Natural Language Processing

6th Semester B.Tech. (CSE)

Course Code:BTCS-T-PE-052

Nayan Ranjan Paul

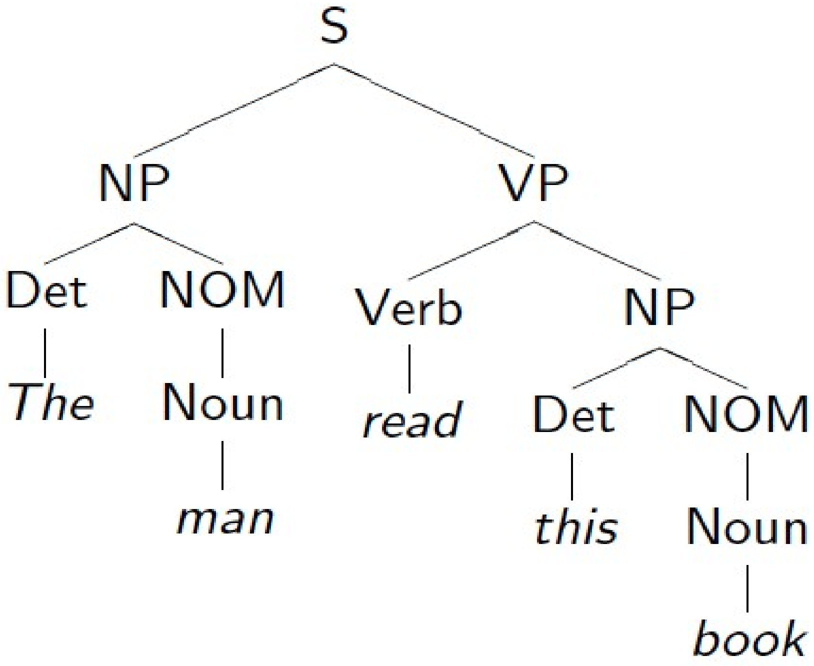
Department of CSE

Silicon Institute of Technology

Module-III

Syntactic Analysis

- The word 'syntax' refers to the grammatical arrangement of words in a sentence and their relationship with each other.
- The objective of syntactic analysis is to find the syntactic structure of the sentence.
- This structure is represented as a tree.
  - Nodes – represents phrases
  - Leaves – corresponds to words
  - Root – represents whole sentence



Syntactic Analysis

- Identifying the syntactic structure is useful in determining the meaning of the sentence.
- The identification is done using a process called parsing.
- Syntactic parsing can be considered as the process of assigning ‘phrase marker’ to a sentence.
- What phrases are?

Syntactic Analysis

- Two important ideas in natural language are constituency and word order.
- Constituency
  - It is about how words are grouped together and how we know that they are really grouping together.
  - Consttuent - A group of words acts as a single unit - phrases, clauses etc.
- Word order
  - It is about how, within a constituent, words are ordered with respect to one another and also how constituents are ordered with respect to one another.
- A widely used mathmetical system for modelling constituent structure in NLP is context-free grammar(CFG)

Modeling Constituency

- Context-free grammar
  - The most common way of modeling constituency
- Consists of production Rules
  - These rules 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

Context-free Grammar for 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
  - P : set of production rules of the form $X \rightarrow \gamma$, $X \in N$ and $\gamma \in (T \cup N)^*$

- Terminals and pre-terminals
  - Terminals mainly correspond to words in the language while pre-terminals mainly correspond to POS categories

Context-free Grammar for Languages

- The rule X → γ says that constituent X can be rewritten as γ. This is also called phrase structure rule.
- It specifies which elements or consttuents can occur in a phrase and in what order .
- Example

  NP → Det Nominal

  NP → ProperNoun

  Nominal → Noun | Noun Nominal
- Now, these can be combined with other rules, that express facts about a lexicon.

  Det → a

  Det → the

  Noun → flight
- Can you identify the terminal, non-terminals and preterminals?

Context-free Grammar for Languages

- A CFG can be used to generate a sentence or assign a structure to a given sentence.
- During generation the arrow in production rule may be read as “rewrite the symbol on the left with symbols on the right.”
- Example
  - NP → Det Nominal
  - NP → ProperNoun
  - Nominal → Noun | Noun Nominal
  - Det → a
  - Det → the
  - Noun → flight

Context-free Grammar for Languages

- NP → Det Nominal

  NP → ProperNoun

  Nominal → Noun | Noun Nominal

  Det → a

  Det → the

  Noun → flight
- Generating ‘a flight’:

Context-free Grammar for Languages

- NP → Det Nominal

  NP → ProperNoun

  Nominal → Noun | Noun Nominal

  Det → a

  Det → the

  Noun → flight
- Generating ‘a flight’:

  NP

Context-free Grammar for Languages

- NP → Det Nominal

  NP → ProperNoun

  Nominal → Noun | Noun Nominal

  Det → a

  Det → the

  Noun → flight
- Generating ‘a flight’:

  NP→ Det Nominal

Context-free Grammar for Languages

- NP → Det Nominal

  NP → ProperNoun

  Nominal → Noun | Noun Nominal

  Det → a

  Det → the

  Noun → flight
- Generating ‘a flight’:

  NP→ Det Nominal→ Det Noun

Context-free Grammar for Languages

- 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

Context-free Grammar for Languages

- 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

Context-free Grammar for Languages

- 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
- Thus 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 = 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

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 side of a rule is out there all by itself (free of context)
- $A \rightarrow BC$
  - 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

Constituency

- Fundamental idea of syntax is that words group together to form constituents(often termed phrases), each of which acts as a single unit.
- They combine with other constituents to form larger constituents and eventually , a sentence.
- Example – NP – “The bird”

  VB – “flies”

  - These two constituents can combine to form the sentence “The bird flies”

Phrase Level Constructions

- Fundamental notion in Natural language is that certain groups of words behave as constituents.
- These constituents are identified by their ability to occur in similar contexts.
- How to decide whether a group of words is a phrase?
  - Use substitution test
- Substitution test – by checking if a group of words in a sentence can be substituted with some other group of words without changing the meaning. If such substitution is possible then the set of words forms phrase.
- Example
  - Hena reads a book.
  - Hena reads a story book.
  - Those girls read a book.
  - She reads a comic book.

Phrase Level Constructions

- Phrase types are named after their head.
- Usually phrases are 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

Noun Phrase

- A noun phrase is a phrase whose head is a noun or a pronoun, optionally accompanied by a set of modifiers.
- It can function as
  - Subject
  - Object
  - Complement
- The modifiers of a noun phrase can be determiners or adjective phrases
- The noun phrase must have a noun as head-all other constituents are optional.
- These structures can be represented using phrase structure rules.
- These rules specify which elements can occur in a phrase and in what order.

Noun Phrase

- Phrase structure rules for noun phrase

  NP $\rightarrow$ Pronoun

  NP $\rightarrow$ Det Noun

  NP $\rightarrow$ Noun

  NP $\rightarrow$ Adj Noun

  NP $\rightarrow$ Det Adj Noun

- Above rules can be combined into a single phrase structure rule as

  NP $\rightarrow$ (Det) (Adj) Noun|Pronoun

- The Adj can be replaced with AP and the Noun|Pronoun can be replaced with Noun PP.

  NP $\rightarrow$ (Det) (AP) Noun (PP)

Noun Phrase

- Examples noun phrase
  - They -Pronoun
  - The foggy morning -Det AP Noun
  - Chilled water -AP Noun
  - A beautiful lake i Kashmir -Det AP Noun PP

Verb Phrase

- A verb phrase is a phrase whose head is a verb.
- The verb phrase is a bit complex.
- The verb phrase organizes various elements of the sentence that depend syntactically on the verb.
- Phrase structure rule for verb phrase
  - VP → Verb (NP) (NP) (PP)*
  - VP → Verb S
- Example
  - Khushbu slept - Verb
  - The boy kicked the ball -Verb NP
  - Khushubu slept in the garden - Verb PP
  - The boy gave the girl a book. -Verb NP NP
  - The boy gave the girl a book with blue cover. -Verb NP NP PP
  - I know that Taj is one of the seven wonder. -Verb S

Prepositional Phrase

- A prepositional phrase is a phrase whose head is a preposition.
- Phrase structure rule for prepositional phrase
  - PP $\rightarrow$ Prep (NP)
- Example
  - We played volleyball on the beach - Prep NP
  - Joan went outside - Prep

Adjective Phrase

- An adjective phrase is a phrase whose head is an adjective.
- Phrase structure rule for prepositional phrase
  - AP → (Adv) Adj (PP)
- Example
  - Ashish is clever - Adj
  - The train is very late - Adv Adj
  - My sister is fond of animals -Adj PP

Adverb Phrase

- An adverb phrase is a phrase whose head is an adverb.
- It consists of an adverb, possibly preceded by a degree adverb.
- Phrase structure rule for prepositional phrase
  - AdvP → (Interns) Adv
- Example
  - Time passes very quickly. - Interns Adv

Sentence Level Construction

- A sentence can have varying structure.
- Four commonly known structure of sentences are:
  - Declarative structure
  - Imperative structure
  - Yes-no question structure
  - Wh-question structure
- Declarative structure
  - It has a subject followed by a predicate.
  - The subject is NP and the predicate is VP
  - Phrase structure rule is $S \rightarrow NP\ VP$
  - Example - “ I like horse riding.”

Sentence Level Construction

- Imperative structure
  - It usually begin with a verb phrase and lack of subject.
  - The subject is implicit and is understood to be ‘you’.
  - Used for commands and suggestion and hence are called imperative.
  - Phrase structure rule is $S \rightarrow VP$
  - Example
    - Look at the door
    - Give me the book
    - Stop talking
    - Show me the latest design

Sentence Level Construction

- Yes-no question structure
  - These are questions which can be answered using yes or no.
  - These are begin with an auxiliary verb, followed by a subject NP, followed by a VP.
  - Phrase structure rule is S → Aux NP VP
  - Examples
    - Do you have a red pen?
    - Is the game over?
    - Can you show me your album?
    - Is there a vacent quarter?

Sentence Level Construction

- Wh-question structure
  - These are begin with wh-words – who, which, where, what, and how.
  - It may have a wh-phrase as a subject or may include another subject.
  - These are similar to a declarative sentence with a wh-word
  - Another type of wh-question is that involves more than one NP.
  - Phrase structure rules are:
    - $S \rightarrow$ Wh-NP VP
    - $S \rightarrow$ Wh-NP Aux NP VP
  - Example
    - Which team won the match?
    - Which cameras can you show me in your shop?

Summary of all grammar rules

- $S \rightarrow NP\ VP$
- $S \rightarrow VP$
- $S \rightarrow Aux\ NP\ VP$
- $S \rightarrow Wh\text{-}NP\ VP$
- $S \rightarrow Wh\text{-}NP\ Aux\ NP\ VP$
- $NP \rightarrow (Det)\ (AP) Nom\ (PP)$
- $VP \rightarrow Verb\ (NP)\ (NP)\ (PP)^{*}$
- $VP \rightarrow Verb\ S$
- $AP \rightarrow (Adv)\ Adj\ (PP)$
- $PP \rightarrow Prep\ (NP)$

Coordination

- There are other sentence level structures that cannot be modelled by the rules discussed here.
- Coordination is one such structure.
- It refers to conjoining phrases with conjuctions like ‘and’, ‘or’, and ‘but’.
- Conjuction rules for NP, VP, and S can be built as:

  NP → NP and NP

  VP → VP and VP

  S → S and S
- Example
  - I ate an apple and a banana
  - It is dazzling and raining
  - I am reading the book and I am also watching the movie

Agreement

- Grammatical categories impose certain constraints: subject-verb agreement is one of them.
- The subject in a sentence must agree with the main verb in number and person.
- The 3rd person singular(3sg) form ends with a -s whereas the non-3sg does not.
- Whenever there is a verb that has some noun acting as a subject, this agreement has to be confirmed.
- Example :

  Does[$_{NP}$ Priya] sing?- here the subject NP is singular so -es form is used

  Do[$_{NP}$ they] eat?- here the subject NP is plural so the form ‘do’ is used
- Grammar rules should be able to ensure this agreement.

Feature Structures

- CFG can not handle subject-verb agreement efficiently.
- Feature structures can be used to efficiently capture the properties of grammatical categories.
- These are feature value pairs.
- Features are simply symbols representing properties that we wish to capture.
- For example, the number property of a noun phrase can be represented by NUMBER feature.
- The value it can take is SG(singular) or PL(Plural)
- Feature structures are represented by a matrix-like diagram called attribute value matrix(AVM)
- An AVM with single number feature with the value SG is represented as

  [NUMBER   SG]

Parsing

- CFG defines the syntax of a language but does not specify how structures are assigned.
- A phrase structure tree constructed from a sentence is called a parse or parse tree
- The process of taking a string and a grammar and returning all possible parse trees for that string is called parsing.
- The syntactic parser is thus responsible for recognizing a sentence and assigning a syntactic structure to it.
- One sentence can have multiple parses. This phenomenon is called syntactic ambiguity.
- Finding the right parse can be viewed as a search process.
- Search find all trees, whose root is the start symbol S , which cover exactly the words in the input.

Parsing

- The search space in this conception corresponds to all possible parse trees defined by the grammar.
- The following constraints guide the search process.
  - Input: A valid parse is one that covers all the words in an input sentence as leaves.
  - Grammar: The root of the final parse tree must be the start symbol of the grammar.
- These two constraints give rise to two search strategies:
  - top-down (goal-oriented) and
  - bottom-up (data-directed)

Parsing

Grammar

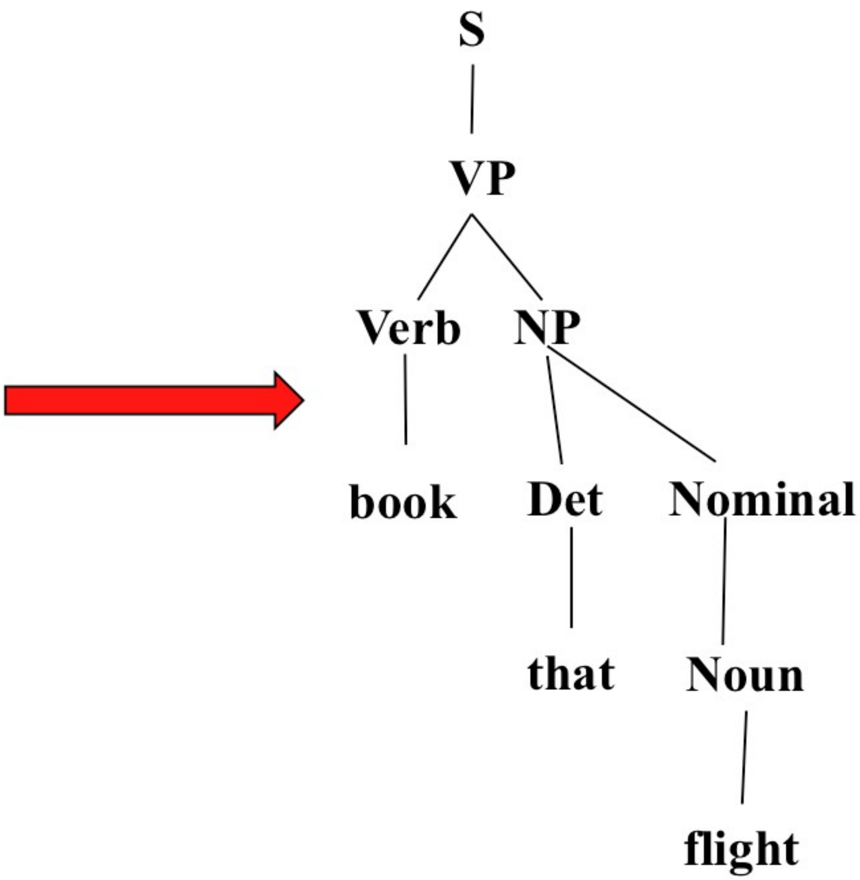
$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$
$S \rightarrow VP$
$NP \rightarrow Pronoun$
$NP \rightarrow Proper\text{-}Noun$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow Noun$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow Verb$
$VP \rightarrow Verb\ NP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Prep\ NP$

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



Top-down Parsing

- Top-down parsing starts its search from the root node S and works downwards towards the leaves.
- i.e. searches for a parse tree by trying to build upon the root node S down to the leaves
- 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

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

Top-down Parsing

S

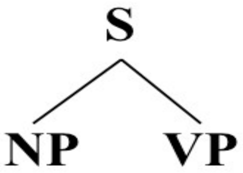
Grammar

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

Top-down Parsing



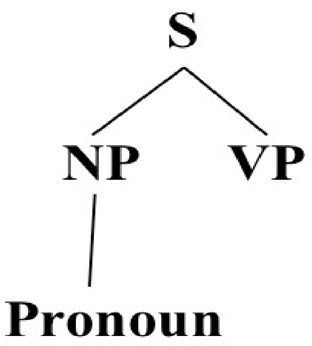
Grammar

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

Top-down Parsing



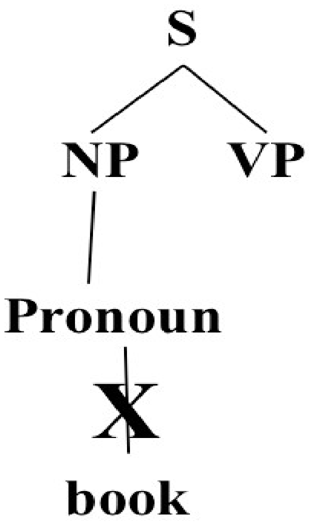
Grammar

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

Top-down Parsing



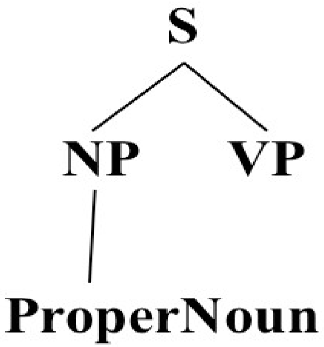
Grammar

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

Top-down Parsing



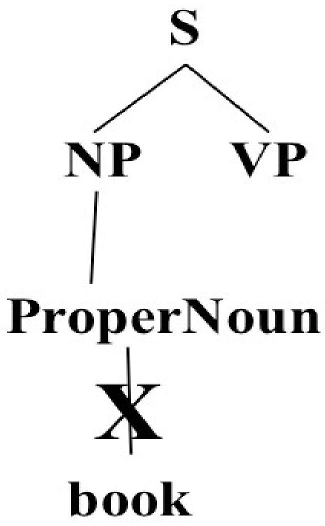
Grammar

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

Top-down Parsing



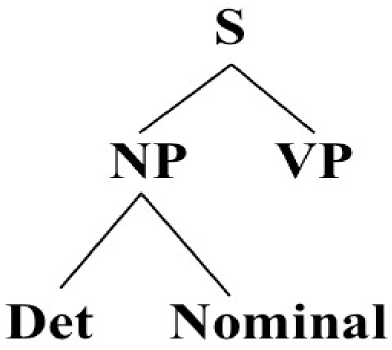
Grammar

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

Top-down Parsing



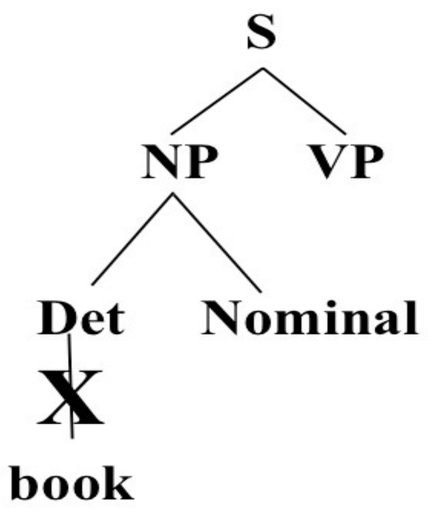
Grammar

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

Top-down Parsing



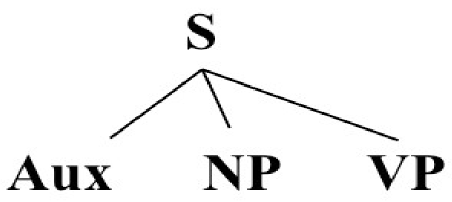
Grammar

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

Top-down Parsing



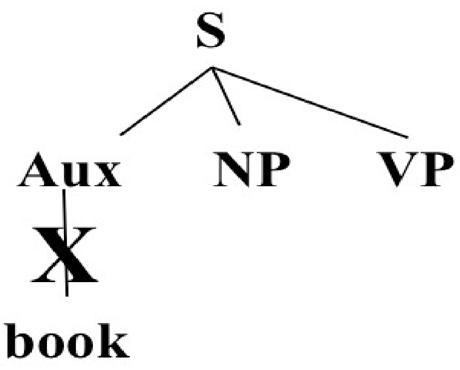
Grammar

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

Top-down 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

Top-down Parsing

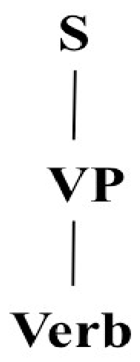
Grammar

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

Top-down Parsing



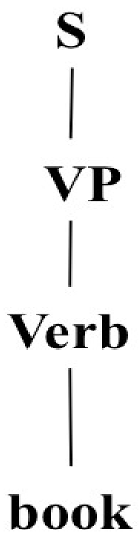
Grammar

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

Top-down Parsing



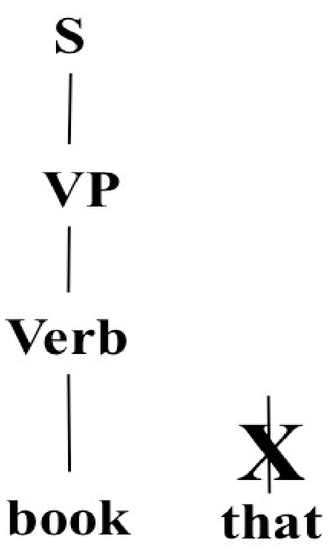
Grammar

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

Top-down Parsing



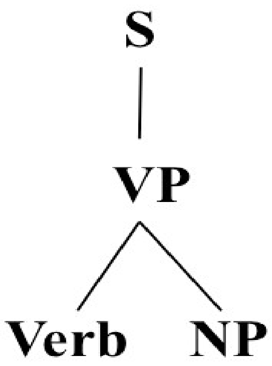
Grammar

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

Top-down Parsing



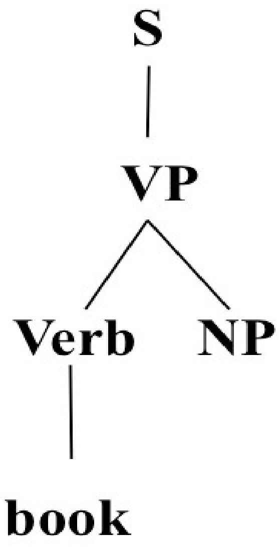
Grammar

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

Top-down Parsing



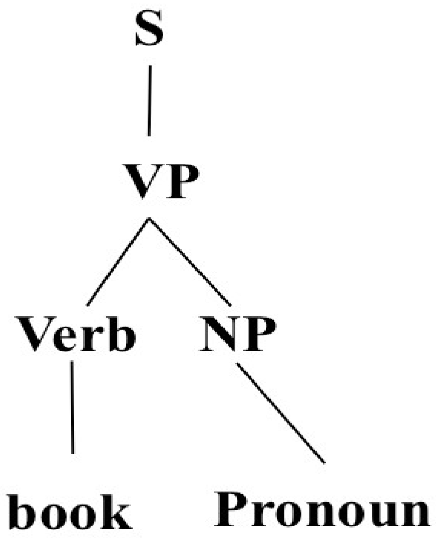
Grammar

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

Top-down Parsing



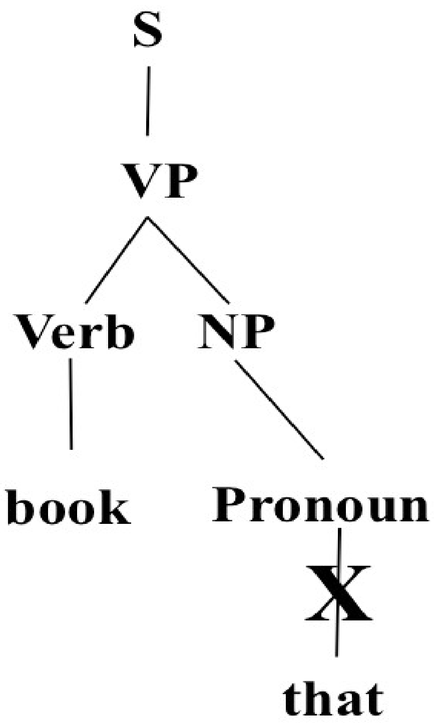
Grammar

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

Top-down Parsing



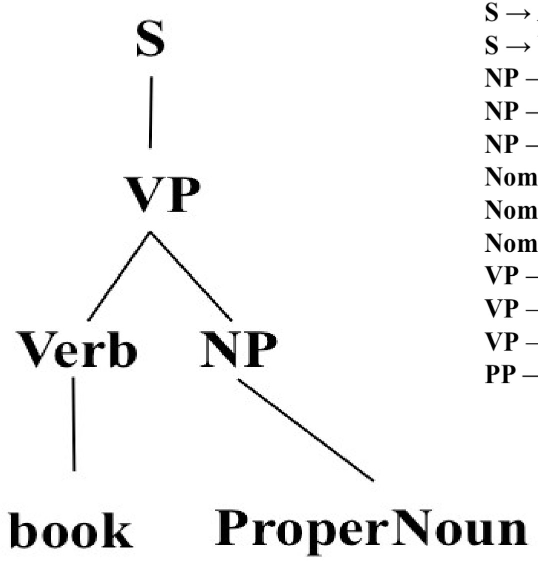
Grammar

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

Top-down Parsing



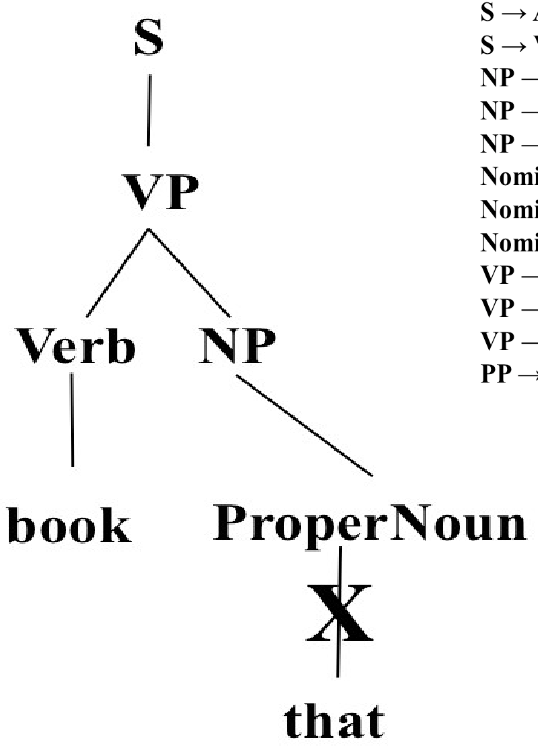
Grammar

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

Top-down Parsing



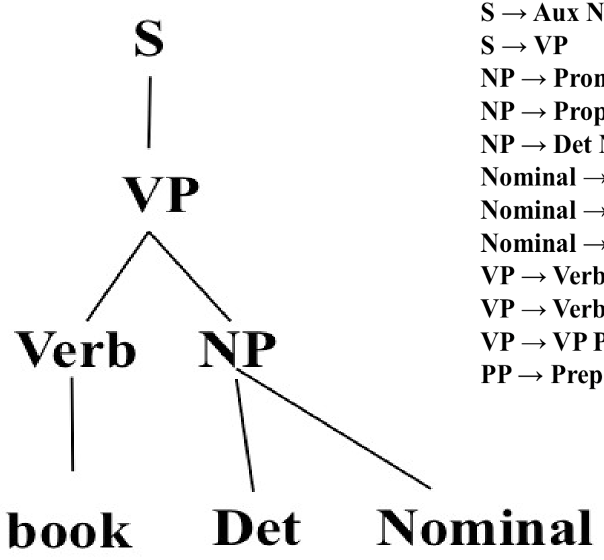
Grammar

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

Top-down Parsing



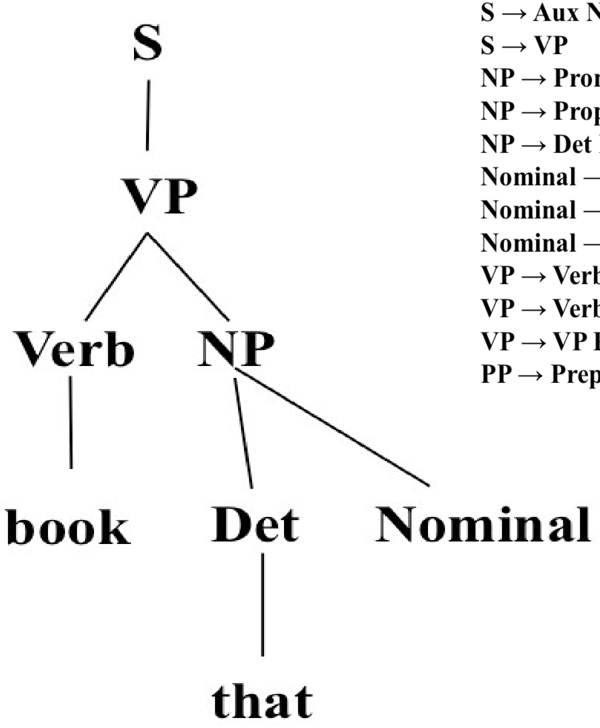
Grammar

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

Top-down Parsing



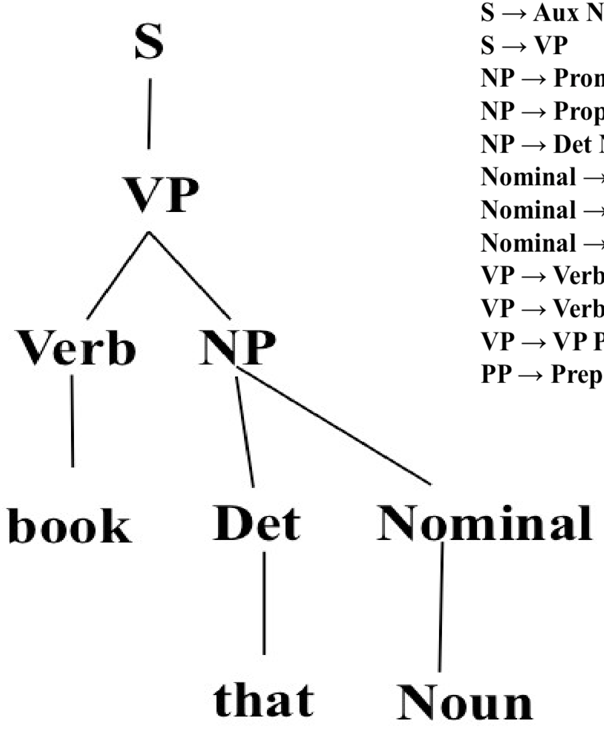
Grammar

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

Top-down Parsing



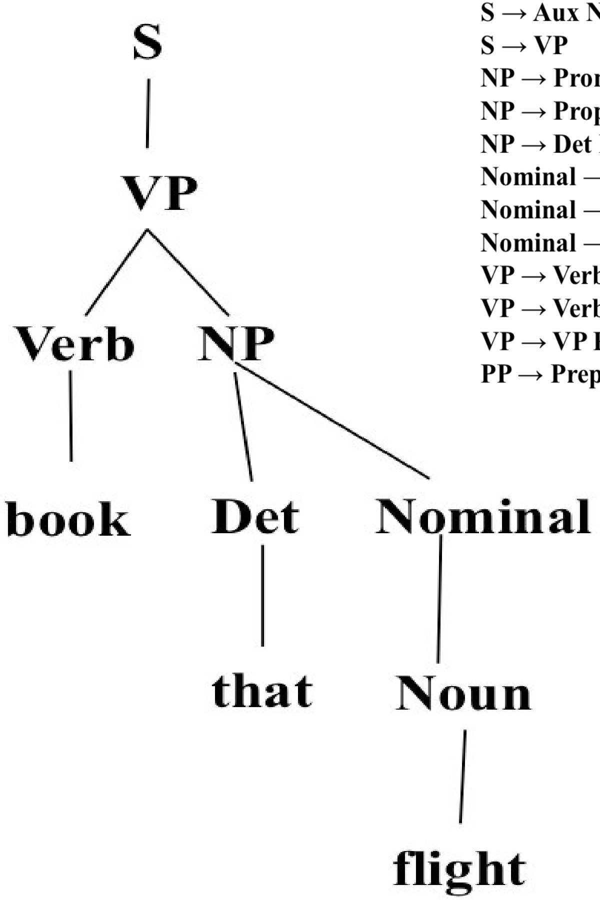
Grammar

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

Top-down 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

Bottom-up Parsing

- Bottom-up parser starts with the words in the input sentence and attempts to construct a parse tree in upwoard direction toward the root.
- At each step parser looks for the places in the parse-in-progress where the right-hand-side of some rule might fit.
- The parse is considered successful if the parser reduces the tree to the start symbol of the grammar.

Bottom-up Parsing Example

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

Bottom-up Parsing Example

Grammar

$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$
$S \rightarrow VP$
$NP \rightarrow Pronoun$
$NP \rightarrow Proper\text{-}Noun$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow Noun$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow Verb$
$VP \rightarrow Verb\ NP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Prep\ NP$

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

book **that** **flight**

Bottom-up Parsing Example

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

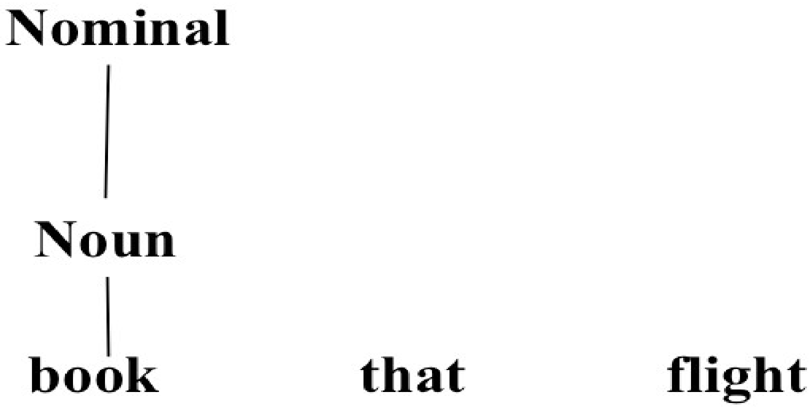
Bottom-up Parsing Example

Grammar

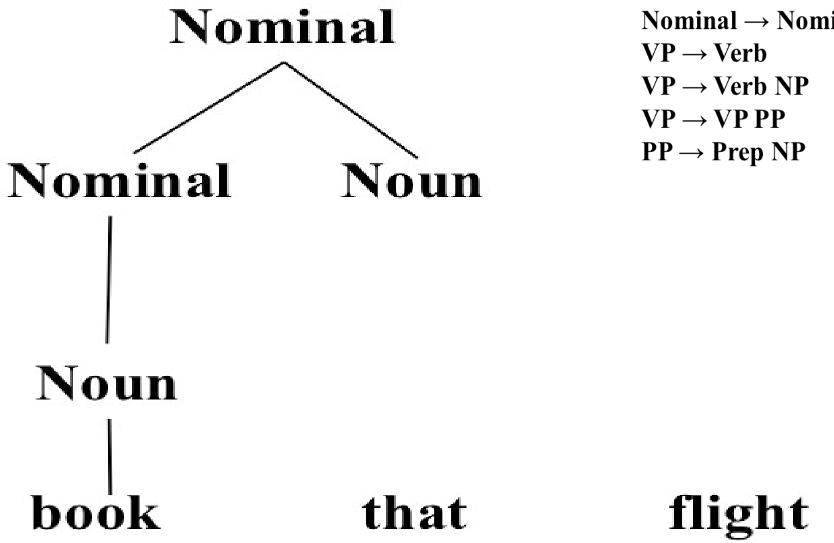
S → NP VP
S → Aux NP VP
S → VP
NP → Pronoun
NP → Proper-Noun
NP → Det Nominal
Nominal → Noun
Nominal → Nominal Noun
Nominal → Nominal PP
VP → Verb
VP → Verb NP
VP → VP PP
PP → Prep NP

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



Bottom-up Parsing Example



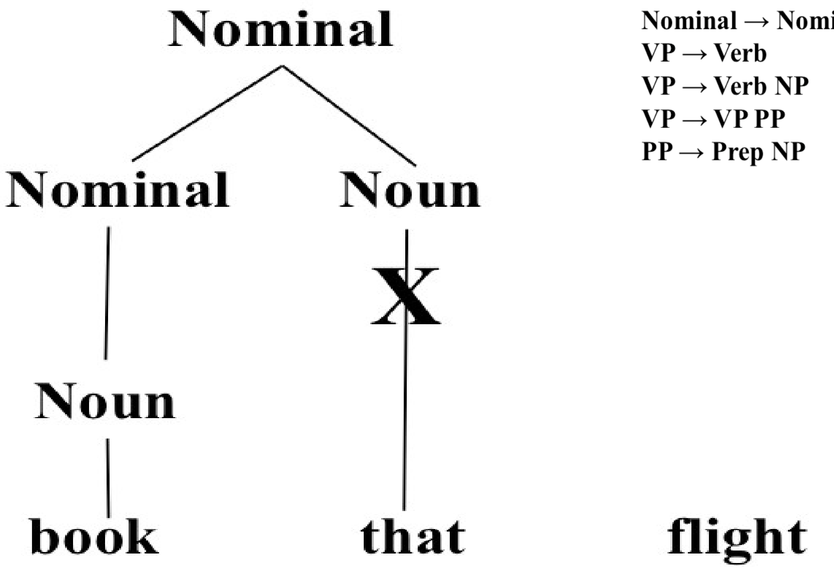
Grammar

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

Bottom-up Parsing Example



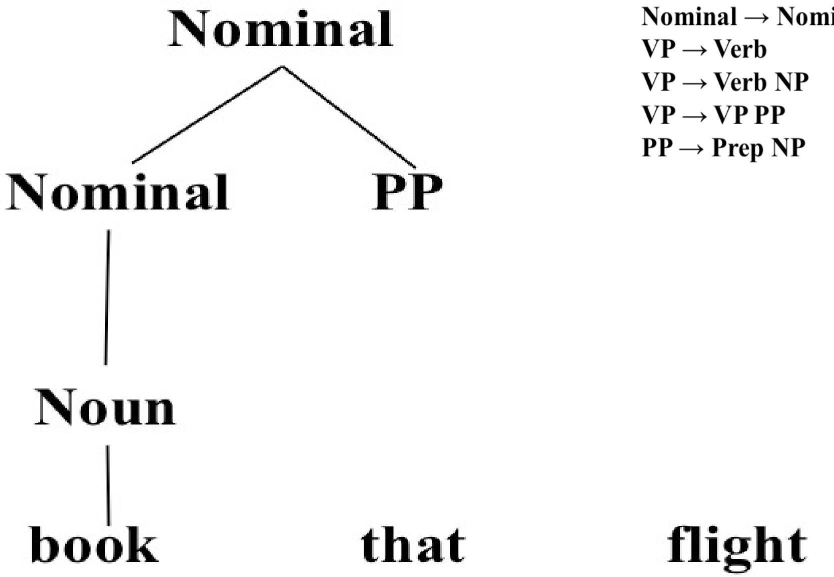
Grammar

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

Bottom-up Parsing Example



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

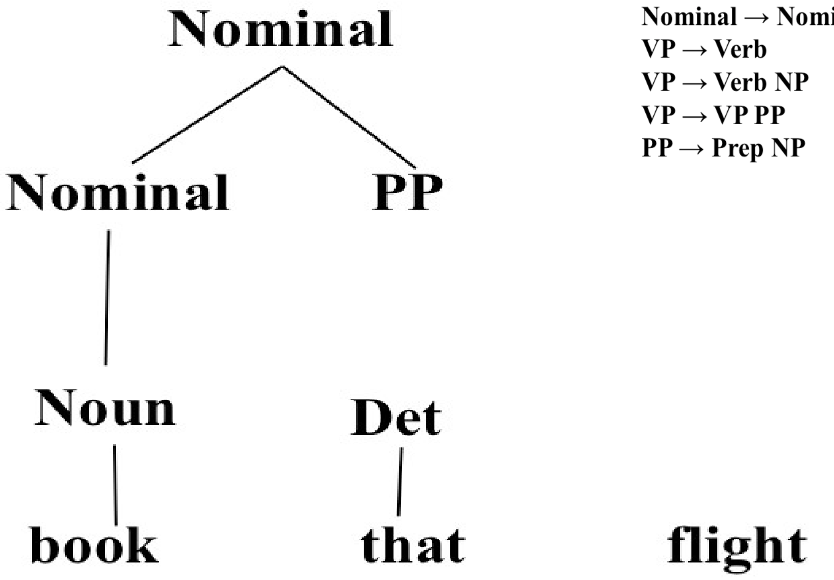
Bottom-up Parsing Example

Grammar

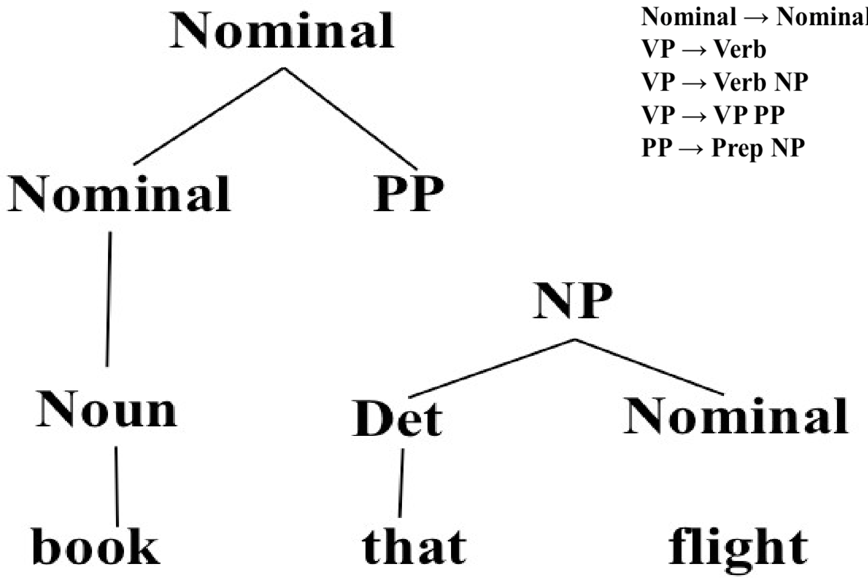
S → NP VP
S → Aux NP VP
S → VP
NP → Pronoun
NP → Proper-Noun
NP → Det Nominal
Nominal → Noun
Nominal → Nominal Noun
Nominal → Nominal PP
VP → Verb
VP → Verb NP
VP → VP PP
PP → Prep NP

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



Bottom-up Parsing Example



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

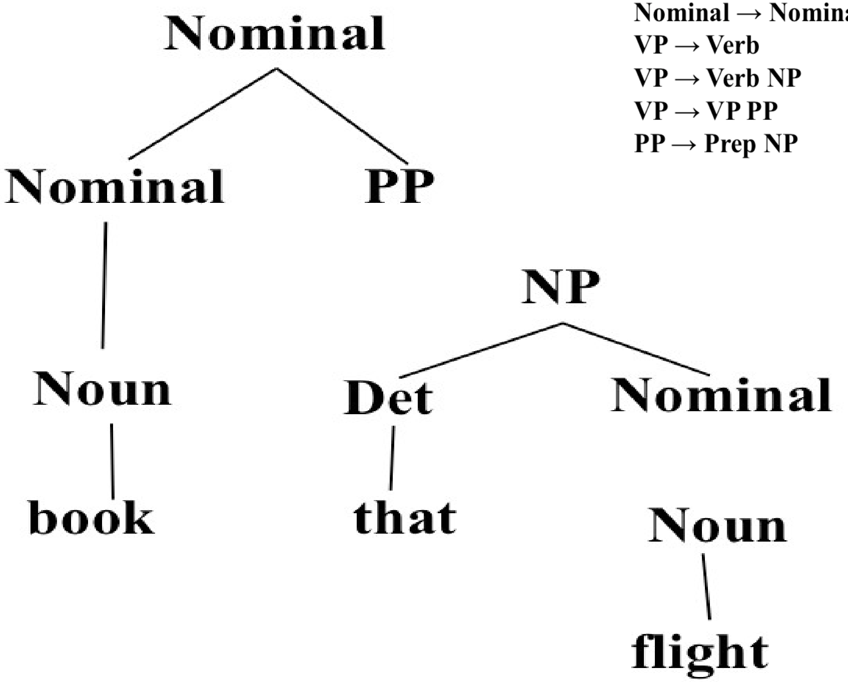
Bottom-up Parsing Example

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



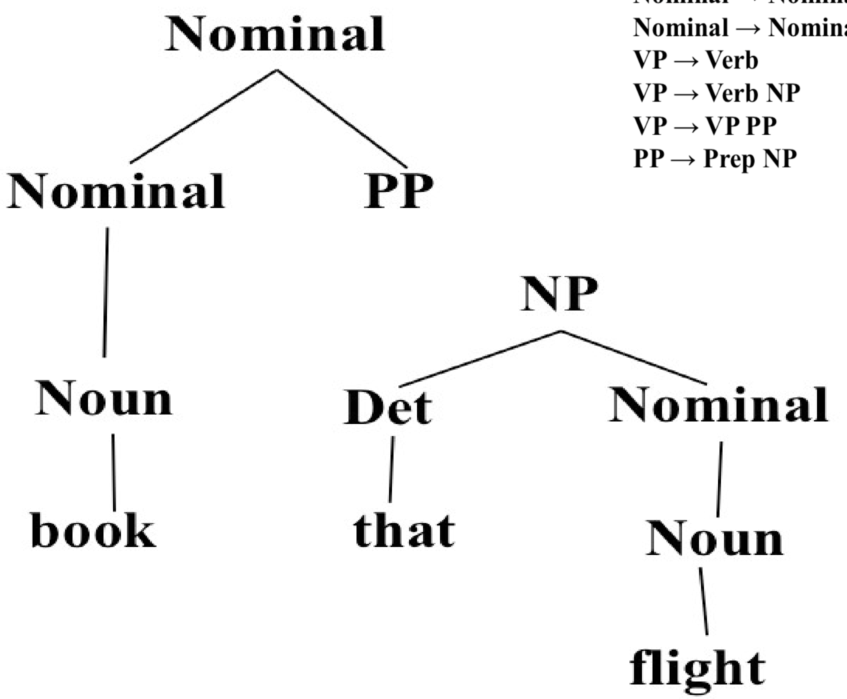
Bottom-up Parsing Example

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



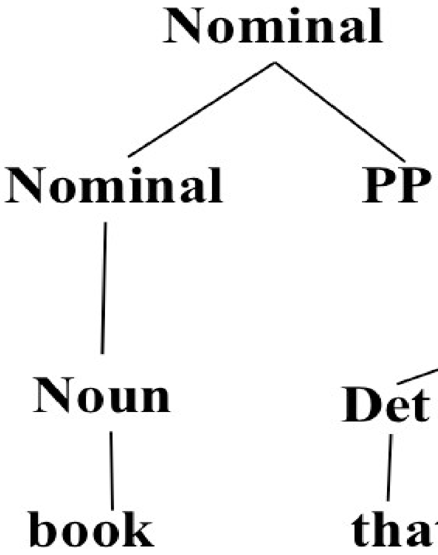
Bottom-up Parsing Example

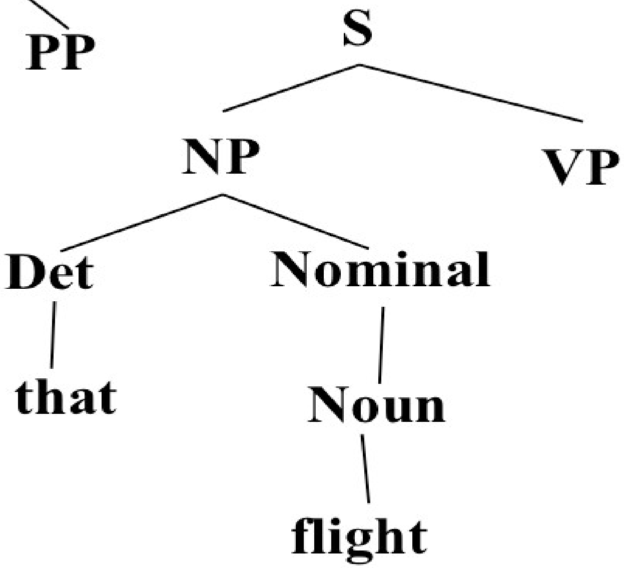
Grammar

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





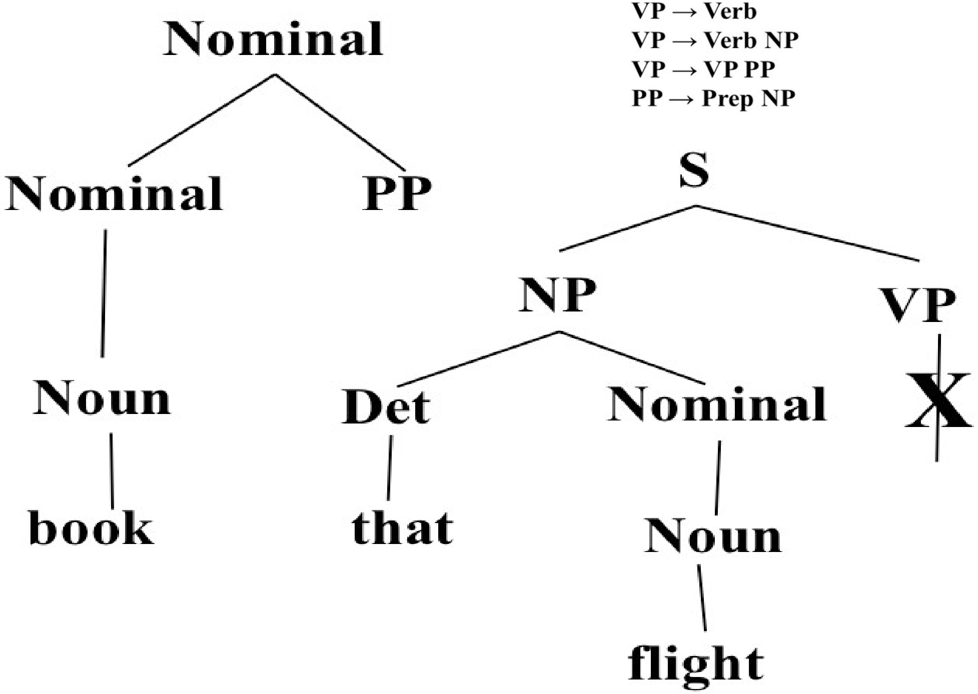
Bottom-up Parsing Example

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



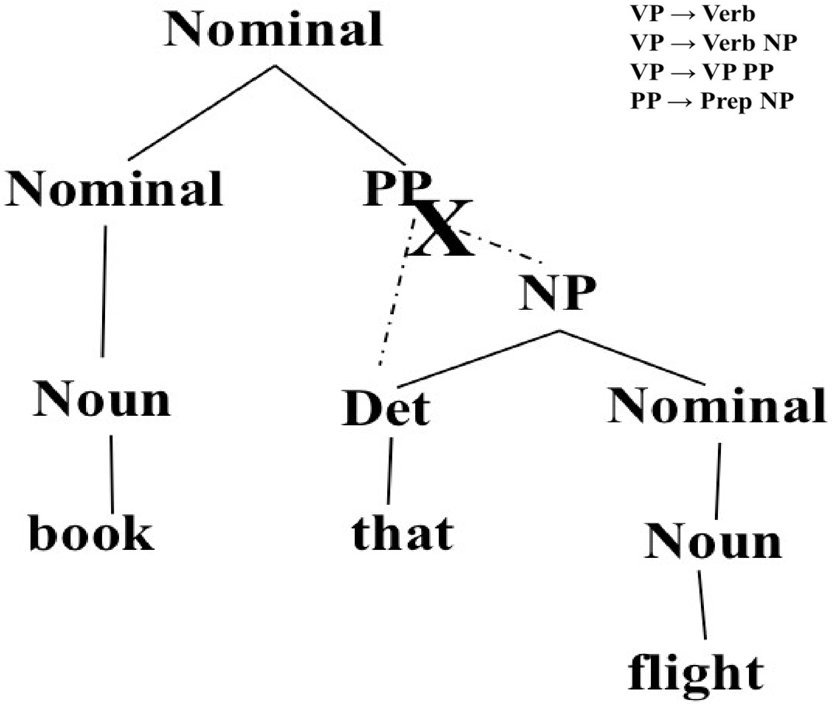
Bottom-up Parsing Example

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



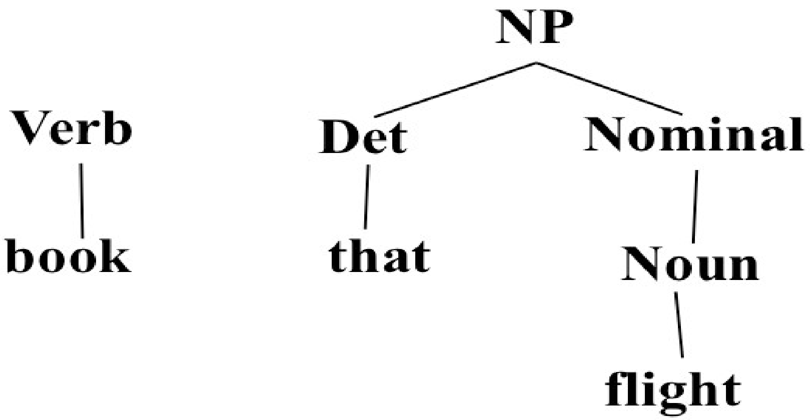
Bottom-up Parsing Example

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



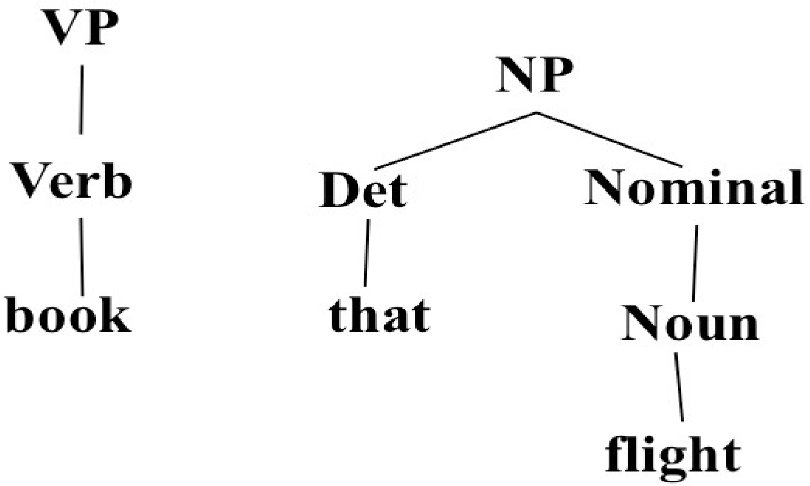
Bottom-up Parsing Example

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



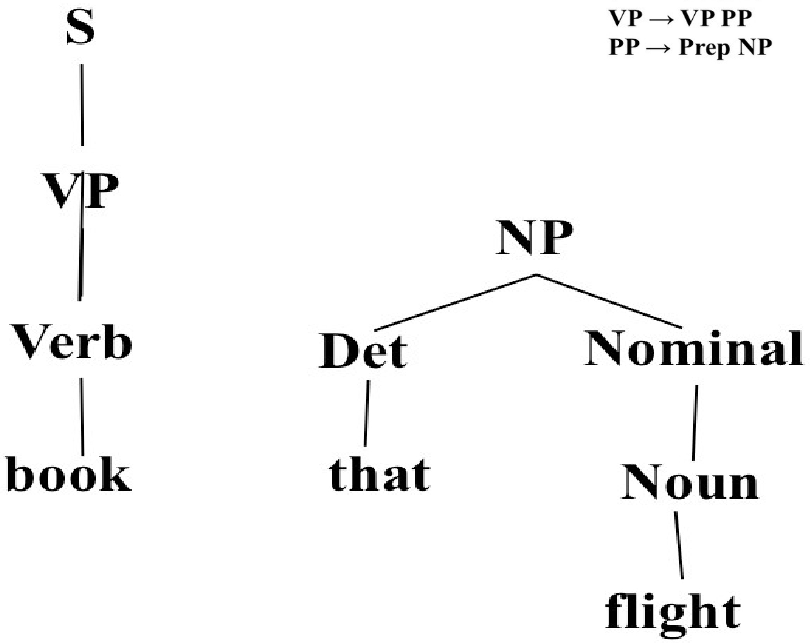
Bottom-up Parsing Example

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



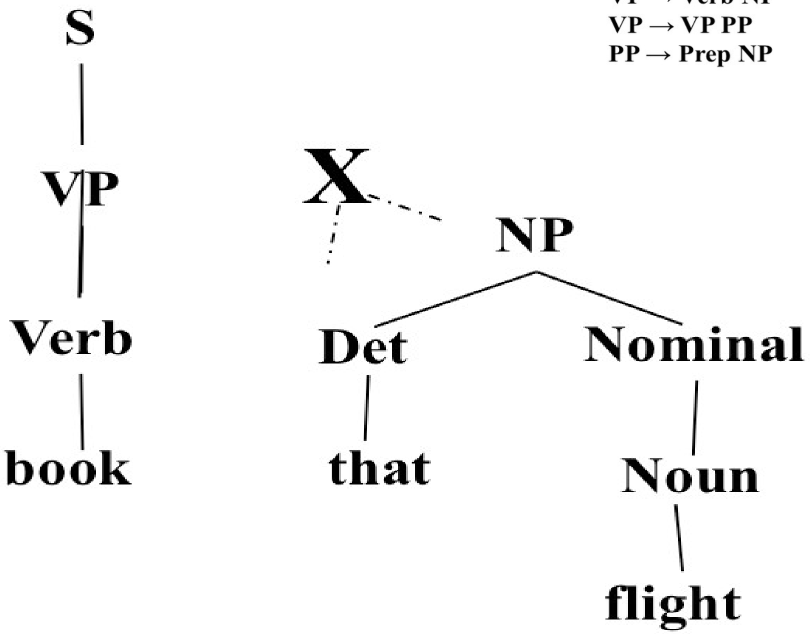
Bottom-up Parsing Example

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



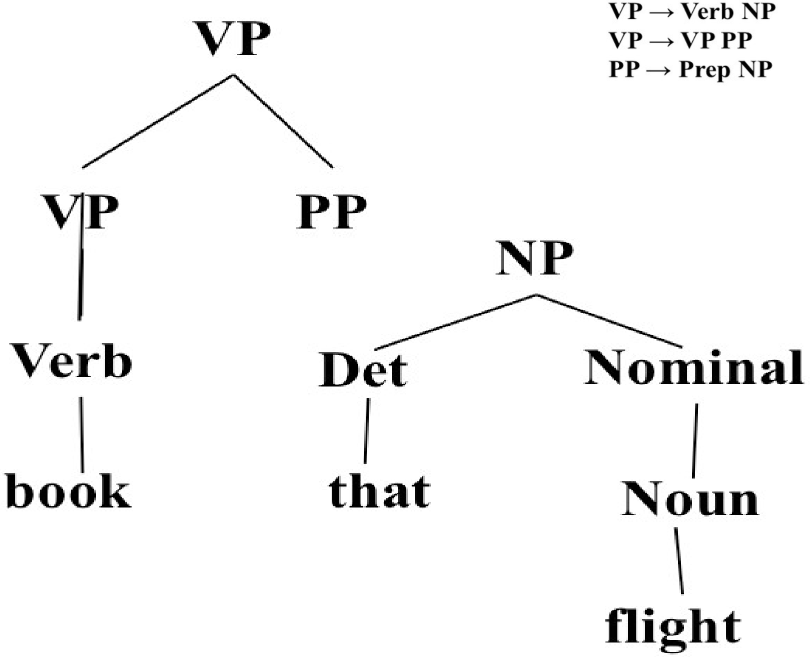
Bottom-up Parsing Example

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



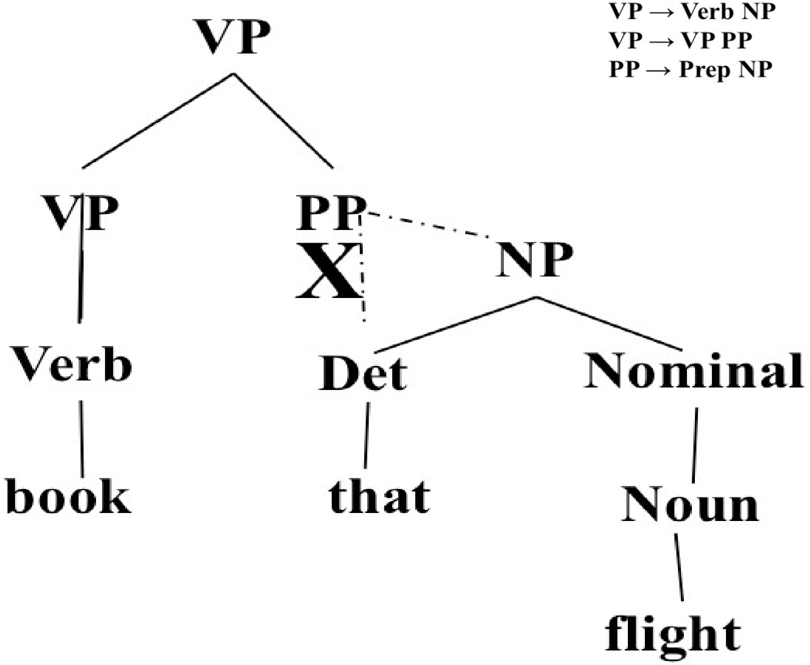
Bottom-up Parsing Example

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



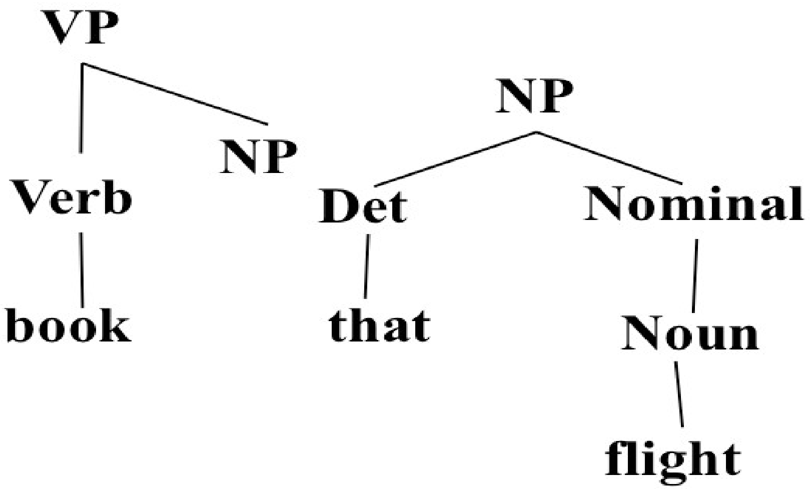
Bottom-up Parsing Example

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



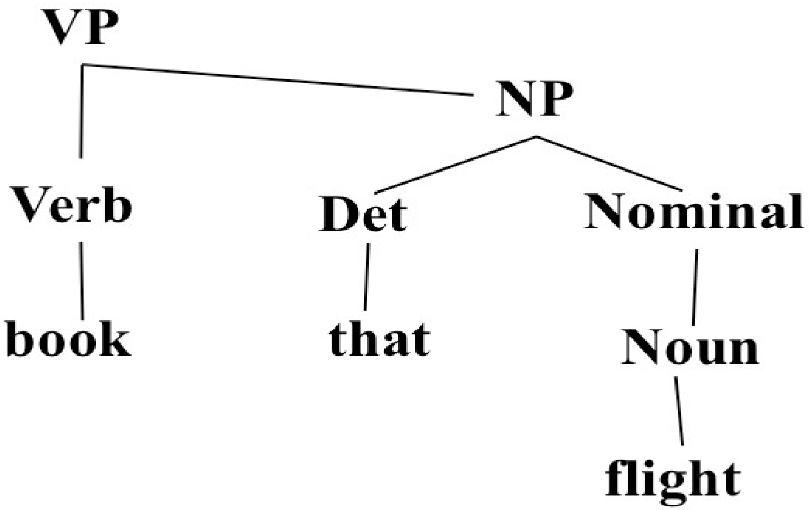
Bottom-up Parsing Example

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



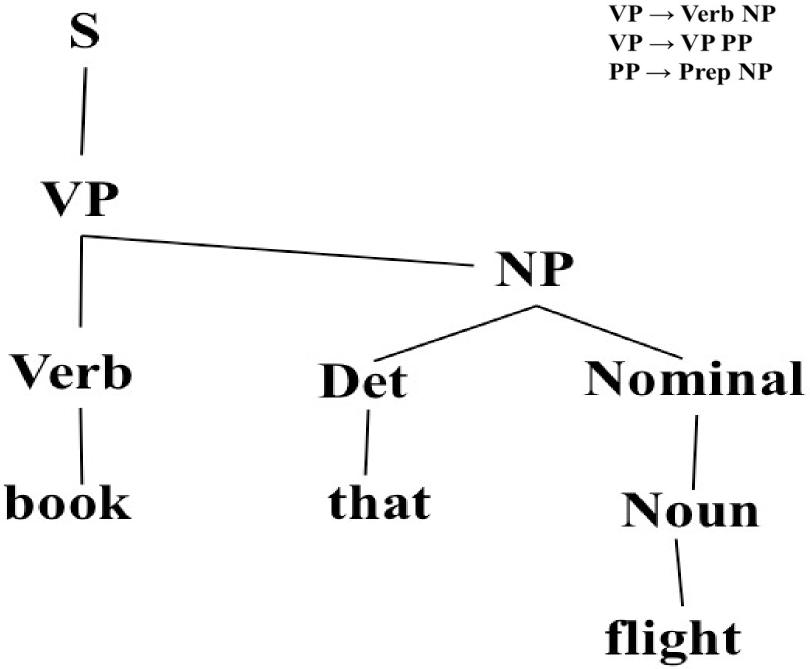
Bottom-up Parsing Example

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



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.

Basic Top-Down Parser

- It is a depth first, left to right search.

- The depth first approach expands the search space incrementally by one state at a time.

- At each step, the left most unexpected leaf nodes of the tree are expanded first using the relevant rules of the grammar.

- When a state arrives that is inconsistent with the input, the search continues by returning to the most recently generated and unexplored tree.

Basic Top-Down Parser

- Algorithm

1. Initialize agenda
2. Pick a state, let it be curr_state, from agenda
3. If (curr_state) represents a successful parse then return parse tree
   else if curr_stat is a POS then
   if category of curr_state is a subset of POS associated with curr_word
   then apply lexical rules to current state
   else reject
   else generate new states by applying grammar rules and push them into agenda
4. If (agenda is empty) then return failure
   else select a node from agenda for expansion and go to step 3.

Basic Top-Down Parser

- The algorithm maintains an agenda of search states.

- Each search state consists of partial trees and a pointer to the next input word in the sentence.

- The algrithm starts with the state at the front of the agenda and generates a set of new states by applying grammar to the left-most unexpanded node of the tree associated with it.

Basic Top-Down Parser

- The newly generated states are put on the front of the agenda.

- The process continues until either a successful parse tree is discovered or the agenda is empty.

- In any successful parse, the current input word must match the first word in the derivation of the node that is being expanded.

- The first word along the left side of the derivation is called the left corner of the tree.

Basic Top-Down Parser

- To utilize the left corner rule a table containing a list of all the valid left corner categories for each non-terminal of the grammar is constructed.

- While selecting a rule for expansion, the table is consulted to see if the non-terminal associated with the rule as a PoS associated with the current input.

- If not then the rule is not considered.

Basic Top-Down Parser

- Example:

$S \rightarrow NP\ VP$ | $S \rightarrow VP$

$NP \rightarrow Det\ Nominal$ | $NP \rightarrow Noun$

$NP \rightarrow Det\ Noun\ PP$ | $Nominal \rightarrow Noun$

$Nominal \rightarrow Noun\ Nominal$ | $VP \rightarrow Verb\ NP$

$VP \rightarrow Verb$ | $Det \rightarrow this|that|a|the$ | $PP \rightarrow Preposition\ NP$

$Verb \rightarrow sleeps|sings|open|paint$ | $Noun \rightarrow door|Hari$ | $Preposition \rightarrow with|to$

- Left corner of each grammar category

Category	Left Corners
S	Det,Noun,Verb
NP	Noun,Det
VP	Verb
PP	Preposition
Nominal	Noun

Basic Top-Down Parser - Disadvantages

- Left recursion
  - It causes the search to get stuck in an infinite loop.
  - This problem arises if the grammar is left recursive,
  - that is, it contains a non terminal A, which derives, in one or more steps, a string begining with the same non-terminal.
  - $A \overset{*}{\Rightarrow} A\beta$ for some β
- Structural Ambiguity
  - It occurs when a grammar assigns more than one parse to a sentence.
  - It occurs in many forms like attachment ambiguity and coordination ambiguity.
  - A sentence has attachment ambiguity if a constituent fits more than one position in a parse tree.
  - Coordination ambiguity occurs when it is not clear which phrases are being combined with a conjuction like ‘and’.
    - Example : “beautiful hair and eyes”
    - Which structure? [beutiful hair] and [eyes] or [beutiful hair] and [beutiful eyes]

Earley Parser

- It is an efficient parallel top-down parser using dynamic programming.

  It can handle recursive rules such as A$\rightarrow$AC without getting into an infinite loop
- The most important component of this algorithm is the Earley chart.
- Early chart has n+1 entries, where n is the number of words in the input.
- The chart contains a set of states for each word position in the sentence.
- The algorithm makes a left to right scan of input to fill the elements in this chart.
- It builds a set of states, that describes the condition of the recognition process at that point in the scan.

Earley Parser

- The states in each entry provides the following information.
  - A sub-tree corresponding to a grammar rule
  - Information about the progress made in completing the sub-tree.
  - Position of the sub-tree with respect to input.
- A state represented as a dotted rule and a pair of numbers representing starting position and the position of dot.
- This representation takes the form

  A $X_1 \ldots\ldots \bullet C \ldots\ldots X_m$, [i,j]

  - Where dot(.) represents the position in the rule's right hand side,
  - [i,j] represents where the state begins and where the dot lies.
  - A dot at the right end of the rule represents a successful parse of the associated non-terminal

Earley Parser

- The algorithm uses three operation to process states in the chart.
  - Predictor
  - Scanner
  - Completer
- The algorithm sequentially constructs the sets for each n+1 chart entries.
- Chart[0] is initialized with a dummy state $S' \rightarrow \bullet S$, [0,0]
- At each step one of the three operations are applicable depending on the state.
- The presence of a state $S \rightarrow \alpha$,[0,N] indicates a successful parse

Earley Parser

- Predictor
  - It generates new states representating potential expansion of the non-terminal in the left-most derivation.
  - A predictor is applied to every non terminal to the right of the dot.
  - The application of this operator results in the creation as many new states as there are grammar rules for this non terminal.
  - These new states are places into the same chart entry as the generating state.
  - If $A \rightarrow X_1 \ldots \bullet B \ldots X_m$, [i,j]

    Then for every rule of the form $B \rightarrow \alpha$, the operation adds to chart[j], the state
  - $B \rightarrow \bullet \alpha$,[j,j]

Earley Parser

- Scanner
  - A scanner is used when a state has a part of speech category to the right of the dot.
  - The scanner examines the input to see if the POS appearing to the right of the dot matches one of the POS of the current input.
  - If yes it creates the new state using the rule that allows the creation of the word with this POS.
  - It advances the pointer over the predicted input category and adds it to the next chart entry
  - If the state is $A \rightarrow \ldots \cdot a \ldots$, [i,j] and a is on of the POS associated with w_j, then it adds $a \rightarrow \ldots w_j \bullet$, [i,j+1] to chart [j+1]

Earley Parser

- Completer
  - The completer is used when the dot reaches the right end of the rule.
  - The completer identifies all previously generated states that expect this grammatical category at this position in the input and creates new states by advancing the dots over the expected category.
  - All the newly generated states are inserted in the current chart entry.
  - If $A \rightarrow \ldots \bullet$,[j,k] then the completer adds $B \rightarrow \ldots.A \bullet \ldots$,[i,k] to chart [k] for all states $B \rightarrow \ldots.A \bullet ..$,[i,j] in chart [j]
  - An item is added to a set only if it is not already in the set

Earley Parser-Example

- Trace the chart entries for the sentence “paint the door” using Earley parser with the following grammar. $S \rightarrow NP\ VP$, $S \rightarrow VP$, $NP \rightarrow Det\ Noun$, $NP \rightarrow Noun$, $VP \rightarrow Verb\ NP$, $VP \rightarrow Verb$, $Det \rightarrow the|this|a$, $Noun \rightarrow paint$, $Verb \rightarrow paint$

- Ans-
  - Chart[0]:
  - S0 $S' \rightarrow .S$ [0,0]
  - S1 $S \rightarrow .NP\ VP$ [0,0]
  - S2 $S \rightarrow .VP$ [0,0]
  - S3 $NP \rightarrow .Det\ Noun$ [0,0]
  - S4 $NP \rightarrow .Noun$ [0,0]
  - S5 $VP \rightarrow .Verb\ NP$ [0,0]
  - S6 $VP \rightarrow .Verb$ [0,0]

Earley Parser-Example

- Sentence "paint the door", Grammar. S → NP VP, S → VP, NP → Det Noun, NP → Noun, VP → Verb NP, VP → Verb, Det → the|this|a, Noun → paint, Verb → paint

  Chart[1]:

  - S7 Noun → paint. [0,1]
  - S8 Verb → paint. [0,1]
  - S9 NP → Noun. [0,1]
  - S10 VP → Verb. [0,1]
  - S11 VP → Verb. NP [0,1]
  - S12 S → NP. VP [0,1]
  - S13 S → VP. [0,1]
  - S14 NP → .Det Noun [1,1]
  - S15 NP → .Noun [1,1]
  - S16 VP → .Verb [1,1]
  - S17 VP → .Verb NP [1,1]

Earley Parser-Example

- Sentence "paint the door", Grammar. S → NP VP, S → VP, NP → Det Noun, NP → Noun, VP → Verb NP, VP → Verb, Det → the|this|a, Noun → paint|door, Verb → paint

Chart[2]:

- S18 Det → the. [0,2]
- S19 NP → Det. Noun [1,2]

Chart[3]:

- S20 Noun → door. [1,3]
- S21 NP → Det Noun. [1,3]
- S22 VP → Verb NP. [0,3]
- S23 S → NP. VP [0,3]
- S24 VP → .Verb NP [3,3]
- S25 VP → .Verb [3,3]
- S26 S → VP. [0,3]

Earley Parser-Question

- Q1.Sentence “pooja sings a song”, Grammar. S → NP VP, S → VP, NP → Det Noun, NP → Noun, VP → Verb NP, VP → Verb, Det → the|this|a, Noun → pooja|song, Verb → paint|sings Derive chart[0], chart[1], chart[2].

- Q2.Derive the chart[0] and chart[1] entries for the below-given sentence using the Earley Parser.

- Grammar rules:

  S-> NP VP, S-> VP, S-> Aux NP VP

  NP-> Det Noun, NP-> Noun

  VP-> Verb NP, VP-> Verb, VP-> verb PP

  PP-> prep NP

  Example sentence: "Book that slot"

CYK Parser

- The CYK (Cocke-Younger-Kasami) parser is a dynamic programming parsing algorithm.
- It follows a bottom-up approach in parsing.
- It builds a parse tree incrementally.
- Each entry in a table is based on previous entries.
- The process is iterated until the entire sentence has been parsed.

CYK Parser

- It requires normalizing the grammar
- Grammar must be converted to Chomsky normal form (CNF) in which all productions must have
  - Either, exactly two non-terminals on the RHS
  - Or, 1 terminal symbol on the RHS
- A CFG is in CNF if all the rules are of only two forms:

  $A \rightarrow BC$

  $A \rightarrow w$, where w is a word

CYK Parser

- It requires normalizing the grammar
- Grammar must be converted to Chomsky normal form (CNF) in which all productions must have
  - Either, exactly two non-terminals on the RHS
  - Or, 1 terminal symbol on the RHS
- A CFG is in CNF if all the rules are of only two forms:

  $A \rightarrow BC$

  $A \rightarrow w$, where w is a word
- Parse bottom-up storing phrases formed from all substrings in a triangular table (chart)

Converting to CNF

Original 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
Pronoun → I | he | she | me
Noun → book | flight | meal | money
Verb → book | include | prefer
Proper-Noun → Houston | NWA

Converting to CNF

Original 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
Pronoun → I | he | she | me
Noun → book | flight | meal | money
Verb → book | include | prefer
Proper-Noun → Houston | NWA

Chomsky Normal Form

S → NP VP
S → X1 VP
X1 → Aux NP
S → book | include | prefer
S → Verb NP
S → VP PP
NP → I | he | she | me
NP → Houston | NWA
NP → Det Nominal
Nominal → book | flight | meal | money
Nominal → Nominal Noun
Nominal → Nominal PP
VP → book | include | prefer
VP → Verb NP
VP → VP PP
PP → Prep NP
Pronoun → I | he | she | me
Noun → book | flight | meal | money
Verb → book | include | prefer
Proper-Noun → Houston | NWA

CYK Algorithm

- Let n be the number of words in the input. Think about n + 1 lines separating them, numbered 0 to n .
- x_{ij} will denote the words between line i and j
- We build a table so that x_{ij} contains all the possible non-terminal spanning for words between line i and j .
- We build the Table bottom-up

CYK Example

a 1	pilot 2	likes 3	flying 4	planes 5

$S \longrightarrow NP\ VP$
$VP \longrightarrow VBG\ NNS$
$VP \longrightarrow VBZ\ VP$
$VP \longrightarrow VBZ\ NP$
$NP \longrightarrow DT\ NN$
$NP \longrightarrow JJ\ NNS$
$DT \longrightarrow a$
$NN \longrightarrow pilot$
$VBZ \longrightarrow likes$
$VBG \longrightarrow flying$
$JJ \longrightarrow flying$
$NNS \longrightarrow planes$

CYK Example

a 1	pilot 2	likes 3	flying 4	planes 5
DT	NP	-	-	S S
	NN	-	-	-
		VBZ	-	VP VP
			JJ VBG	NP VP
				NNS

$S \longrightarrow NP\ VP$
$VP \longrightarrow VBG\ NNS$
$VP \longrightarrow VBZ\ VP$
$VP \longrightarrow VBZ\ NP$
$NP \longrightarrow DT\ NN$
$NP \longrightarrow JJ\ NNS$
$DT \longrightarrow a$
$NN \longrightarrow pilot$
$VBZ \longrightarrow likes$
$VBG \longrightarrow flying$
$JJ \longrightarrow flying$
$NNS \longrightarrow planes$

CYK Algorithm-Home Exercise

- Use CKY algorithm to find the parse tree for “Book the flight through Houston” using the CNF form shown in the previous slide.

Probabilistic Parsing

- Statistcal parser requires a corpus of hand-parsed text.
- One such corpora is Penn tree-bank which is a large corpus of articles from the Wall Street Journal.
- A statistcal parser works by assigning probabilities to possible parses of a sentence and returning the most likely parse as the final one.
- In order to construct a statistcal parser
  - First find all possible parses of a sentence
  - Then assign probabilities to them
  - Finally return the most probable parse
- One such probabilistic parser is PCFG(Probabilistic Context-free Grammar)

Benifit of Probabilistic Parsing

- The first benifit that a probabilistic parser offers is removal of ambiguity from parsing, by taking a parse tree with highest probability.
- Another benifit is that it is more efficient because instead of searching the whole search space here probabilities are used to guide the search process.
- PCFG
  - A PCFG is a CFG in which avery rule is assigned a probability.
  - It extends the CFG by augmenting each rule $A \rightarrow \alpha$ in set of productions P, witha conditional probability p:

$$A \rightarrow \alpha \quad [p]$$

Where p gives the probability of expanding a constituent using the rule $A \rightarrow \alpha$

PCFG

- PCFG: G = (T, N, S, R, P)
  - 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 $\rightarrow$ γ , X $\in$ N and γ $\in$ $(T \cup N)^*$
- P(R) gives the probability of each rule.

$$\forall X \in N, \sum_{X \rightarrow \gamma \in R} P(X \rightarrow \gamma) = 1$$

A simple PCFG

- An example of PCFG is shown below.
- We can verify that for each non-terminal, the sum of probabilities is 1.

S	→	NP VP	1.0	NP	→	NP PP	0.4
VP	→	V NP	0.7	NP	→	*astronomers*	0.1
VP	→	VP PP	0.3	NP	→	*ears*	0.18
PP	→	P NP	1.0	NP	→	*saw*	0.04
P	→	*with*	1.0	NP	→	*stars*	0.18
V	→	*saw*	1.0	NP	→	*telescope*	0.1

Estimating Rule Probabilities

- One way to estimate probabilities for a PCFG is to manually construct a corpus of a parse tree for a set of sentences, and
- Then estimate the probabilities of each rule being used by counting over the corpus.
- The MLE estimate for a rule $A \rightarrow \alpha$ is given by the expression

$$P_{MLE}(A \rightarrow \alpha) = \frac{Count(A \rightarrow \alpha)}{\sum_{\alpha} Count(A \rightarrow \alpha)}$$

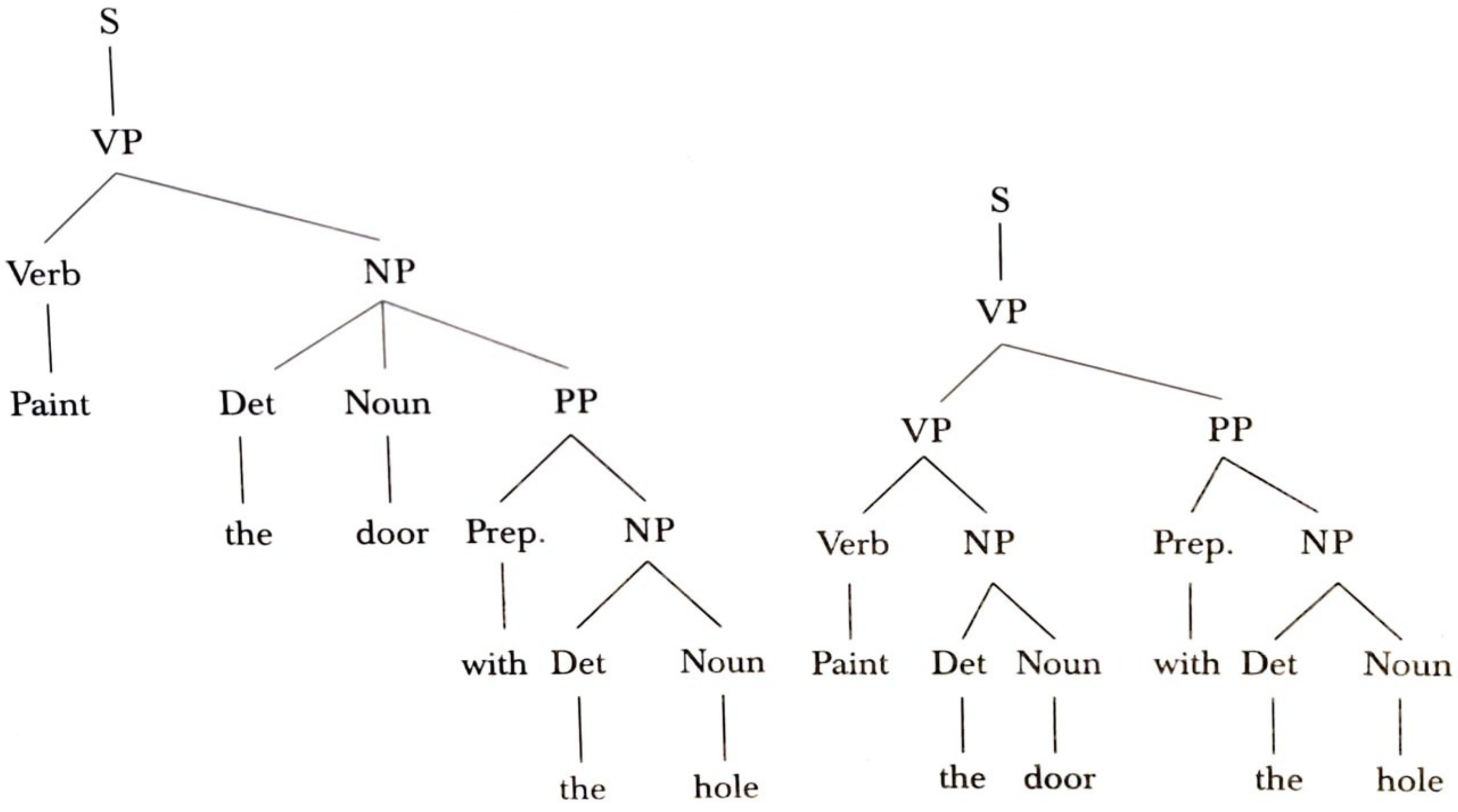
Estimating Rule Probabilities

- Example
  - Consider the following grammar. Construct two parse tree for the sentence “Paint the door with the hole”.Using these two parse tree as the corpus estimate the probabilities of eacg rule in the grammar.

$S \rightarrow VP$
$NP \rightarrow Det\ Noun\ PP$
$NP \rightarrow Det\ Noun$
$VP \rightarrow Verb\ NP$
$VP \rightarrow VP\ PP$
$Det \rightarrow the$
$Noun \rightarrow hole$
$Noun \rightarrow door$
$Prep \rightarrow with$
$Verb \rightarrow Paint$

Estimating Rule Probabilities

- Solution
  - Two possible parse tree are



Estimating Rule Probabilities

- Solution
  - The MLE estimates are

Rule	Count (A → α)	Count A	MLE estimates
S → VP	2	2	1
NP → Det Noun PP	1	4	0.25
NP → Det Noun	3	4	0.75
VP → Verb NP	2	3	0.66
VP → VP PP	1	3	0.33
Det → the	2	2	1
Noun → hole	2	4	0.5
Noun → door	2	4	0.5
Prep → with	1	1	1
Verb → Paint	1	1	1

Question?

- Use the below-given grammar rules and construct two possible parse trees for the example sentence given. Use these two trees as the training data, and obtain the maximum Likelihood estimates for the grammar rules used in the trees. Also, find out which parse tree will lead to a more correct interpretation.

- Grammar rules:
- S->NP VP,
- VP->VP PP, VP->verb NP, VP-> verb
- NP->Det Noun, NP->Det Noun PP, NP-> Noun
- PP->prep. NP
- Example sentence: "Pooja saw the boy with a telescope"

What to do with these probabilities?

- These probabilities can be used to calculate the probability of the parse tree and the probability of a sentence.
- Probability of a parse tree p(t) is the product of the probabilities of the rules used to generate it.
- Probability of a sentence p(s) is the sum of the probabilities of the trees which have that string as their yield.
- For the above example the probability of parse trees are calculated as

  p(t1)= 0.2 * 0.5 * 0.2 * 0.2 * 0.35 * 0.25 * 1.0 * 0.25 * 0.4 * 0.35 * 0.25 = 0.0000030625

  p(t2)=0.2 * 0.2 * 0.5 * 0.2 * 0.4 * 0.35 * 0.25 * 1.0 * 0.25 * 0.4 * 0.35 * 0.25 = 0.000001225
- Thus the probability of the sentence “Paint the door with the hole” is

  p(s)=p(t1)+p(t2)= 0.0000030625 + 0.000001225 = 0.0000042875

What to do with these probabilities?

- These probabilities can be used to calculate the probability of the parse tree and the probability of a sentence.
- Probability of a parse tree p(t) is the product of the probabilities of the rules used to generate it.
- Probability of a sentence p(s) is the sum of the probabilities of the trees which have that string as their yield.
- For the above example the probability of parse trees are calculated as

  p(t1)= 0.2 * 0.5 * 0.2 * 0.2 * 0.35 * 0.25 * 1.0 * 0.25 * 0.4 * 0.35 * 0.25 = 0.0000030625

  p(t2)=0.2 * 0.2 * 0.5 * 0.2 * 0.4 * 0.35 * 0.25 * 1.0 * 0.25 * 0.4 * 0.35 * 0.25 = 0.000001225
- Thus the probability of the sentence “Paint the door with the hole” is

  p(s)=p(t1)+p(t2)= 0.0000030625 + 0.000001225 = 0.0000042875

Problems with PCFG

- The first problem lies in the independence assumption.
  - Structural Independency: the expansion of any one non-terminal is independent of any other nonterminal
  - Each rule is independent of each other rule
  - But the choice of how a node expands is dependent on the location of the node in the parse tree, e.g.,

    NP → Pronoun or NP → Det Noun
  - These dependencies are not captured by a PCFG
- NP is a subject in a sentence?

  "Talk about topic or old information"

  NP is an object in a sentence?

  "Introduce new referents"

Problems with PCFG

- Lexical independency: lack of sensitivity to lexical information or words
- Lexical information in PCFGs can only be represented via the probability of pre-terminal nodes (Verb, Noun, Det) to expanded lexically
- But the lexical information plays an important role in selecting the correct parsing, e.g., the ambiguous prepositional phrase attachment
- Two structurally different parses that use the same rules will have the same probability under a PCFG, making it difficult to identify the correct or most probable parse.
- This however requires a model which captures lexical dependency statistics for different word.
- One such model is Lexicalization

Indian languages

- Not all natural languages have the same charecterstics.
- CFG is used for English language.
- But CFG may not be a viable choice for other languages like Indian languages.
- The majority of the Indian languges are free word order.
- By free word order language we mean that the order of words in a sentence can be changed without leading to a grammatically incorrect sentence.
- For example:

सबा खाना खाती है खाना सबा खाती है

- Both are valid Hindi sentences meaning Saba eats food.
- The CFG used for parsing English is positional .
- It can be used to model language in which a position of the constituents carries useful information, but it fails to model free word order languages.

Unit -IV

Semantic Analysis

- Semantics is associated with the meaning of language.
- Semantic analysis involves mapping of natural language utterances to some representation of meaning.
- Semantic analysis is concerned with creating representations for the meaning of linguistic inputs.
- Semantics can be divided into two parts
  - The study of the meaning of individual words(lexical semantics)
  - The study of how individual words combine to give meaning to a sentence.
- The principle of semantic compositionality (Freg’s principle) states that the meaning of the whole is comprised of the meanings of its parts.

Lexicalization

- In PCFG, the chance of a non-terminal expanding using a particular rule is independent of the actual words involved.
- But this assumption is not resonable.
- Words do affect the choice of the rule.
- Investigations suggest that the probabilities of various common sub-categorization frames differ depending on the verb that heads the verb phrase.
- This suggests the need for lexicalization i.e. involvement of actual words in the sentences, to decide the structure of the parse tree.
- This model of lexicalization is based on the idea that there are strong lexical dependencies between heads and their dependents.
- Exaple, between a verb and a noun phrase object, between head noun and its modifiers etc.