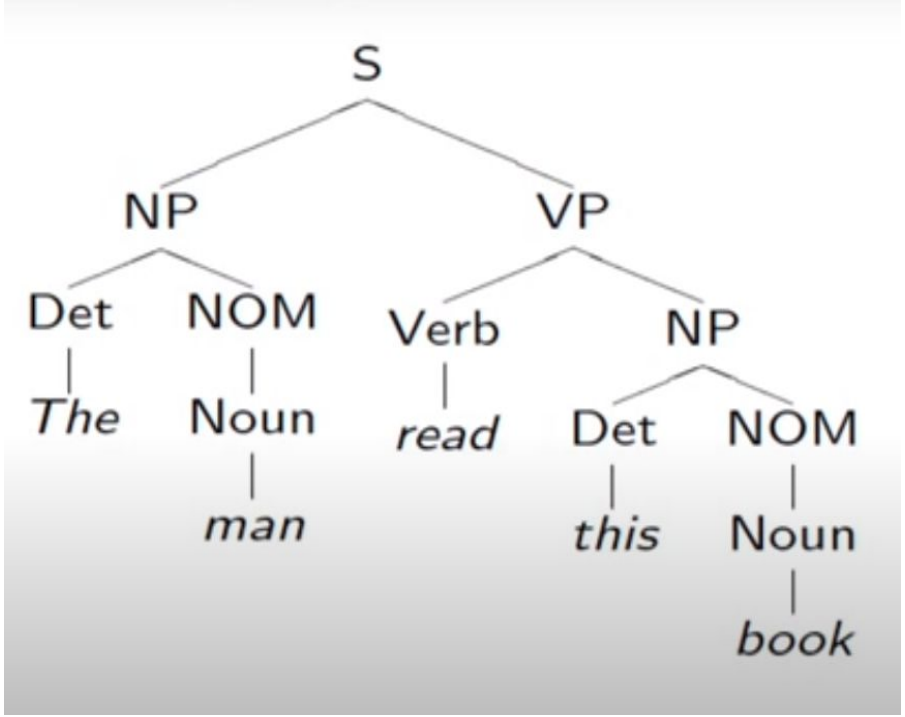
SYNTAX AND SEMANTICS

What is Syntax?

Refers to the way words are arranged together, and the relationship between them.

- Language Models: Importance of modeling word order.
  - Which words occur after what are the words? How we can make use of that in assigning probability for a sentence or finding out the next word in completion task or in spelling correction task,
- POS Categories: An equivalence class for words
  - talking about part of speech task, what are the grammatical categories, part of speech categories, etc.
- Syntax : some more complex notions like word is constituency,
  - what are different grammatical relations among the words? What groups or what words are grouped together in constituency and sub categorization

Example



Sentence: The man read this book

- By syntax we are trying to find out what are the various groups of word that are coming together
- example, in part of speech text we found out that **the** word the as part of speech of **determiner, man** as the part of speech of **noun** read as **verb** and so on.

Defining the notions: Constituency

- A group of words acts as a single unit - phrases, clauses etc.

In part of speech: “Substitution Test”

- The {sad, intelligent, green, fat, …} one is in the corner.

Constituency: Noun Phrase

- They, December twenty-sixth, the reason he is running for president.

Constituent Phrases

Usually named based on the word that heads the constituent:

the man from Amherst	is a Noun Phrase (NP) because the head man is a noun
extremely clever	is an Adjective Phrase (AP) because the head clever is an adjective
down the river	is a Prepositional Phrase (PP) because the head down is a preposition
killed the rabbit	is a Verb Phrase (VP) because the head killed is a verb

Example

Words can also act as phrases

- Joe grew potatoes
  - Joe and potatoes are both nouns and noun phrases
- The man from Amherst grew beautiful russet potatoes.
  - Joe appears in a place that a larger noun phrase could have been.

Evidence that constituency exists

They appear in similar environments

Kermit the frog comes on stage
They come to Massachusetts every summer
December twenty-sixth comes after Christmas
The reason he is running for president comes out only now.
But not each individual word in the consituent
**The comes our...* **is comes out...* **for comes out...*

Can be placed in a number of different locations

Consituent = Prepositional phrase: *On December twenty-sixth*
On December twenty-sixth I'd like to fly to Florida.
I'd like to fly on December twenty-sixth to Florida.
I'd like to fly to Florida on December twenty-sixth.
But not split apart
**On December I'd like to fly twenty-sixth to Florida.*
**On I'd like to fly December twenty-sixth to Florida.*

Modeling Constituency

Context-free grammar

- The most common way of modeling constituency.

Consists of production Rules

- Rules that express the ways in which the symbols of the language can be grouped and ordered together.

Example

Noun phrase can be composed of either a ProperNoun or a determiner (Det) followed by a Nominal; a Nominal can be more than one nouns

NP → Det Nominal

NP → ProperNoun

Nominal → Noun | Noun Nominal

CFG of Languages

$CFG: G = (T, N, S, R)$

- T: set of terminals
- N: set of non-terminals
  - For NLP, we distinguish out a set $P \subset N$ of pre-terminals, which always rewrite as terminals
- S : start symbol
- R: Rules/productions of the form $X \rightarrow \gamma$, $X \in N$ and $\gamma \in (T \cup N)*$

- Terminals mainly correspond to words in the language while
- Pre-terminals mainly correspond to POS (Parts of Speech) categories

Example

- NP → Det Nominal
- NP → ProperNoun
- Nominal → Noun | Noun Nominal

These can be combined with other rules, that express facts about a lexicon.

- Det → a
- Det → the
- Noun → flight

Terminals: a, the, flight. **Non-Terminals**: NP, Nominal. **Pre-Terminals**: Determiner, Noun

CFG as a generator

NP → Det Nominal

NP → ProperNoun

Nominal → Noun | Noun Nominal

Det → a

Det → the

Noun → flight

Generating ‘a flight’: NP→ Det Nominal → Det Noun → a Noun → a flight

A CFG can be used to randomly generate a series of strings.

This Sequence of rule expansions is called a derivation of the string of words. Usually represented as a tree

CFGs and Grammaticality

A CFG defines a formal language, a set of all sentences (string of words) that can be derived by the grammar.

- Sentences in this set are said to be grammatical
- Sentences outside this set are said to be ungrammatical

CFGs and Recursion

In Language we can make lot of big sentences by using recursion.

Recursion Definition

- PP → Prep NP
  - A prepositional phrase can be written as a preposition followed by a noun phrase
- NP → Noun PP
  - Noun phrase can be written as a noun phrase followed by a preposition phrase

Example Sentence

[$_S$The mailman ate his [$_{NP}$ lunch [$_{PP}$ with his friend [$_{PP}$ from the cleaning staff [$_{PP}$ of the building [$_{PP}$ at the intersection [$_{PP}$ on the north end [$_{PP}$ of town]]]]]]].

What does context stand for in CFG?

The notion of context has nothing to do with the ordinary meaning of word context in language

All it really means is that the non-terminal on the left-hand of a rule is out there all by itself. (free of context)

Example:

A→ BC

__ A __ → __ BC __

x A y → x BC y

$A \rightarrow BC$

- I can rewrite A as *B followed by C* regardless of the context in which A is found
- Or when I see a *B followed by a C*, I can infer an A regardless of the surrounding context

Syntax Parsing

S → NP VP

S → Aux NP VP

S → VP

NP → Det NOM

NOM → Noun

NOM → Noun NOM

VP → Verb

VP →Verb NP

Det → that | this | a | the

Noun → book | flight | meal | man

Verb → book | include | read

Aux → does

Generating a sentence

S → NP VP

→ Det NOM VP

→ The NOM VP

→ The Noun VP

→ The man VP

→ The man Verb NP

→ The man read NP

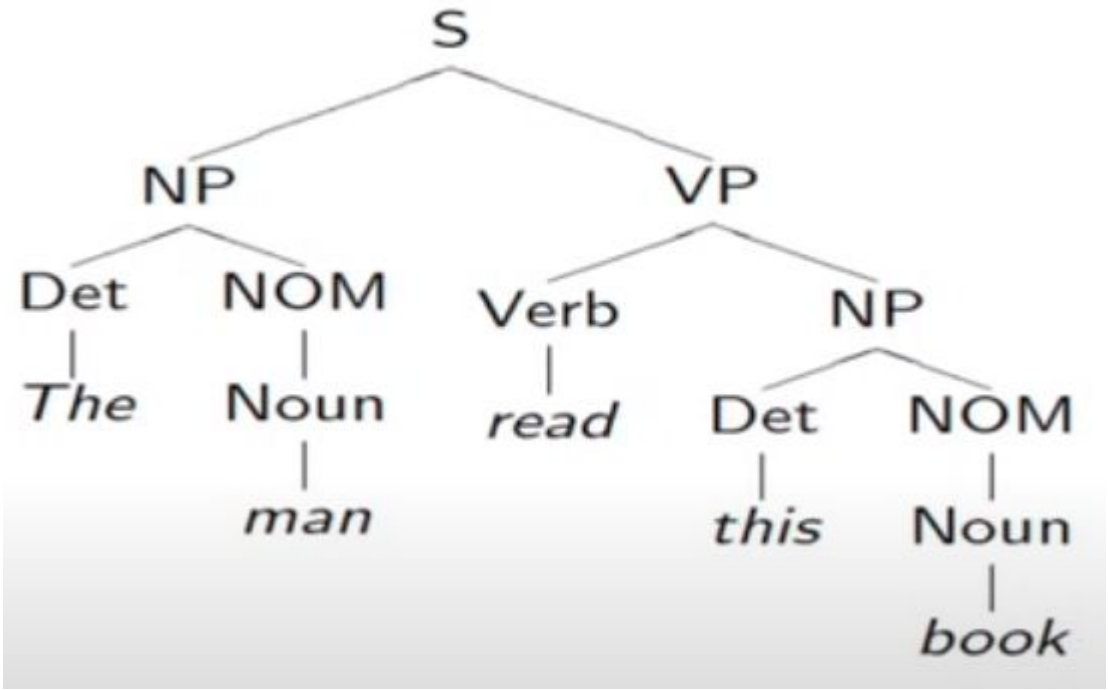
→ The man read Det NOM

→ The man read this NOM

→ The man read this Noun

→ The man read this book

Parse Tree



What is Parsing?

The process of taking a string and a grammar and returning all possible parse trees for that string.

- So the process of parsing is to find out all the possible parse trees for a given sentence as per my grammar.
- That is, find all trees, whose root is the start symbol S, Which cover exactly the words in the input.

What are the constraints? "book that flight"

- There must be three leaves, *book*, *that* and *flight*
- The tree must have one root, the start symbol *S*
- Give rise to two search strategies: *top-down* (goal-oriented) and *bottom-up* (data-directed)

Parsing

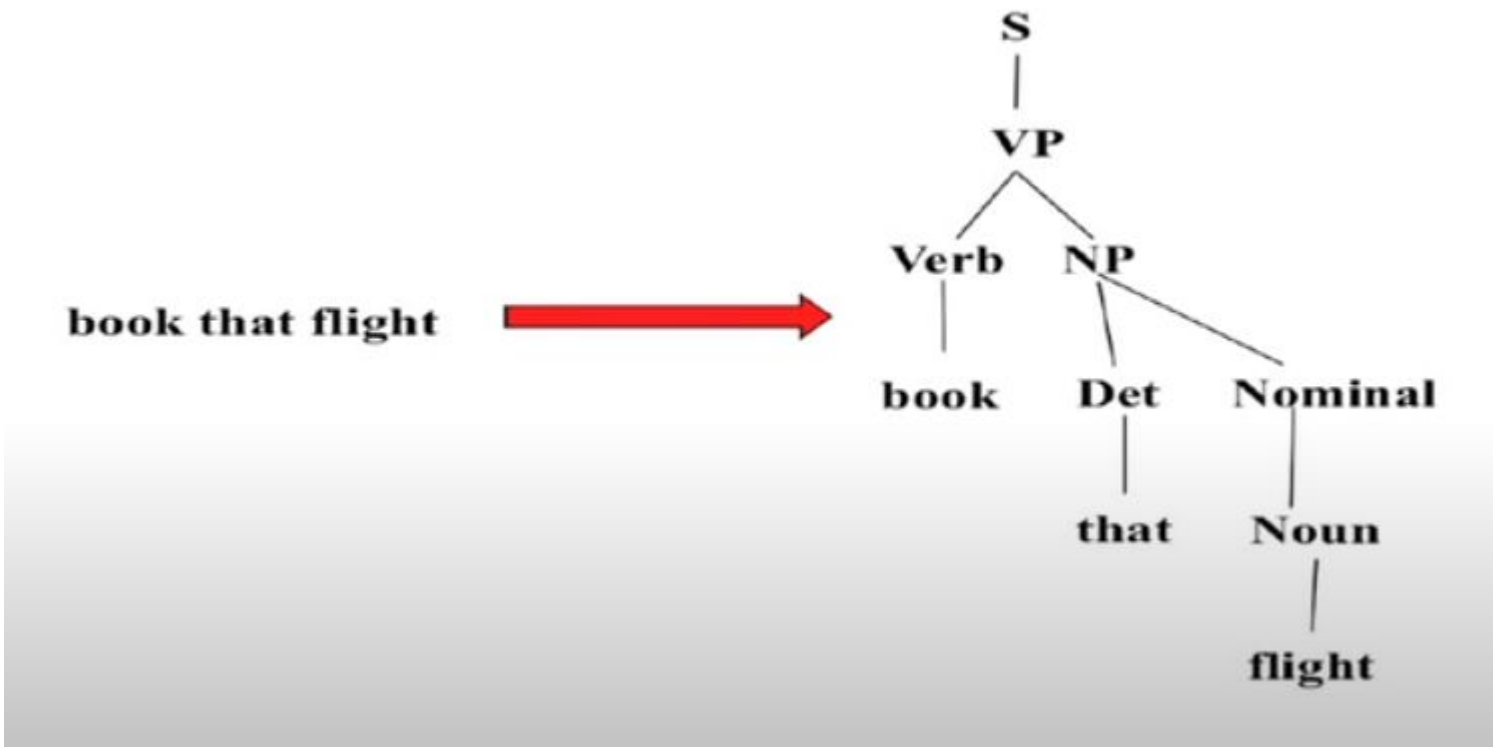
Grammar

S → NP VP
S → Aux NP VP
S → VP
NP → Pronoun
NP → Proper-Noun
NP → Det Nominal
Nominal → Noun
Nominal → Nominal Noun
Nominal → Nominal PP
VP → Verb
VP → Verb NP
VP → VP PP
PP → Prep NP

Lexicon

Det → the | a | that | this
Noun → book | flight | meal | money
Verb → book | include | prefer
Pronoun → I | he | she | me
Proper-Noun → Houston | NWA
Aux → does
Prep → from | to | on | near | through

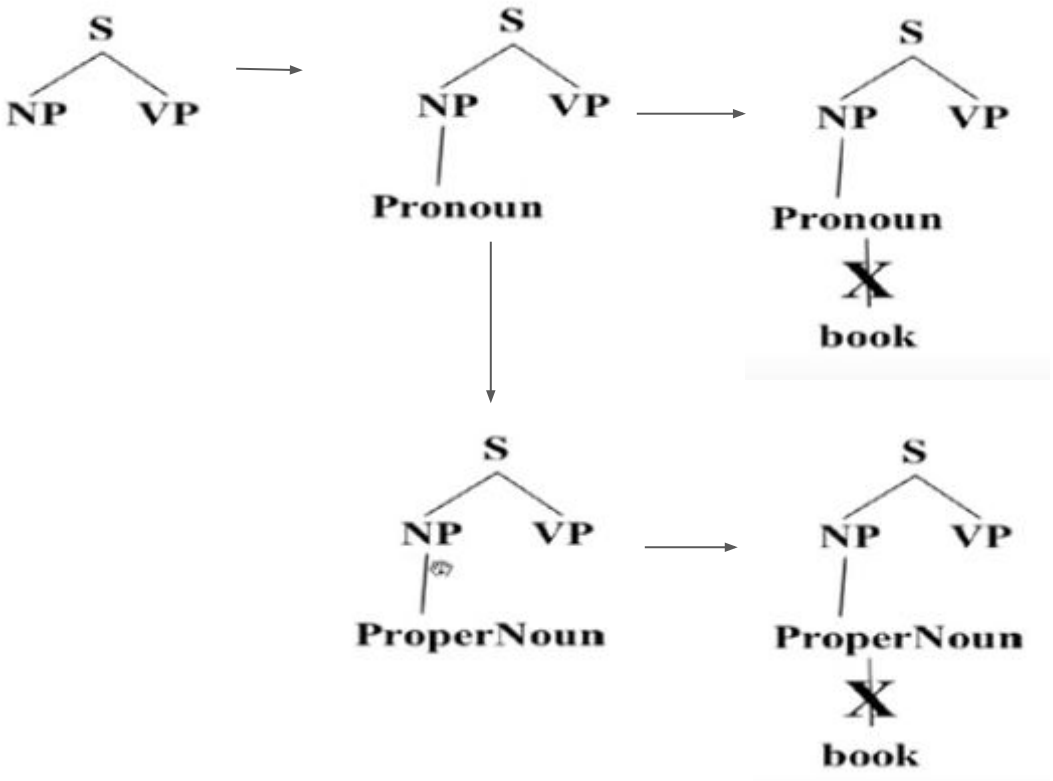
Parsing


book that flight
S
VP
Verb
NP
book
Det
Nominal
that
Noun
flight

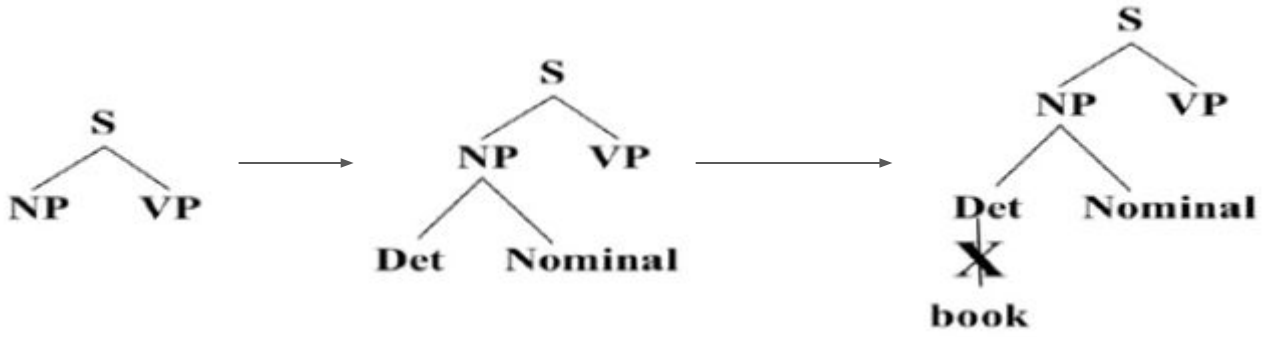
Top Down Parsing

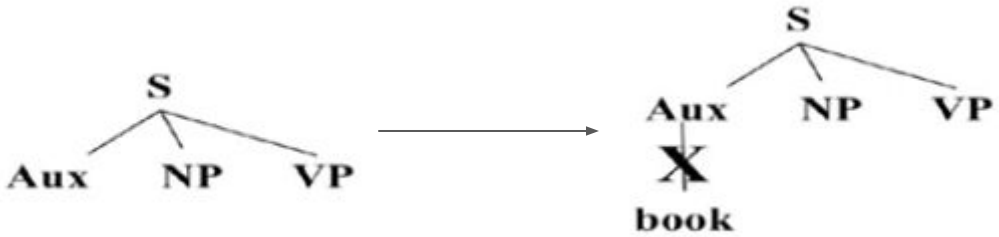
- Searches for a parse tree by trying to build upon the root node S down to the leafs
- Start by assuming that the input can be derived by the designated start symbol S
- Find all trees that can start with S, by looking at the grammar rules with S on the left-hand side
- Trees are grown downward until they eventually reach the POS categories at the bottom
- Trees whose leaves fail to match the words in the input can be rejected.

Top Down Parsing

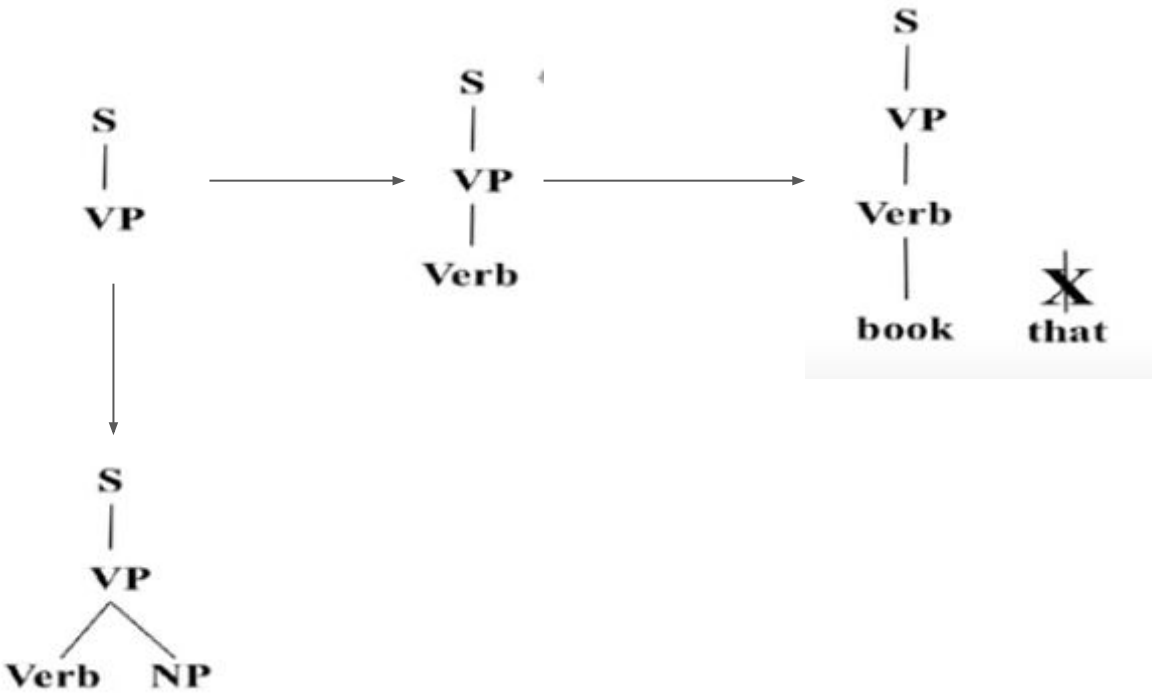


Top Down Parsing

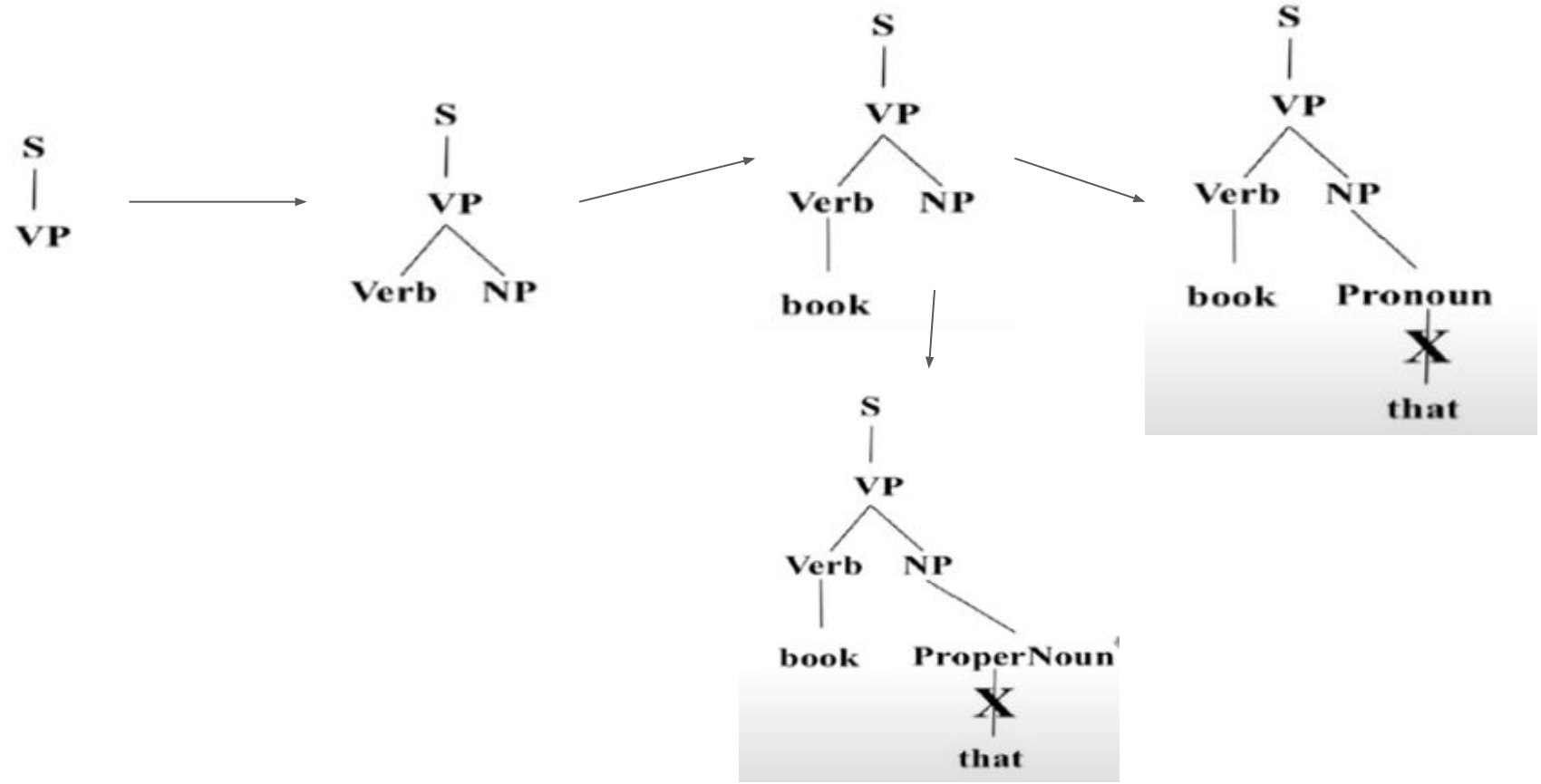




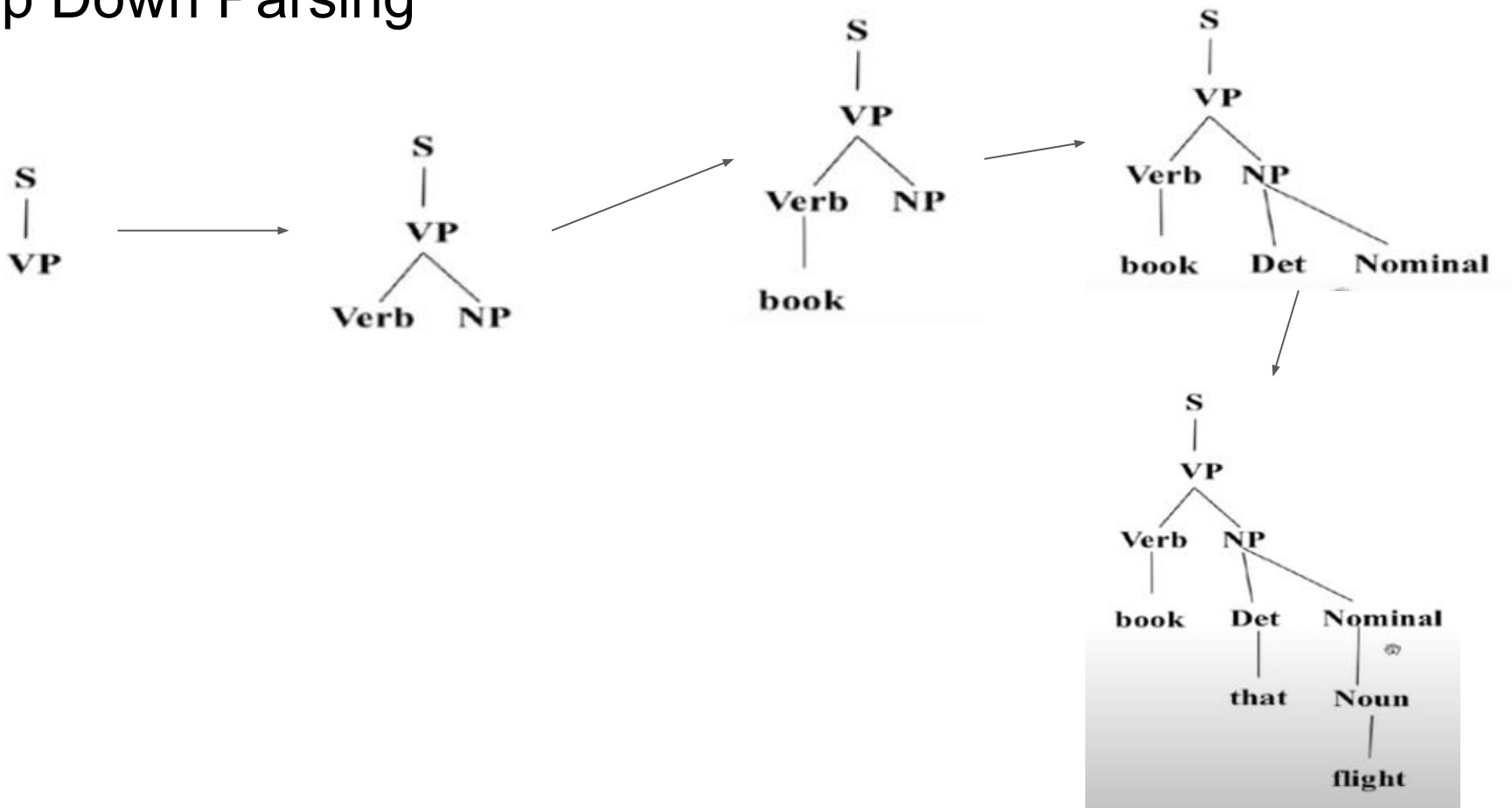
Top Down Parsing



Top Down Parsing



Top Down Parsing



Top Down Approach

Parse tree starts with S and covers all leaf nodes with Top Down approach.

Bottom-Up Parsing

The parser starts with the words of the input, and tries to build trees from the words up, by applying rules from the grammar one at a time.

- Parser looks for the places in the parse-in-progress where the right-hand-side of some rule might fit.

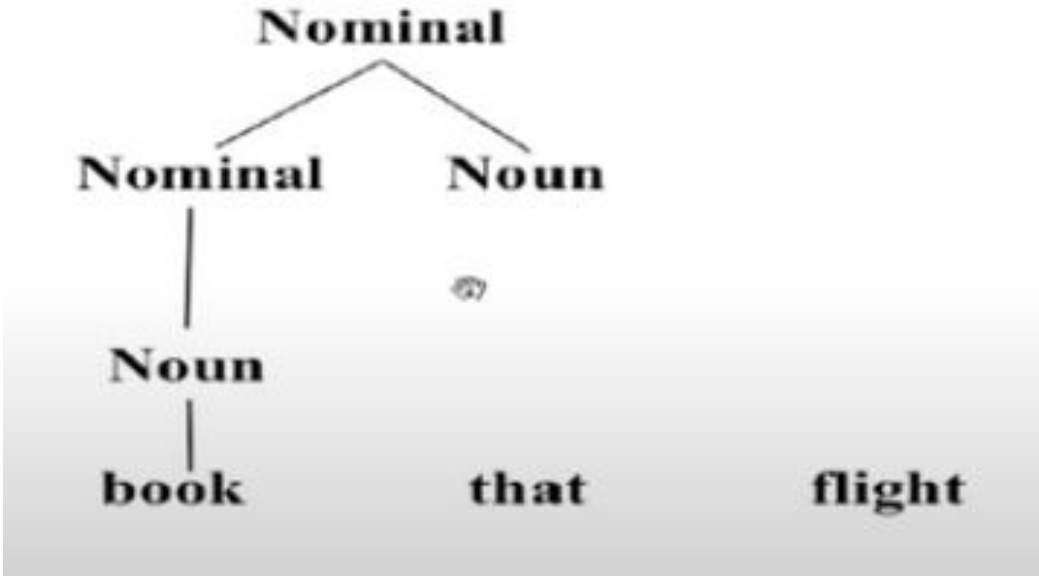
Bottom-Up Parsing

book that flight

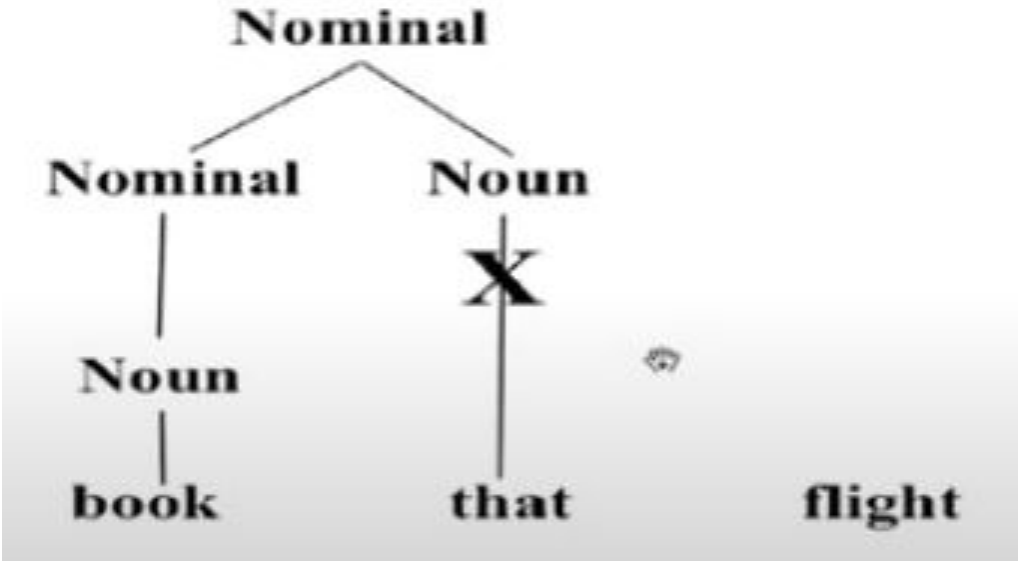
Bottom-Up Parsing

Bottom-Up Parsing

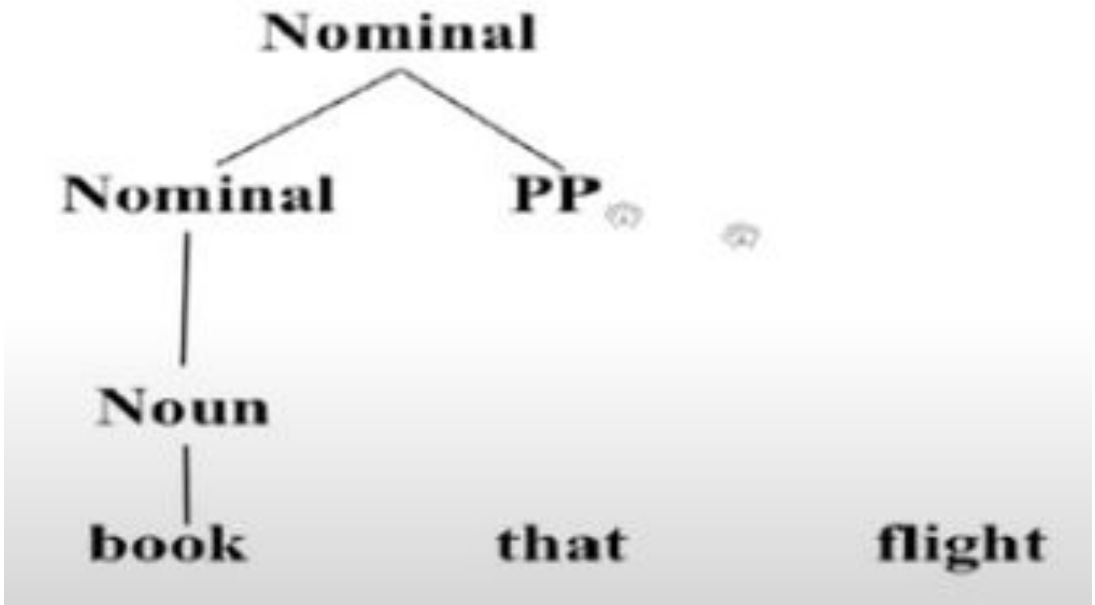
Bottom-Up Parsing



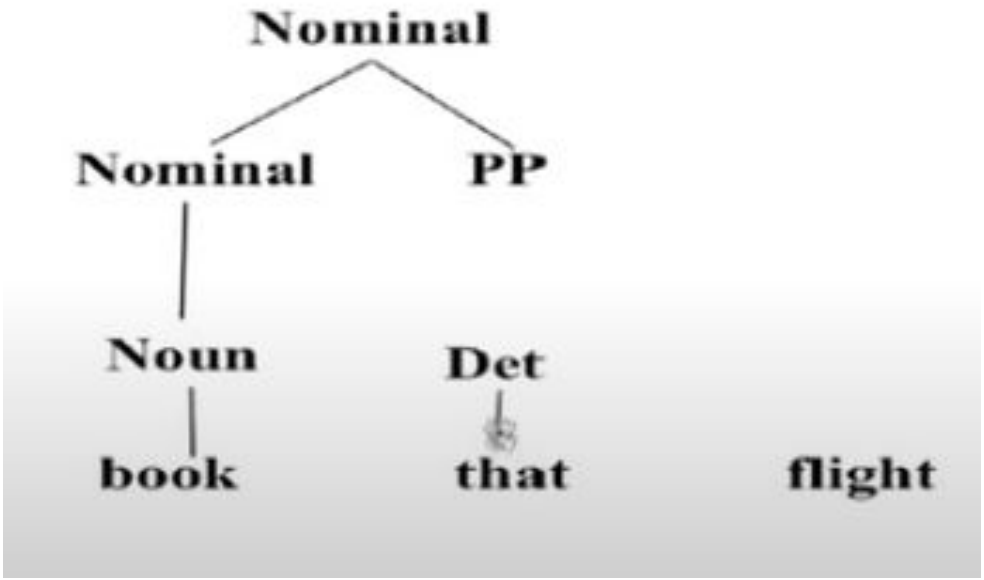
Bottom-Up Parsing



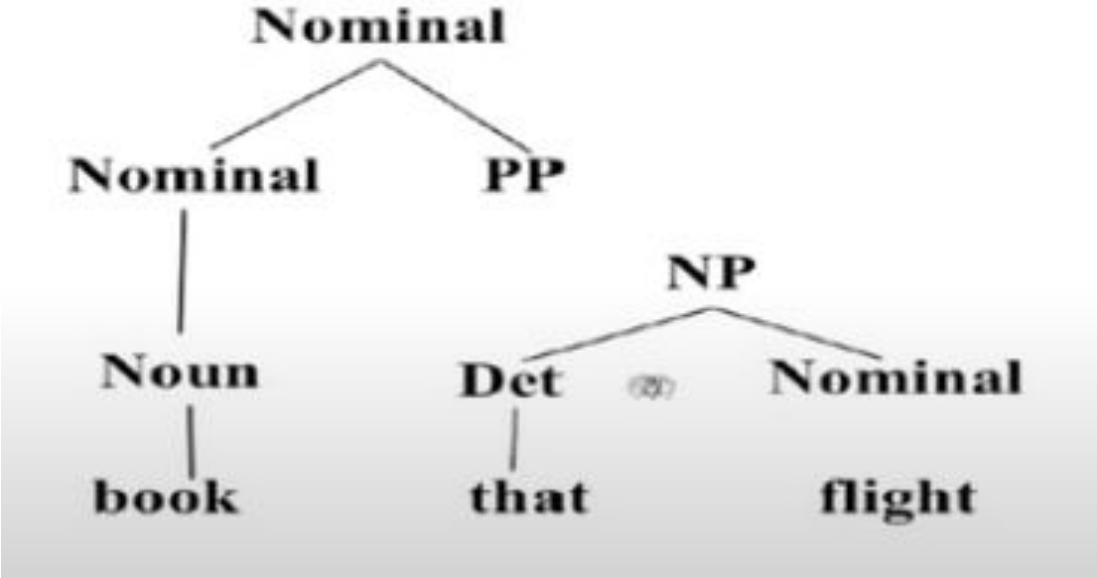
Bottom-Up Parsing



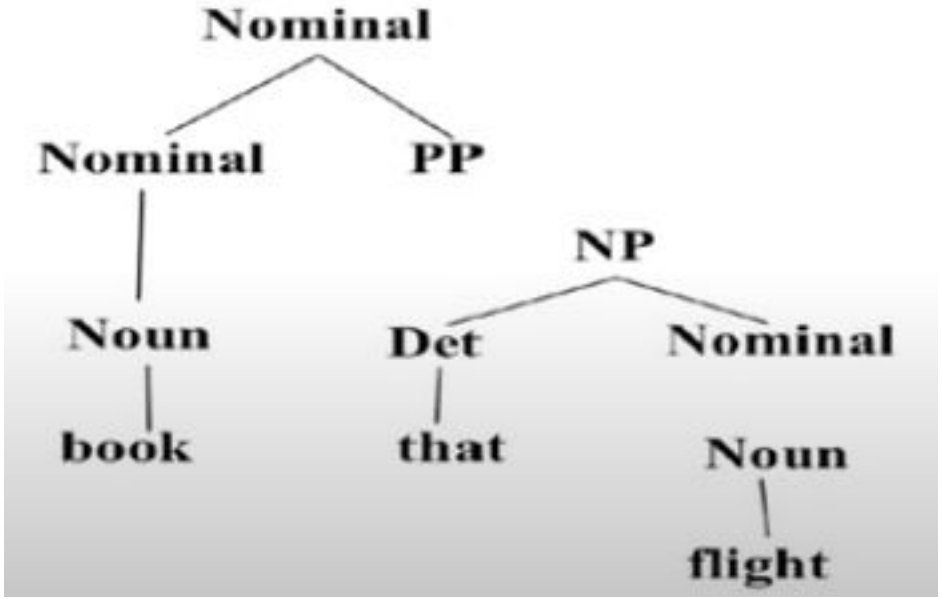
Bottom-Up Parsing



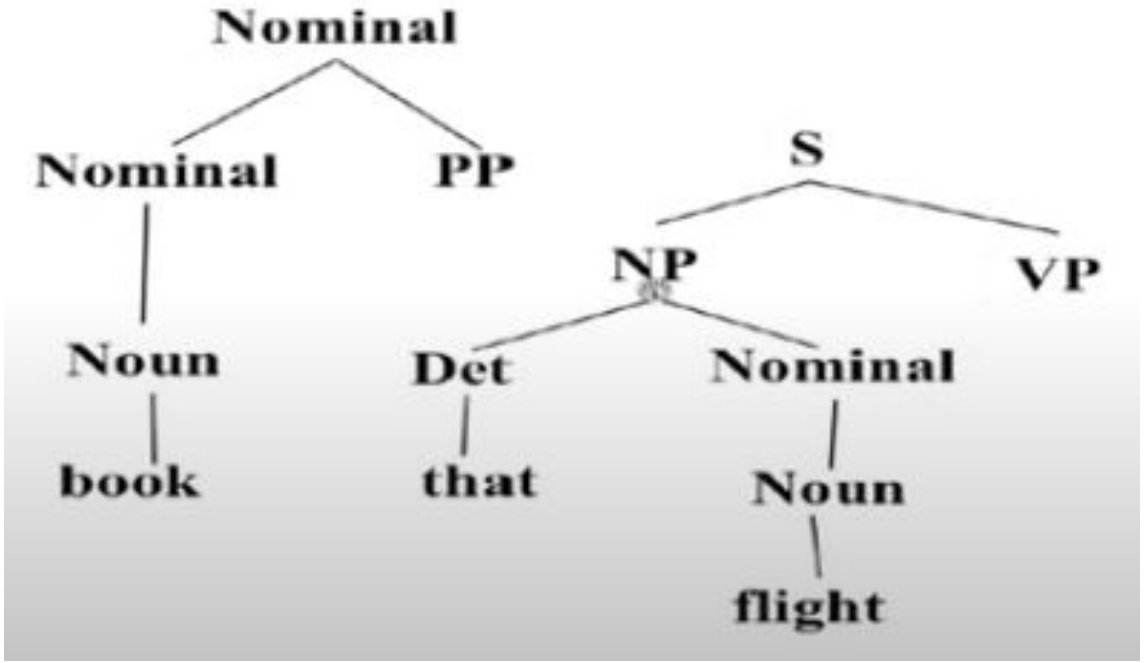
Bottom-Up Parsing



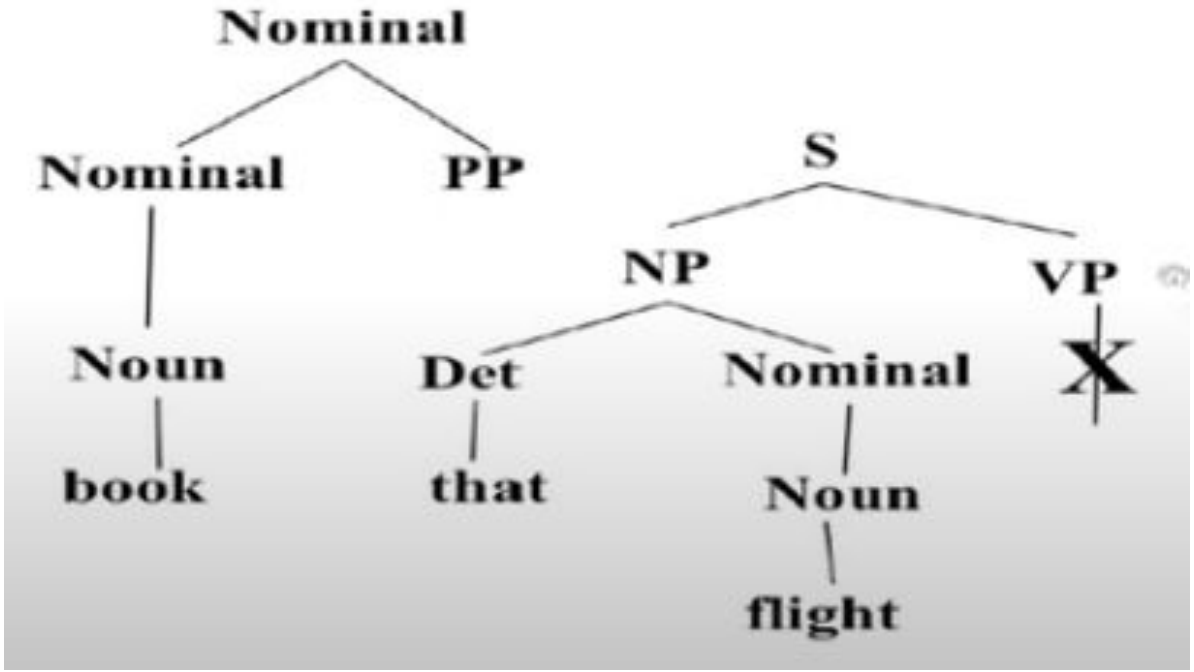
Bottom-Up Parsing



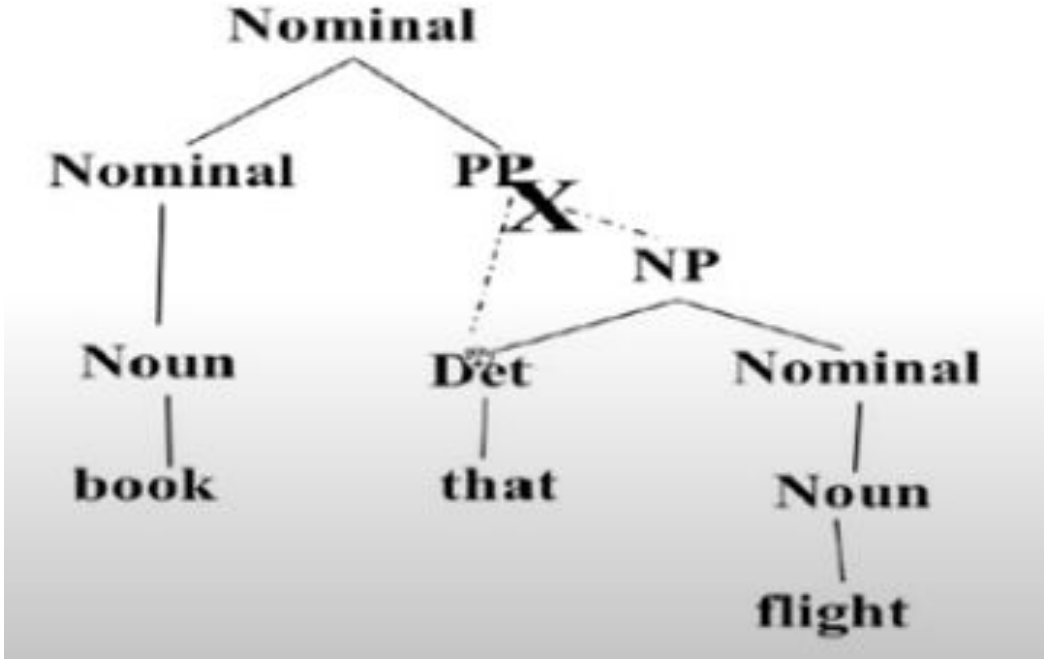
Bottom-Up Parsing



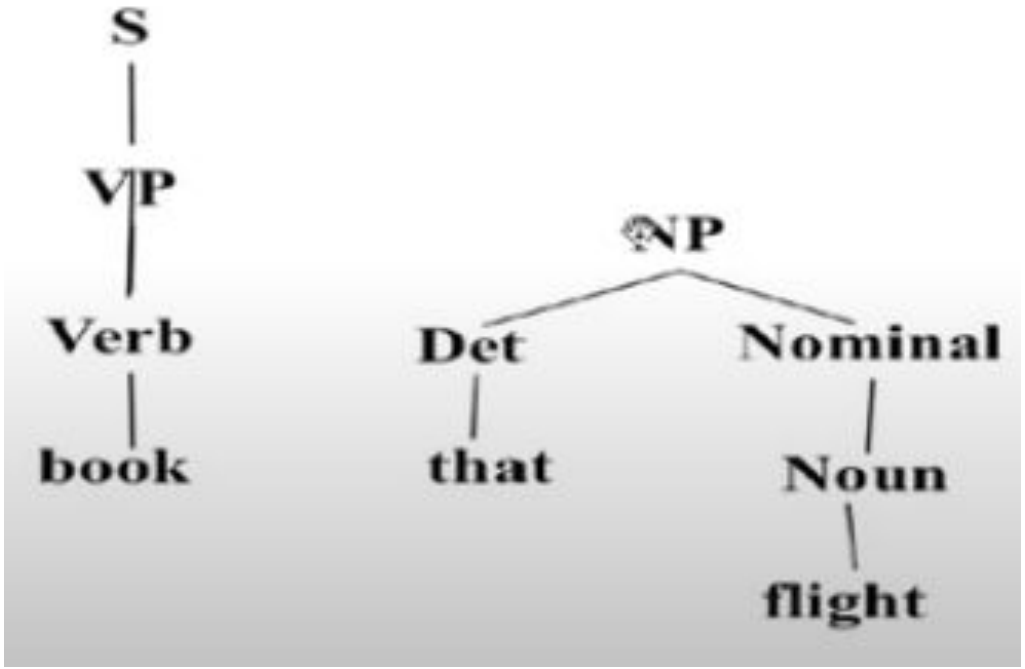
Bottom-Up Parsing



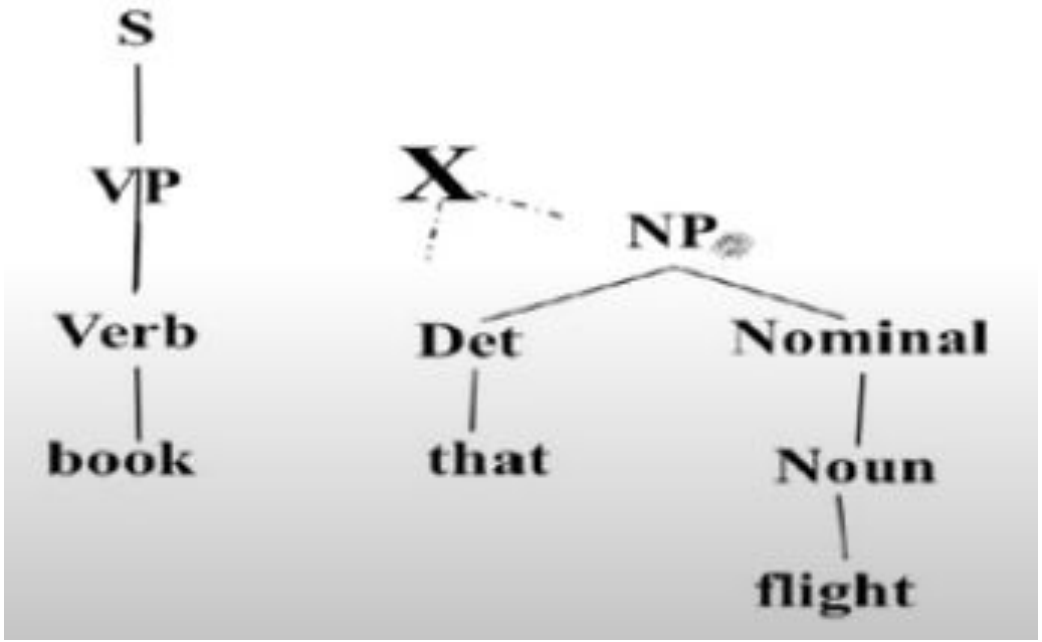
Bottom-Up Parsing



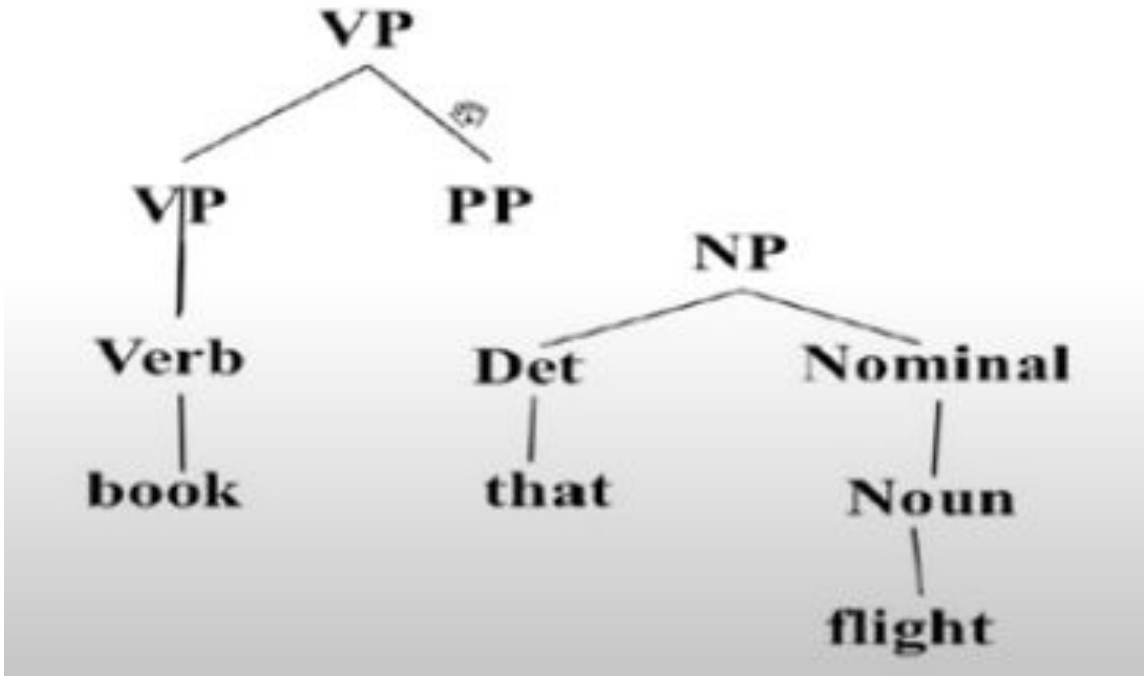
Bottom-Up Parsing



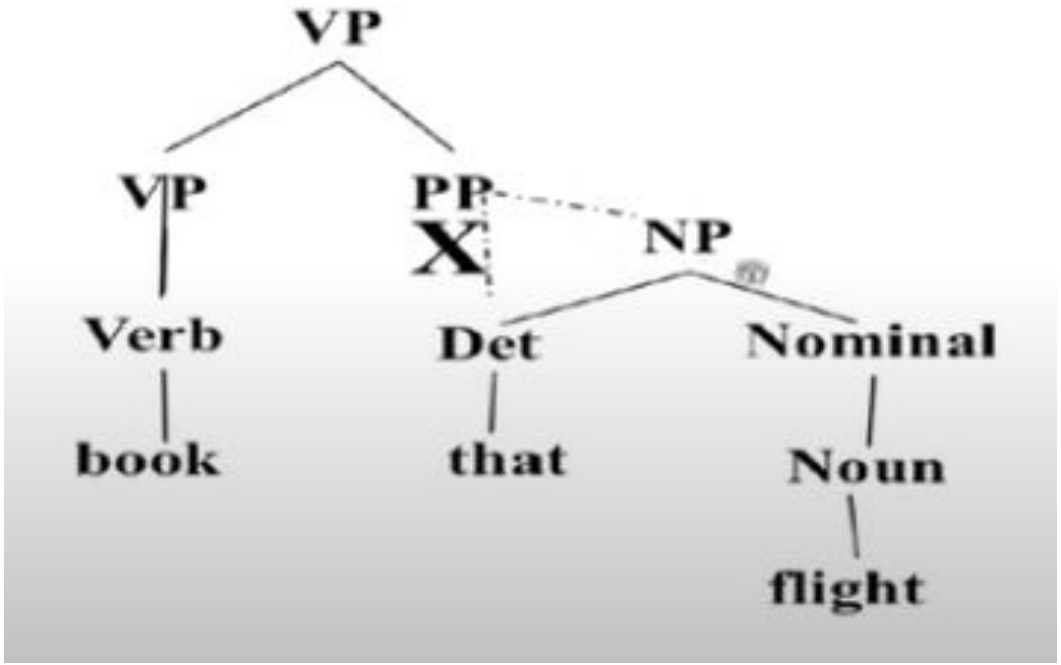
Bottom-Up Parsing



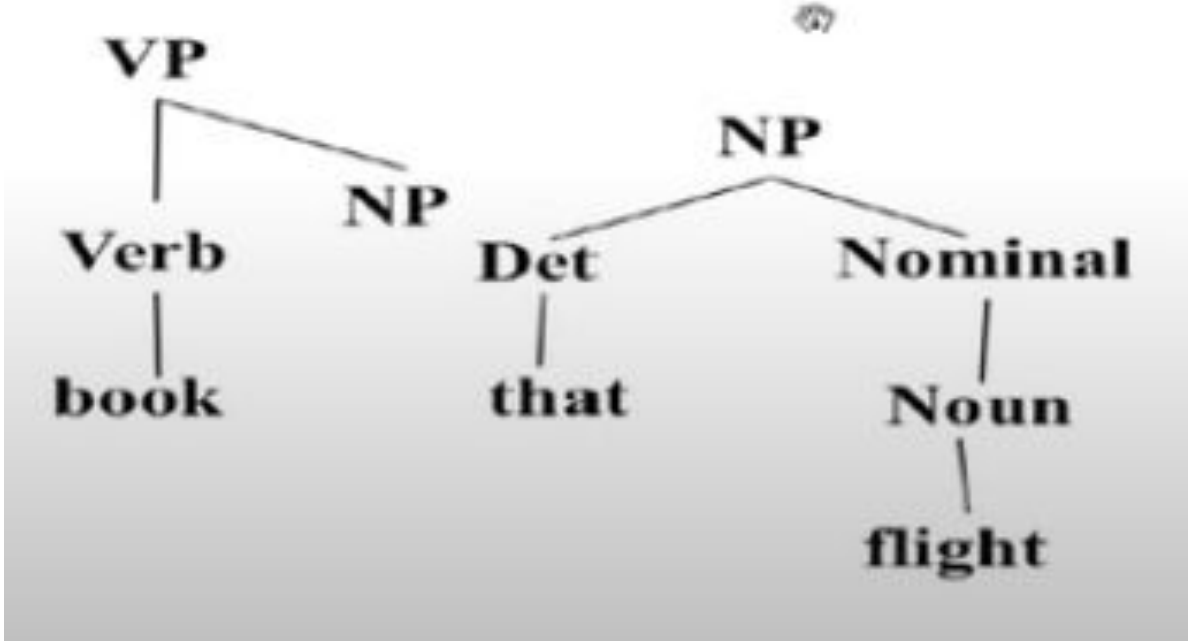
Bottom-Up Parsing



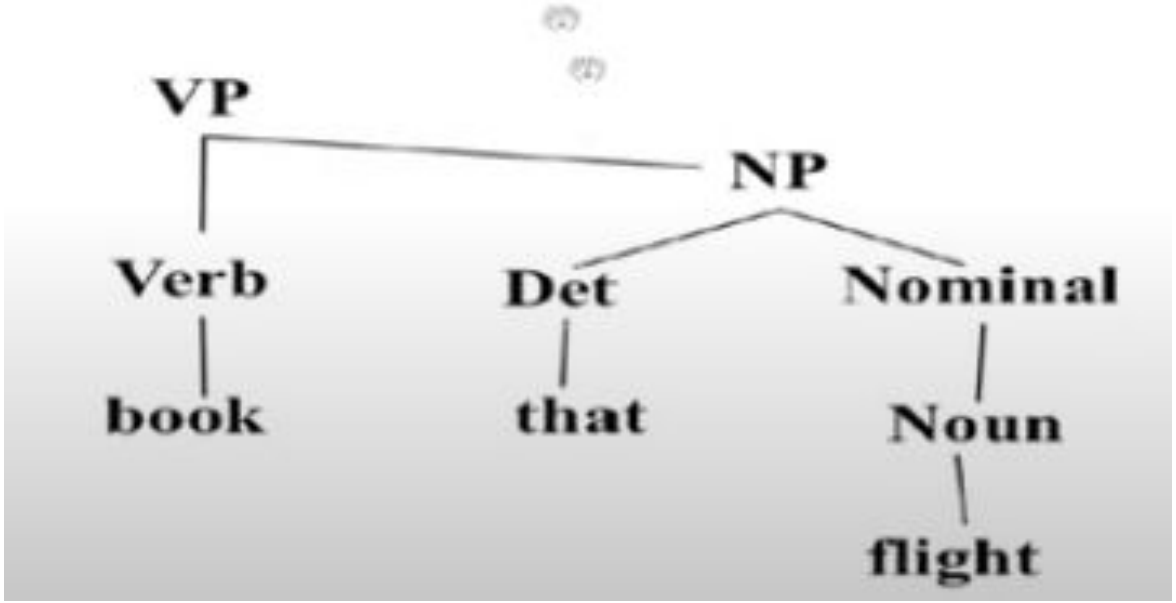
Bottom-Up Parsing



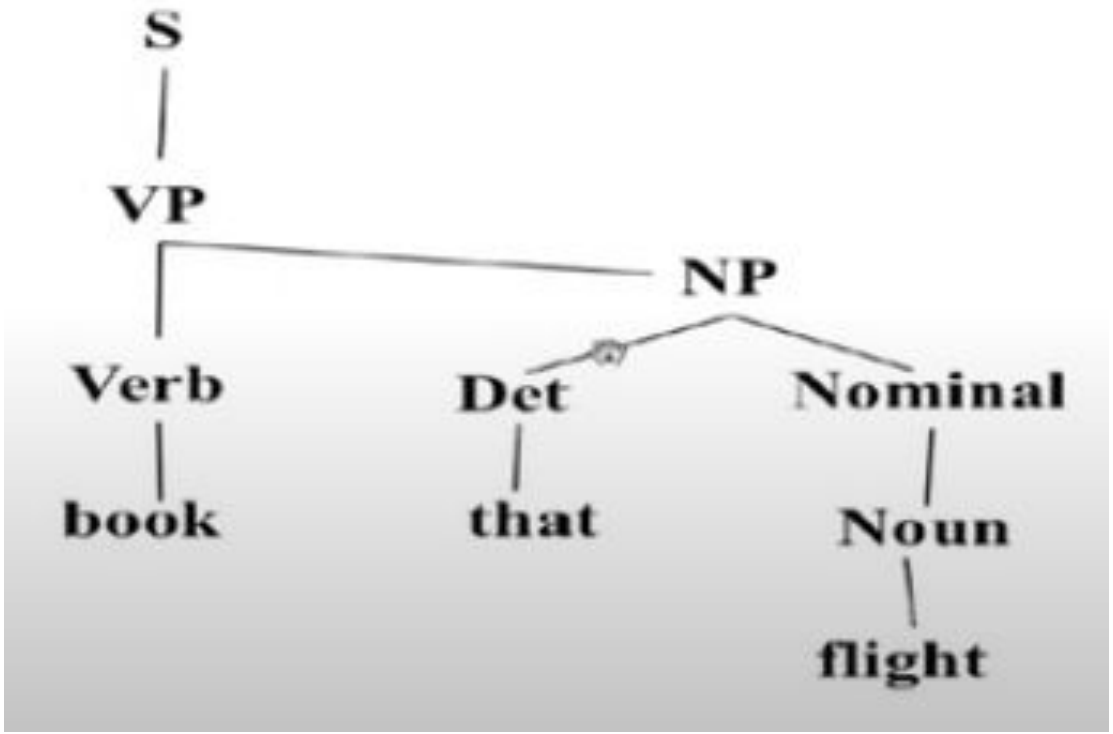
Bottom-Up Parsing



Bottom-Up Parsing



Bottom-Up Parsing



Top-Down vs. Bottom-Up

- Top down never explores options that will not lead to a full parse, but can explore many options that never connect to the actual sentence.
- Bottom up never explores options that do not connect to the actual sentence but can explore options that can never lead to a full parse.
- Relative amounts of wasted search depend on how much the grammar branches in each direction.