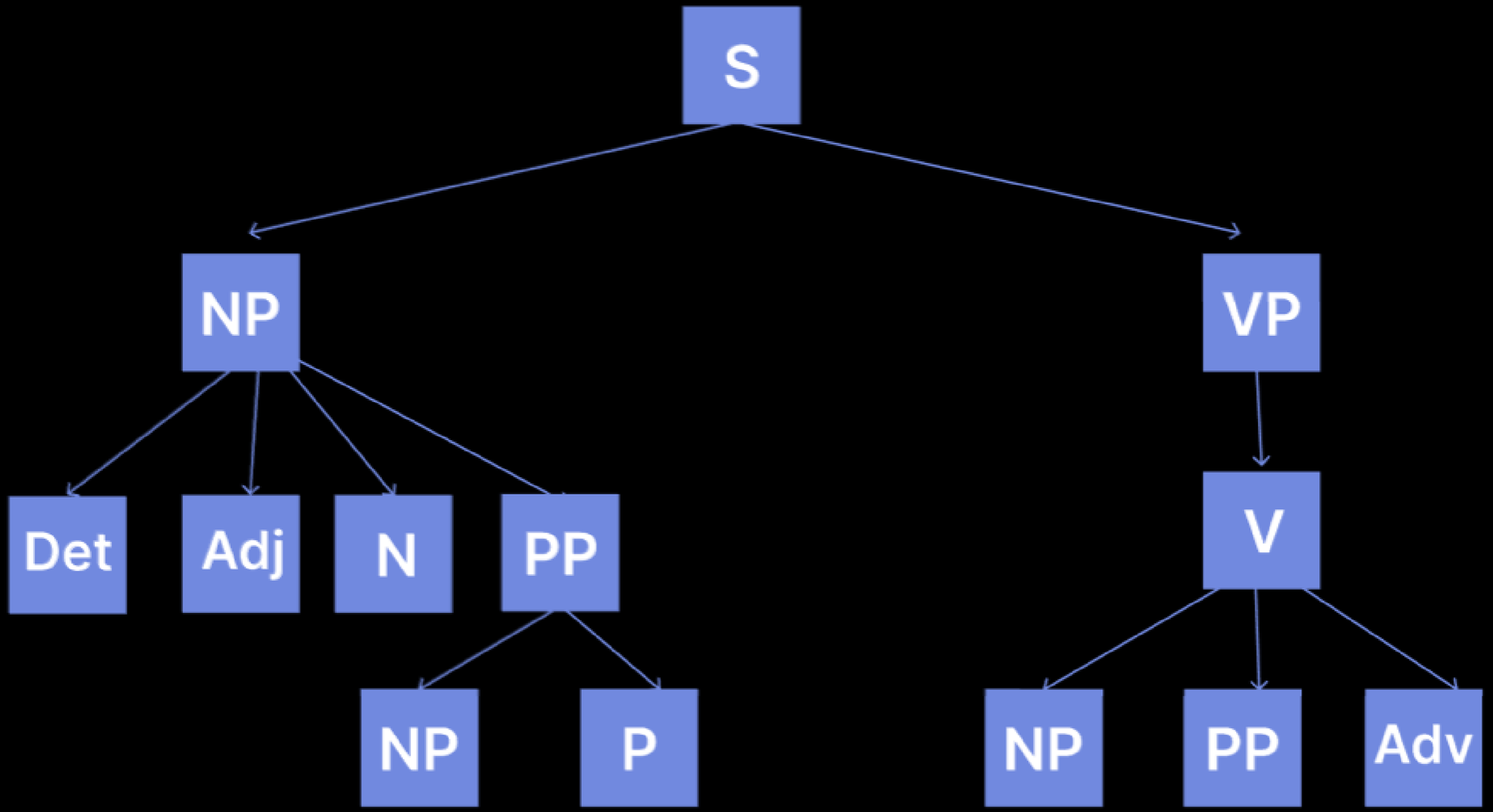
Solve the left factoring

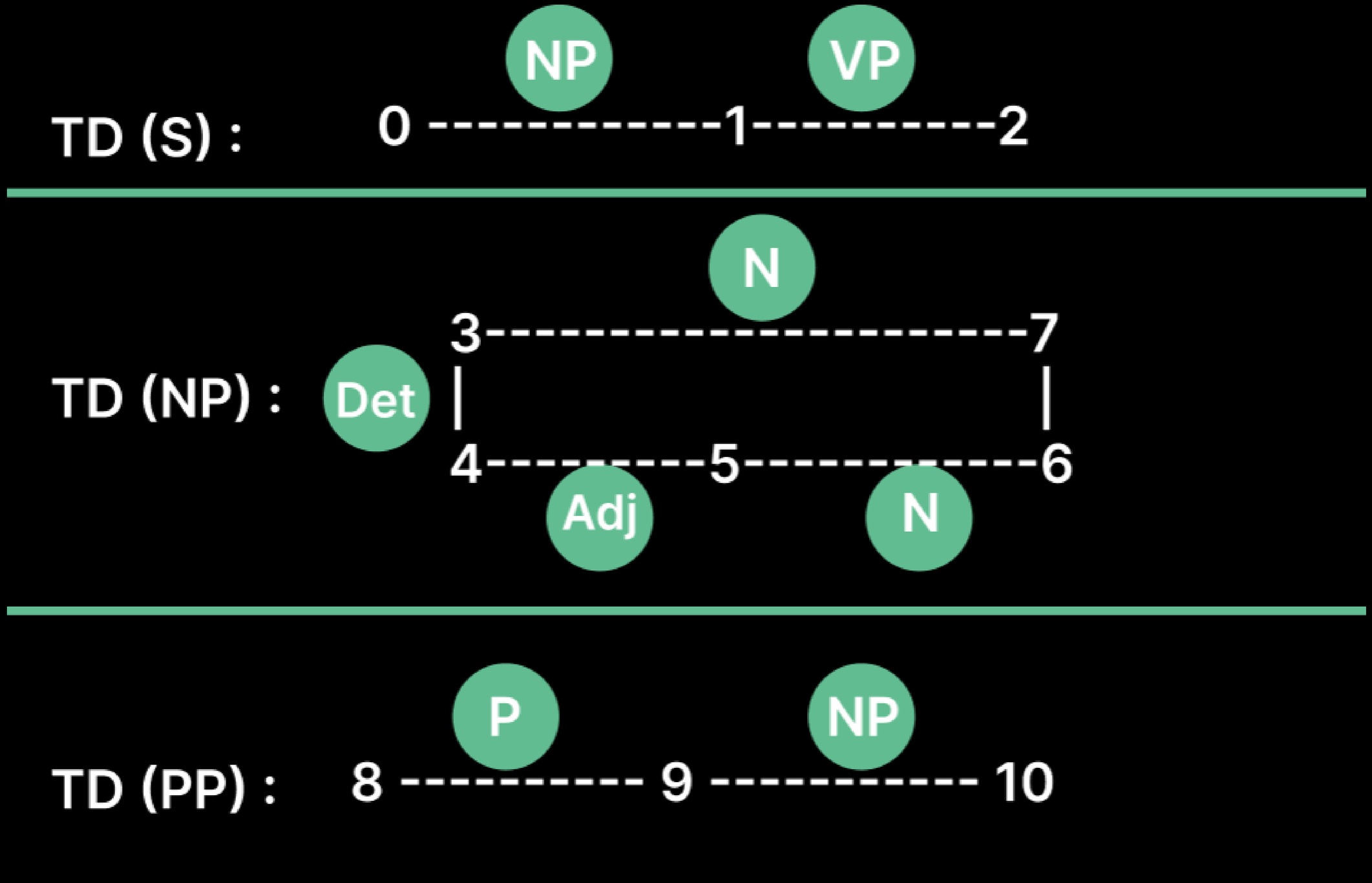
S -> NP VP
NP -> N
NP -> Det Adj N PP
PP -> P NP
VP -> V
V -> V NP PP Adv

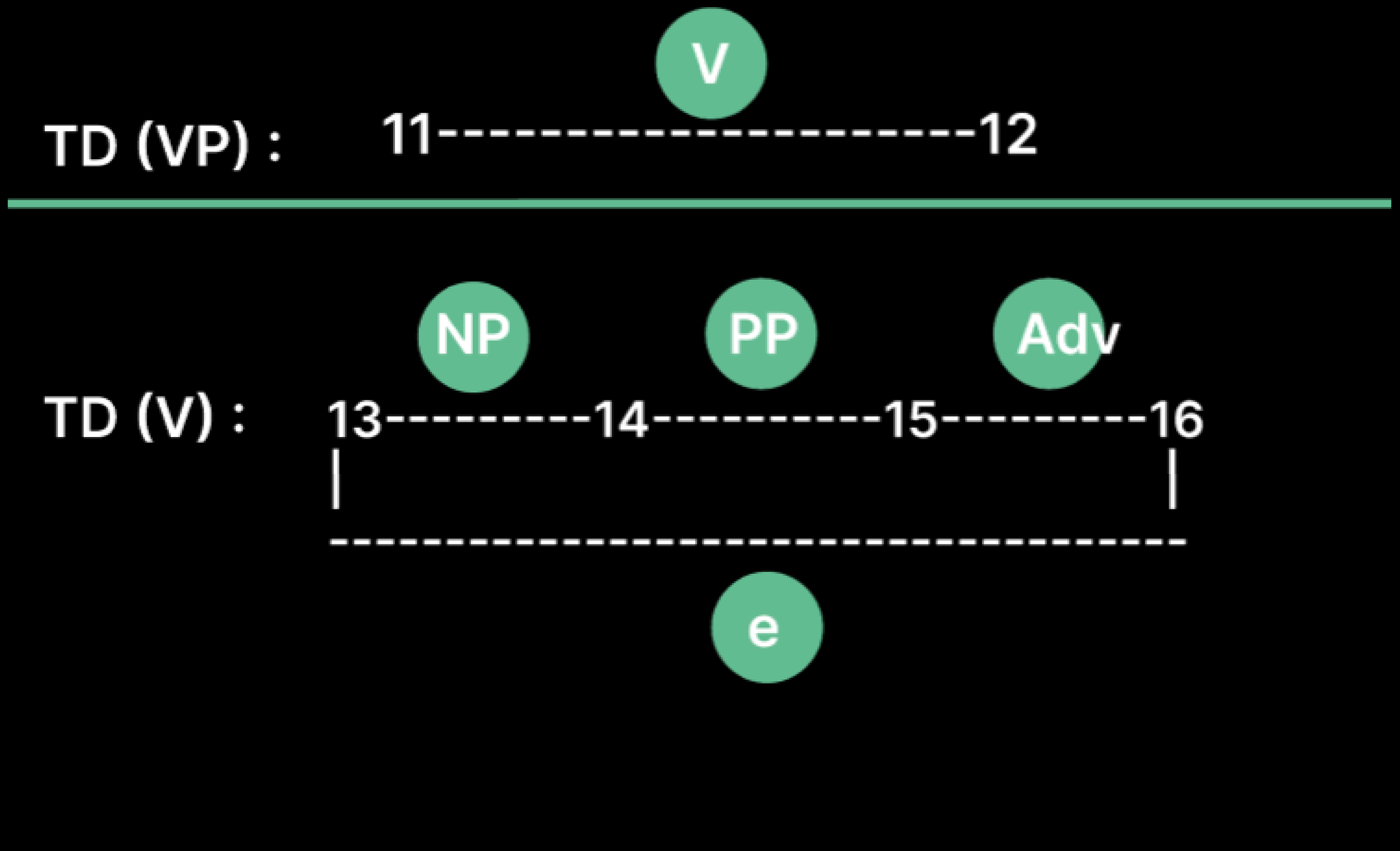
S -> NP VP
NP -> N
NP -> Det Adj N PP
PP -> P NP
VP -> V
V -> NP PP Adv



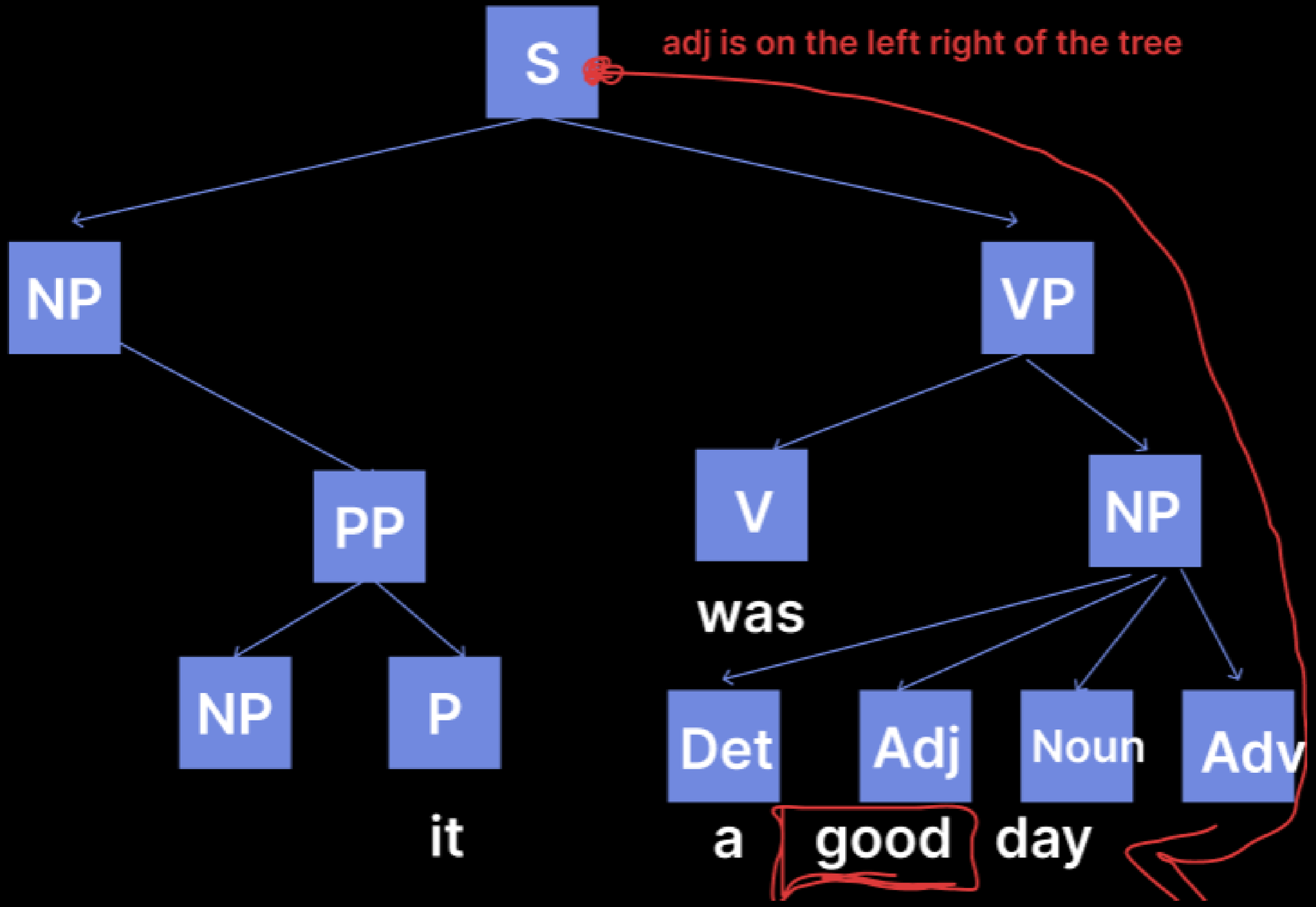
Derivative Tree




NP
VP
TD (S) :
0
1
2
N
3
7
TD (NP) :
Det
4
5
6
Adj
N
P
NP
TD (PP) :
8
9
10


V
TD (VP) :
11----------12
NP
PP
Adv
TD (V) :
13---------14---------15---------16
e

Extract the adjective for nlp


S
adj is on the left right of the tree
NP
VP
PP
V
NP
was
NP
P
Det
Adj
Noun
Adv
it
a
good
day