Artificial Intelligence

Open Elective

Module 5: NLP CH15

Dr. Santhi Natarajan
Associate Professor
Dept of AI and ML
BMSIT, Bangalore

Contents

Introduction

NLP

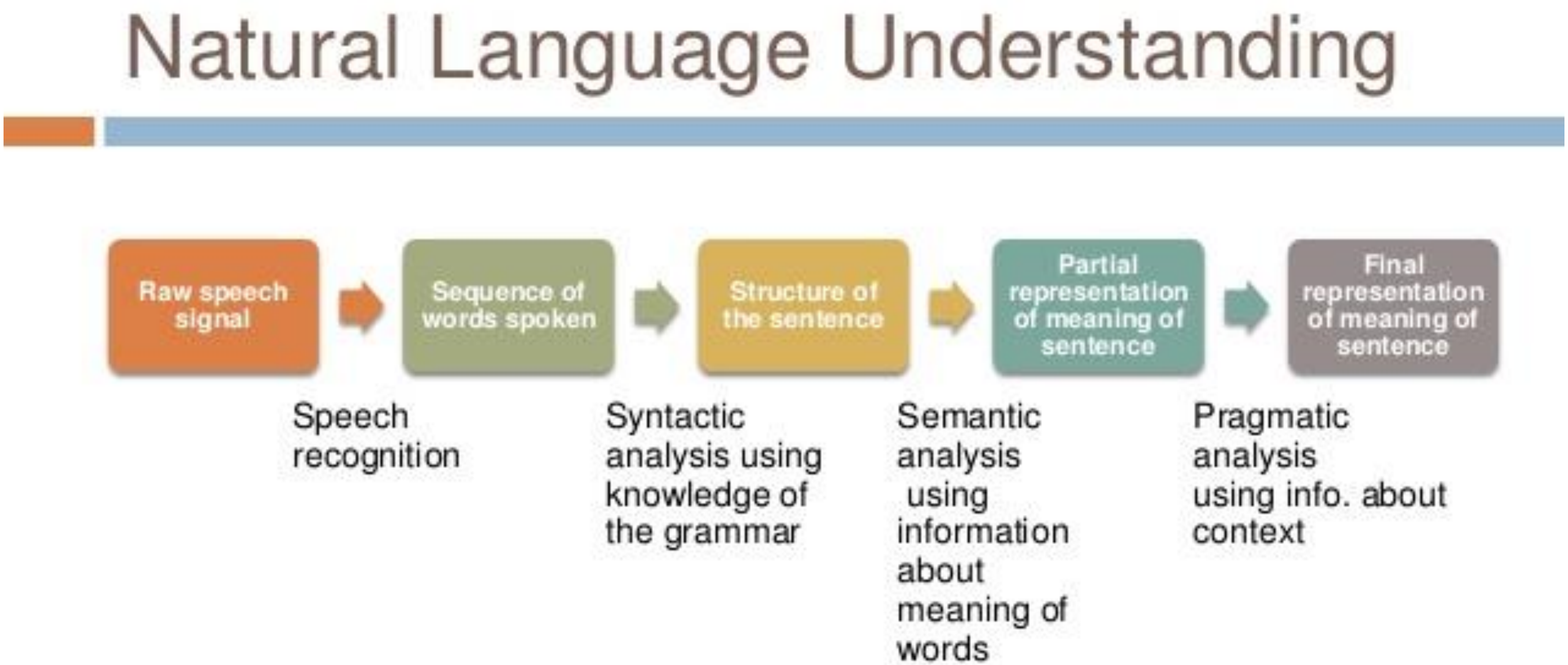
- Natural Language Understanding
  - Taking some spoken/typed sentence and working out what it means
- Natural Language Generation
  - Taking some formal representation of what you want to say and working out a way to express it in a natural human language like English
- Fundamental goal: deep understand of broad language
  - Not just string processing or keyword matching
- Target end systems
  - speech recognition, machine translation, question answering…
  - spelling correction, text categorization…

Introduction

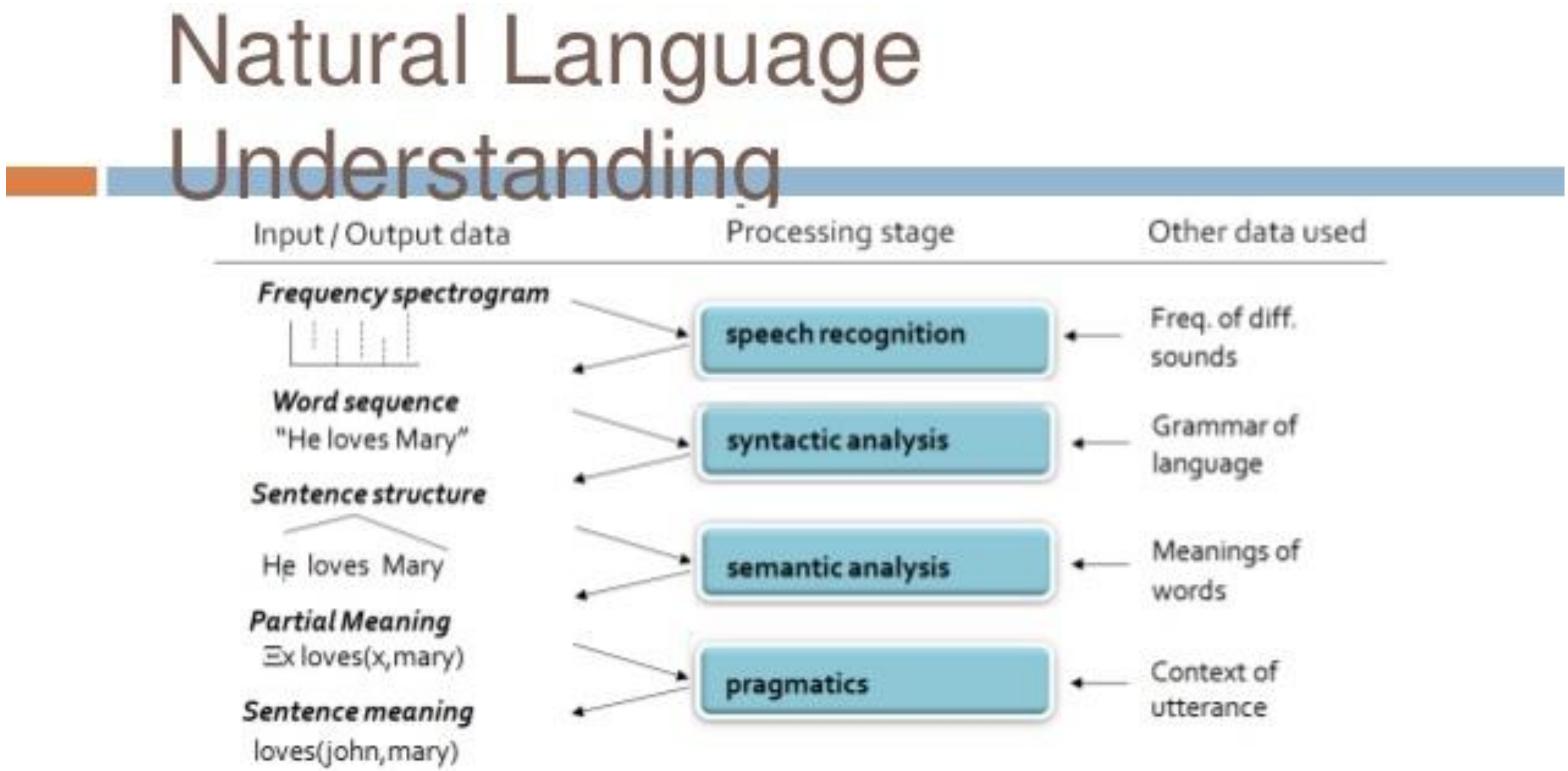
Applications

- Text Categorization - classify documents by topics, language, author, spam filtering, sentiment classification (positive, negative)
- Spelling & Grammar Corrections
- Speech Recognition
- Summarization
- Question Answering
- Better search engines
- Text-to-speech
- Machine aided translation
- Information Retrieval
  - Selecting from a set of documents the ones that are relevant to a query
- Extracting data from text
  - Converting unstructured text into structure data

Introduction



Introduction



Property of Language

Problem: Sentences are incomplete descriptions of ideas they want to convey.

Advantage: Language allows speakers to be as vague or as precise about anything they want to convey. Speakers can leave out things that they assume the listener knows already.

Some children visited us today.

↓

- Some children visited us today.
- Four children visited us today.
- Roy, Aman, Arya and Shyam visited us today

Property of Language

Problem: same expression means different things in different context.

Advantage: Language allows us to communicate about an infinite world using a finite and learnable number of symbols

Give me the scissors.(Kitchen)

Give me the scissors.(Tailoring shop)

Give me the scissors.(OT)

Property of Language

Problem: No language is complete. It continuously evolves as new words, expressions and meanings get generated very frequently.

Advantage: languages can evolve because the experiences that we want to communicate also evolve.

I shall call you

I shall SMS you

I shall instagram.

Property of Language

Problem: There are lots of way to say the same thing.

Advantage: When we know a lot, the facts imply each other. Language in AI is intended to be used by agents who know a lot.

The college closes on 14th December after the VTU exams .

The vacations start on 15th December.

The two major questions in NLP

Two main questions:

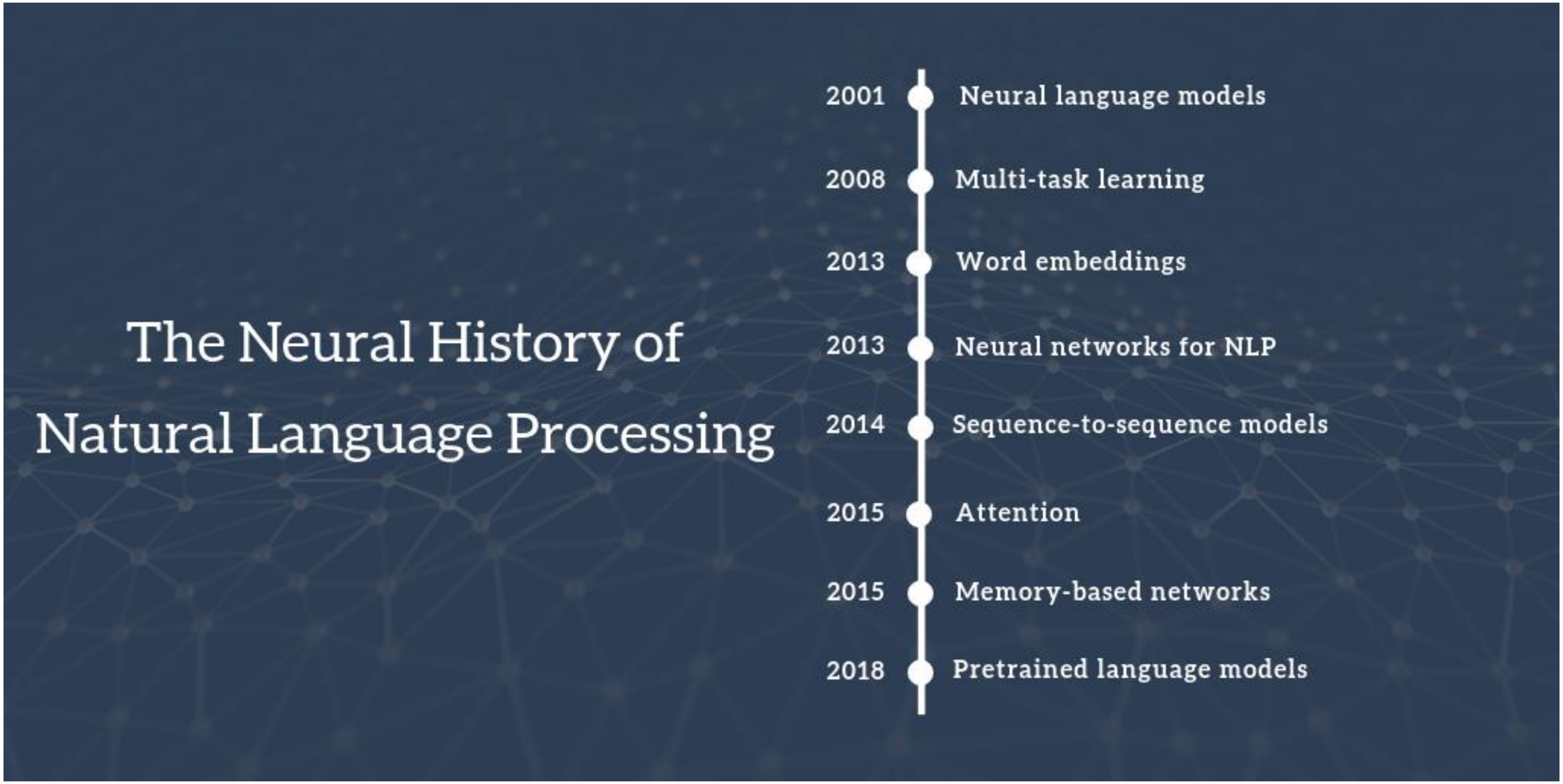
- **What is language understanding?**
- **What does a sentence mean?**

The two major questions in NLP

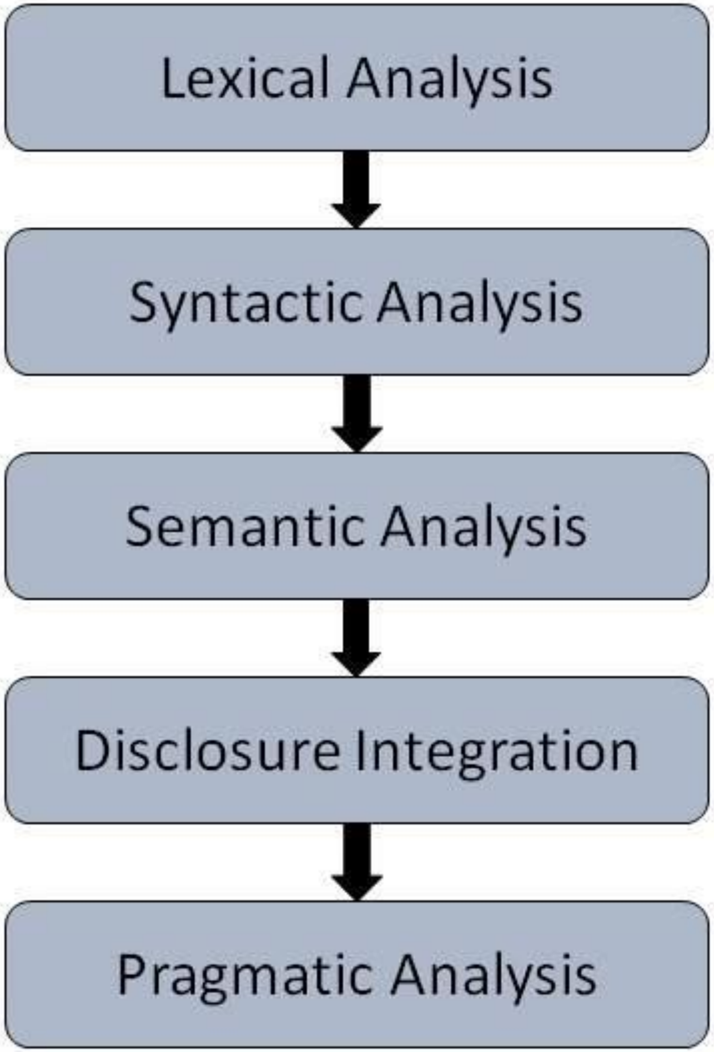
Two main questions:

- **What is language understanding?**
- **What does a sentence mean?**

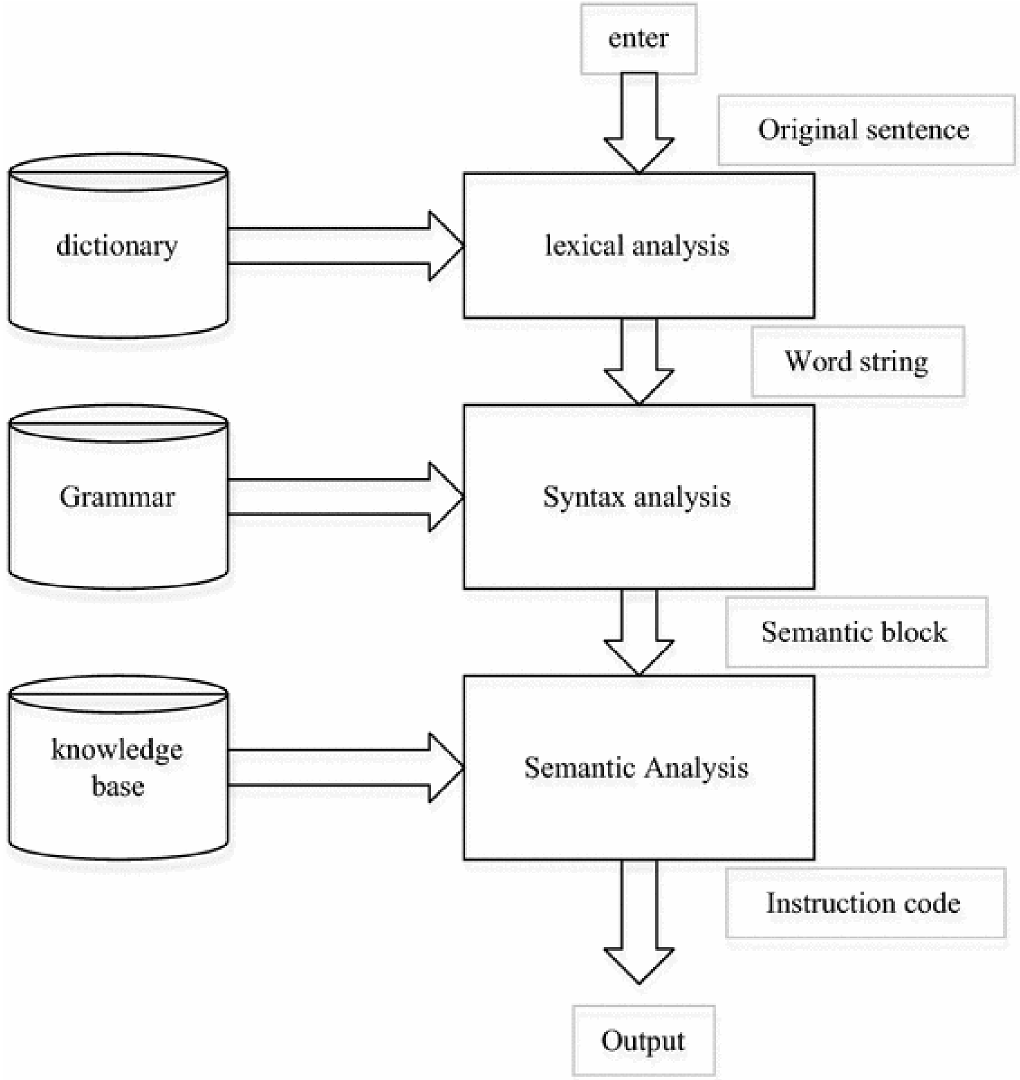
Neural History of NLP



Phases of NLP



Phases of NLP

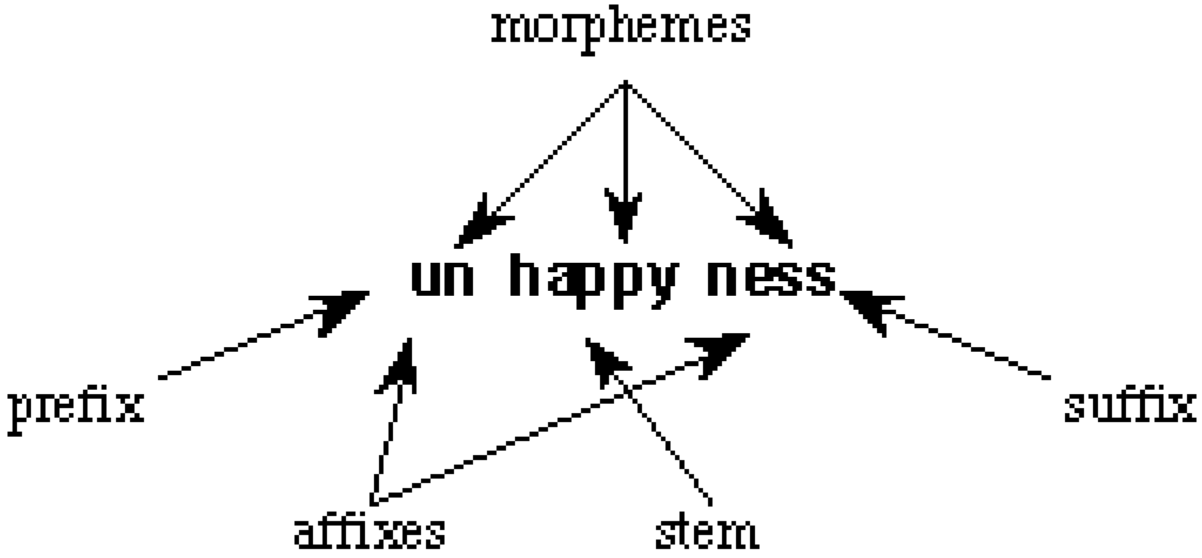


Phase 1: Morphological Analysis

Morphology: Morphology is the study of the structure and formation of words. Its most important unit is the morpheme, which is defined as the "minimal unit of meaning". (Linguistics textbooks usually define it slightly differently as "the minimal unit of grammatical analysis".)

Morphological Analysis: Individual words are analyzed into their components. Non word tokens are separated from their parent words. E.g., punctuations and apostrophes.

Knowledge: typically large amount of knowledge is required to solve complex problems in AI



Phase 1: Morphological Analysis

Morphological Parsing: Examples

Input word	Output morphemes
cats	cat +N +PL
cat	cat + N + SG
cities	city + N + PL
walks	walk + V + 3SG
cook	cook +N +SG or cook +V

Phase 2: Syntactic Analysis

Syntactic Analysis

- Rules of syntax (grammar) specify the possible organization of words in sentences and allows us to determine sentence's structure(s)
  - John saw Mary with a telescope
    - John saw (Mary with a telescope)
    - John (saw Mary with a telescope)
- Parsing: given a sentence and a grammar
  - Checks that the sentence is correct according with the grammar and if so returns a parse tree representing the structure of the sentence

Phase 2: Syntactic Analysis

Syntactic Analysis

- Syntax mapped into semantics
  - Nouns ↔ things, objects, abstractions.
  - Verbs ↔ situations, events, activities.
  - Adjectives ↔ properties of things, ...
  - Adverbs ↔ properties of situations, ...
- A parser recovers the phrase structure of an utterance, given a grammar (rules of syntax)
- Parser's outcome is the structure (groups of words and respective parts of speech)

Phase 2: Syntactic Analysis

Syntactic Analysis

- Phrase structure is represented in a parse tree
- Parsing is the first step towards determining the meaning of an utterance
- Outcome of the syntactic analysis can still be a series of alternate structures with respective probabilities
- Sometimes grammar rules can disambiguate a sentence
  - John set the set of chairs
- Sometimes they can't.
- …the next step is semantic analysis

Phase 2: Syntactic Analysis

- Grammar is very essential and important to describe the syntactic structure of well-formed programs.
- In the literary sense, they denote syntactical rules for conversation in natural languages.
- The theory of formal languages is also applicable in the fields of Computer Science mainly in programming languages and data structure.

A mathematical model of grammar was given by **Noam Chomsky** in 1956, which is effective for writing computer languages.

Mathematically, a grammar G can be formally written as a 4-tuple (N, T, S, P) where −

- **N** or $\mathbf{V_N}$ = set of non-terminal symbols, i.e., variables.
- **T** or $\sum$ = set of terminal symbols.
- **S** = Start symbol where $S \in N$
- **P** denotes the Production rules for Terminals as well as Non-terminals. It has the form $\alpha \rightarrow \beta$, where α and β are strings on $V_N \cup \sum$ and least one symbol of α belongs to V_N

Phase 2: Syntactic Analysis

Symbols in Grammar

- S - Sentence
- NP- Noun Phrase
- PN- Proper Noun
- N-Noun
- VP-Verb Phrase
- Adv-Adverb
- V-Verb
- Adj-Adjective
- Prep-Preposition
- Art-Article

- Pro-Pronoun
- PP-Prepositional Phrase
- * Ungrammatical Sentence
- → Consists of / rewrites as
- ( ) Optional Constituent
- { } Only one of these constituents must be selected

Phase 2: Syntactic Analysis

- A *sentence* in the language defined by a CFG is a series of words that can be derived by systematically applying the rules, beginning with a rule that has s on its left-hand side.

- A *parse* of the sentence is a series of rule applications in which a syntactic category is replaced by the right-hand side of a rule that has that category on its left-hand side, and the final rule application yields the sentence itself.

- Grammar:
  - **Context Free Grammar (CFG)**
  - **Phrase Structure or Constituency Grammar**
  - **Dependency Grammar**

Phase 2: Syntactic Analysis

Dependency Grammar (DG)

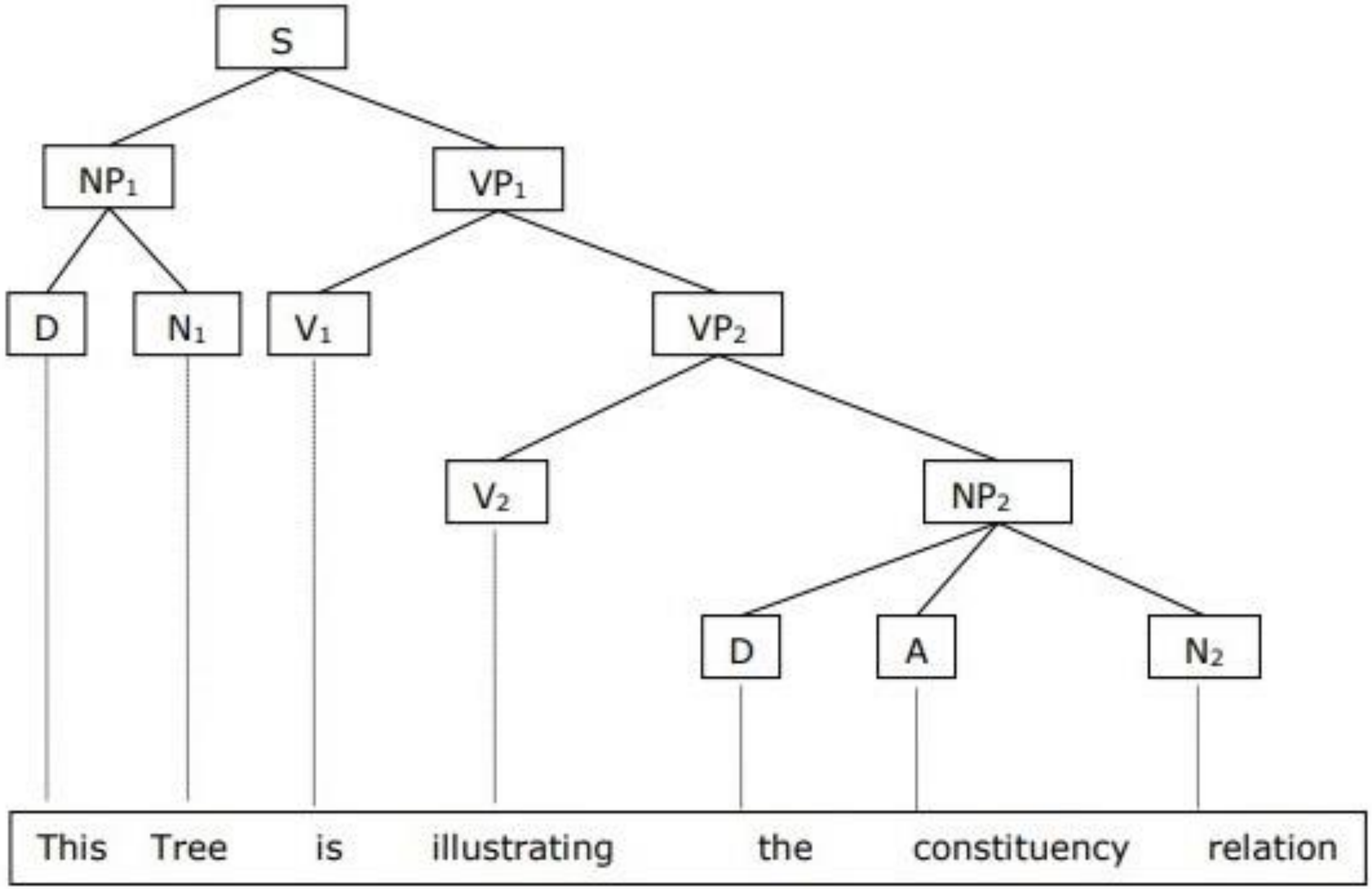
- It is opposite to the constituency grammar and based on dependency relation. It was introduced by Lucien Tesniere.

- Dependency grammar (DG) is opposite to the constituency grammar because it lacks phrasal nodes.

- In DG, the linguistic units, i.e., words are connected to each other by directed links.

- The verb becomes the center of the clause structure. Every other syntactic units are connected to the verb in terms of directed link. These syntactic units are called ***dependencies***.

Phase 2: Syntactic Analysis



Phase 2: Syntactic Analysis

Phrase Structure or Constituency Grammar (CG)

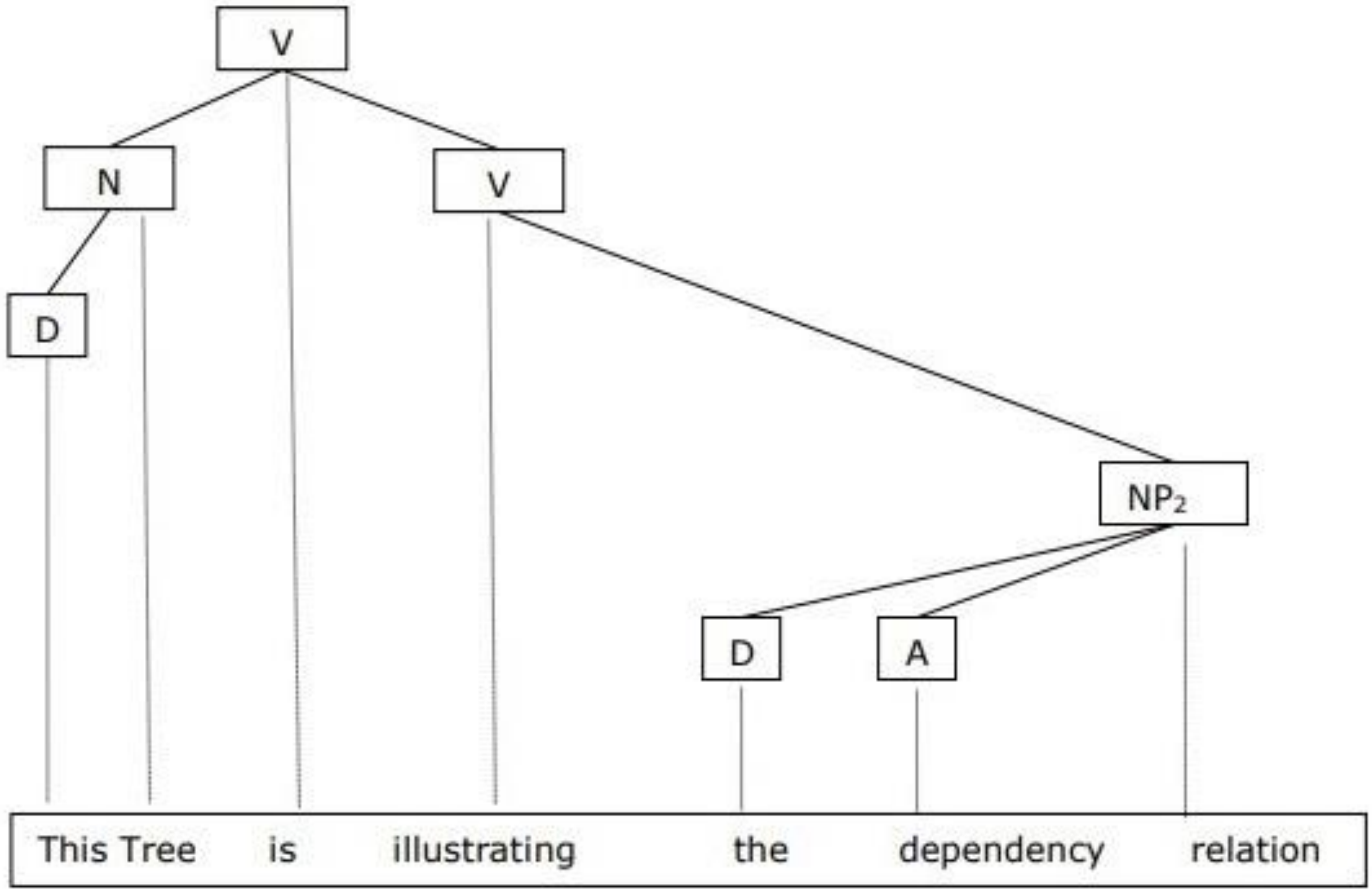
- Phrase structure grammar, introduced by Noam Chomsky, is based on the constituency relation. That is why it is also called constituency grammar. It is opposite to dependency grammar.

- All the related frameworks view the sentence structure in terms of constituency relation.

- The constituency relation is derived from the subject-predicate division of Latin as well as Greek grammar.

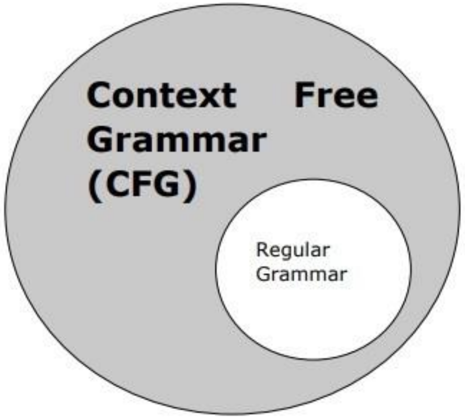
- The basic clause structure is understood in terms of **noun phrase NP** and **verb phrase VP**.

Phase 2: Syntactic Analysis



Phase 2: Syntactic Analysis

Context Free Grammar (CFG)



Set of Non-terminals

It is denoted by V. The non-terminals are syntactic variables that denote the sets of strings, which further help defining the language, generated by the grammar.

Set of Terminals

It is also called tokens and defined by Σ. Strings are formed with the basic symbols of terminals.

Set of Productions

It is denoted by P. The set defines how the terminals and non-terminals can be combined. Every production(P) consists of non-terminals, an arrow, and terminals (the sequence of terminals). Non-terminals are called the left side of the production and terminals are called the right side of the production.

Start Symbol

The production begins from the start symbol. It is denoted by symbol S. Non-terminal symbol is always designated as start symbol.

Phase 2: Syntactic Analysis

- np - noun phrase
- vp - verb phrase
- s - sentence
- det - determiner (article)
- n - noun
- tv - transitive verb (takes an object)
- iv - intransitive verb
- prep - preposition
- pp - prepositional phrase
- adj - adjective

Phase 2: Syntactic Analysis

Figure 1. A grammar and a parse tree for "the giraffe dreams".

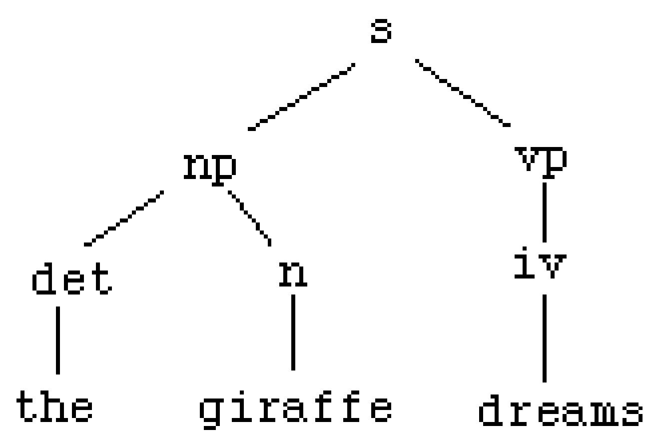
```
s    --> np vp

np   --> det n
vp   --> tv np
     --> iv

det  --> the
     --> a
     --> an

n    --> giraffe
     --> apple

iv   --> dreams
tv   --> eats
     --> dreams
```

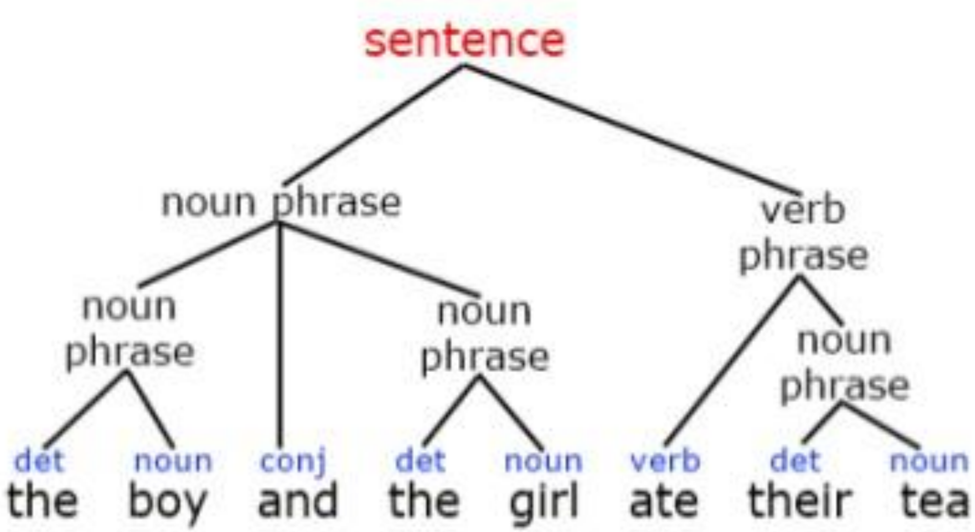


- E.g., a parse of the sentence "the giraffe dreams" is:

s => np vp => det n vp => the n vp => the giraffe vp => the giraffe iv => the giraffe dreams

Phase 2: Syntactic Analysis

Syntactic Analysis - Grammar



Phase 2: Syntactic Analysis

Parsing

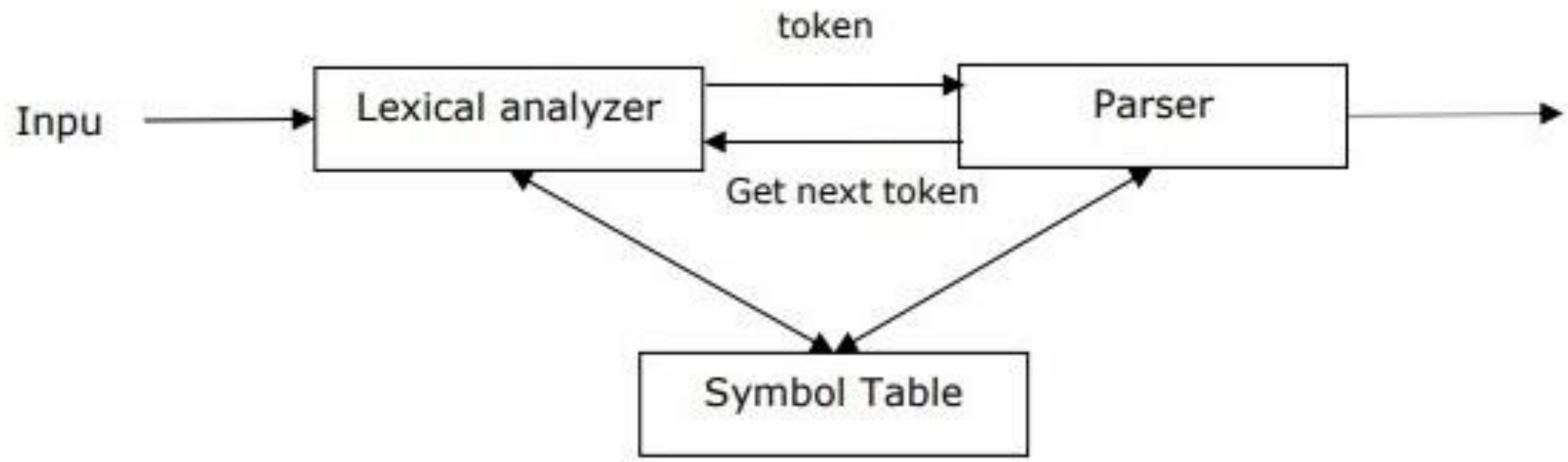
- A method to analyze a sentence to determine its structure as per grammar
- Grammar - formal specification of the structures allowable in the language
- Syntax is important - a skeleton on which various linguistic elements, meaning among them depends
- So recognizing syntactic structure is also important

Phase 2: Syntactic Analysis

Parsing

- Some researchers deny syntax its central role
- There is a verb-centered analysis that builds on Conceptual Dependency - a verb determines almost everything in a sentence built around it - Verbs are fundamental in many theories of language
- Another idea is to treat all connections in language as occurring between pairs of words, and to assume no higher-level groupings
    - Structure and meaning are expressed through variously linked networks of words

Phase 2: Syntactic Analysis



- It is used to implement the task of parsing. It may be defined as the software component designed for taking input data (text) and giving structural representation of the input after checking for correct syntax as per formal grammar.
- It also builds a data structure generally in the form of parse tree or abstract syntax tree or other hierarchical structure.

Phase 2: Syntactic Analysis

The main roles of the parse include −

- To report any syntax error.
- To recover from commonly occurring error so that the processing of the remainder of program can be continued.
- To create parse tree.
- To create symbol table.
- To produce intermediate representations (IR).

Phase 2: Syntactic Analysis

- Types of Parsing
- Derivation divides parsing into the followings two types −

- Top-down Parsing

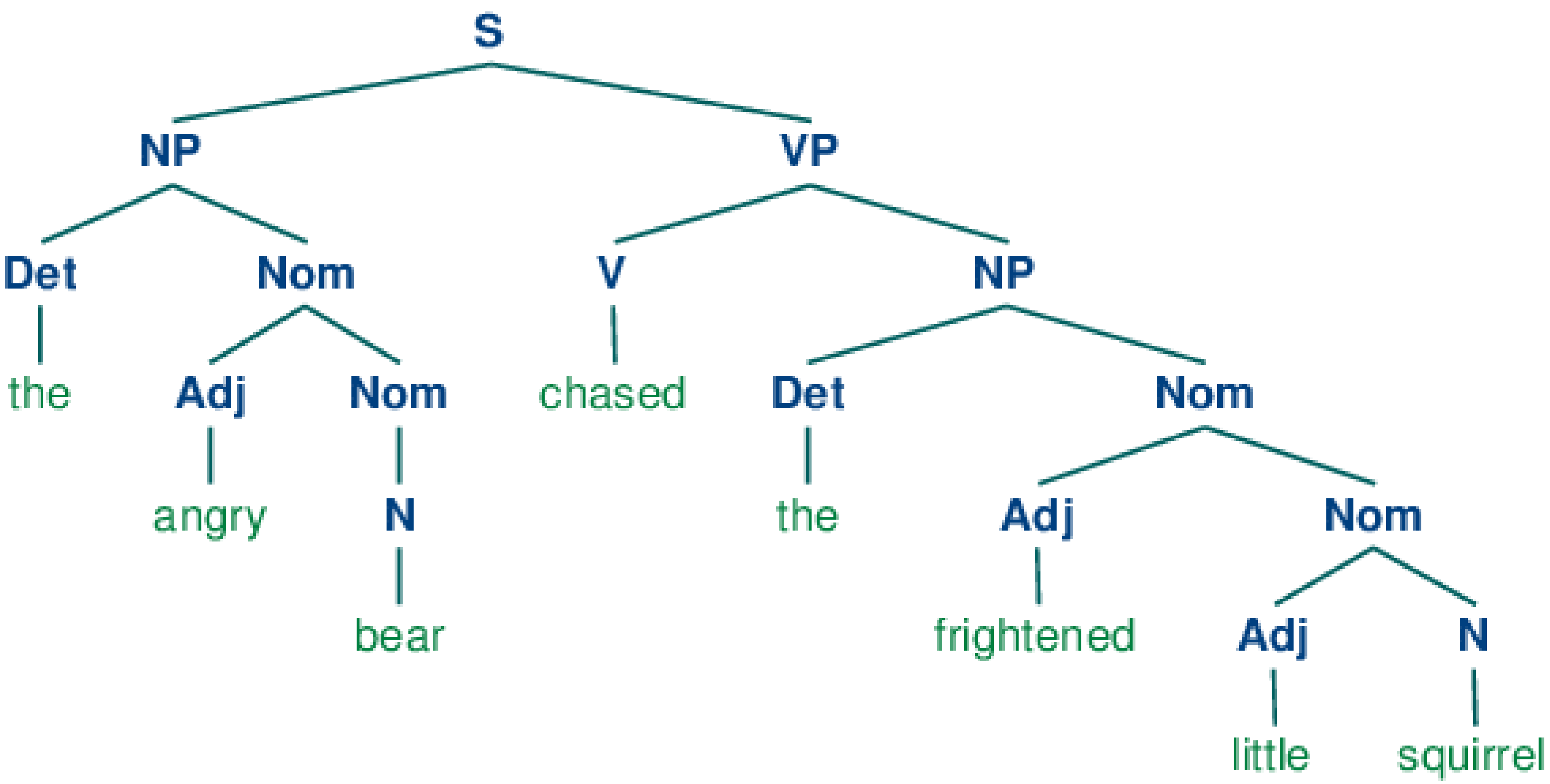
- Bottom-up Parsing

Phase 2: Syntactic Analysis

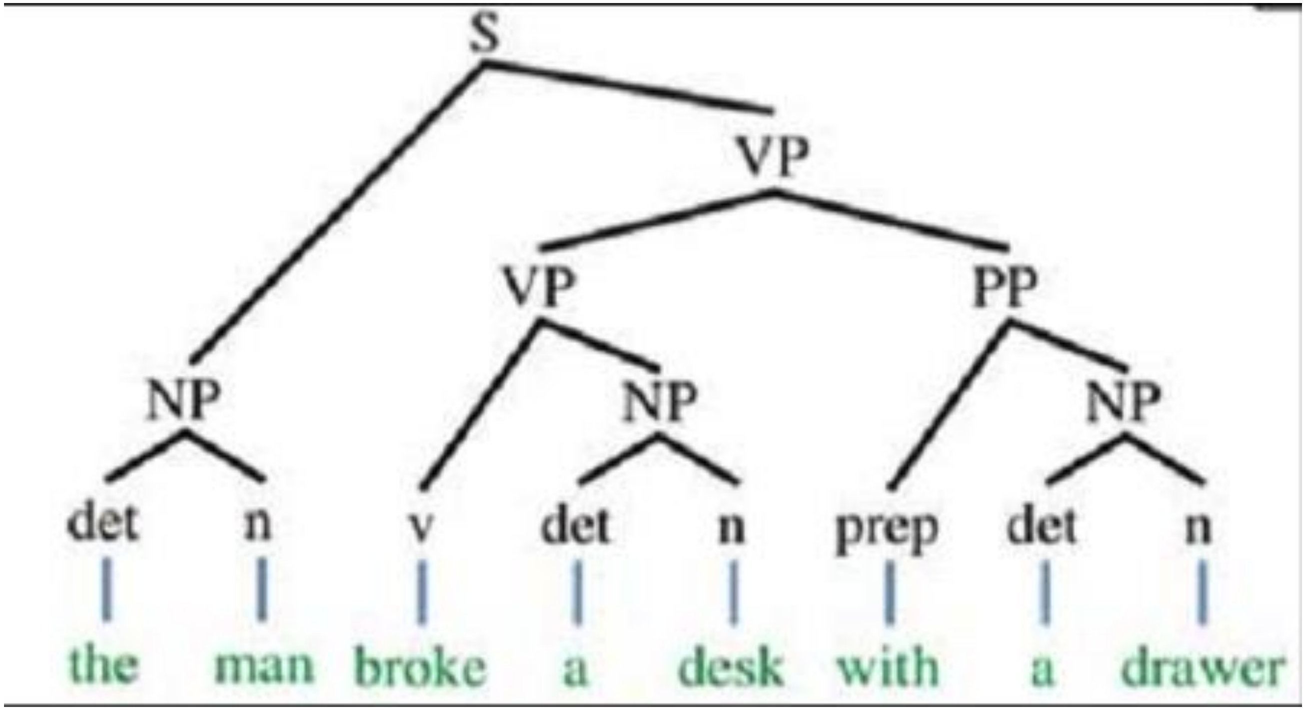
Syntactic Analysis - Challenges

- Number (singular vs. plural) and gender
  - sentence-> noun_phrase(n), verb_phrase(n)
  - proper_noun(s) -> [mary]
  - noun(p) -> [apples]
- Adjective
  - noun_phrase-> determiner,adjectives,noun
  - adjectives-> adjective, adjectives
  - adjective->[ferocious]
- Adverbs, …
- Handling ambiguity
  - Syntactic ambiguity - fruit flies like a banana
- Having to parse syntactically incorrect sentences

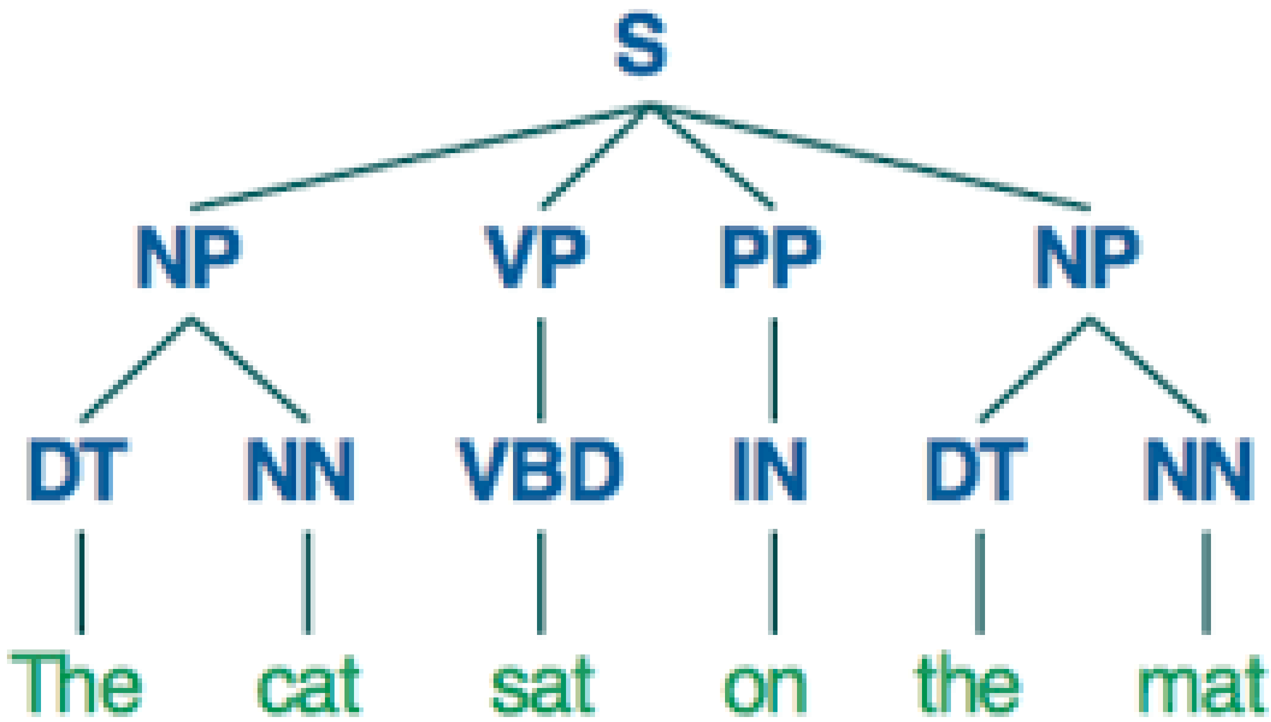
Phase 2: Syntactic Analysis



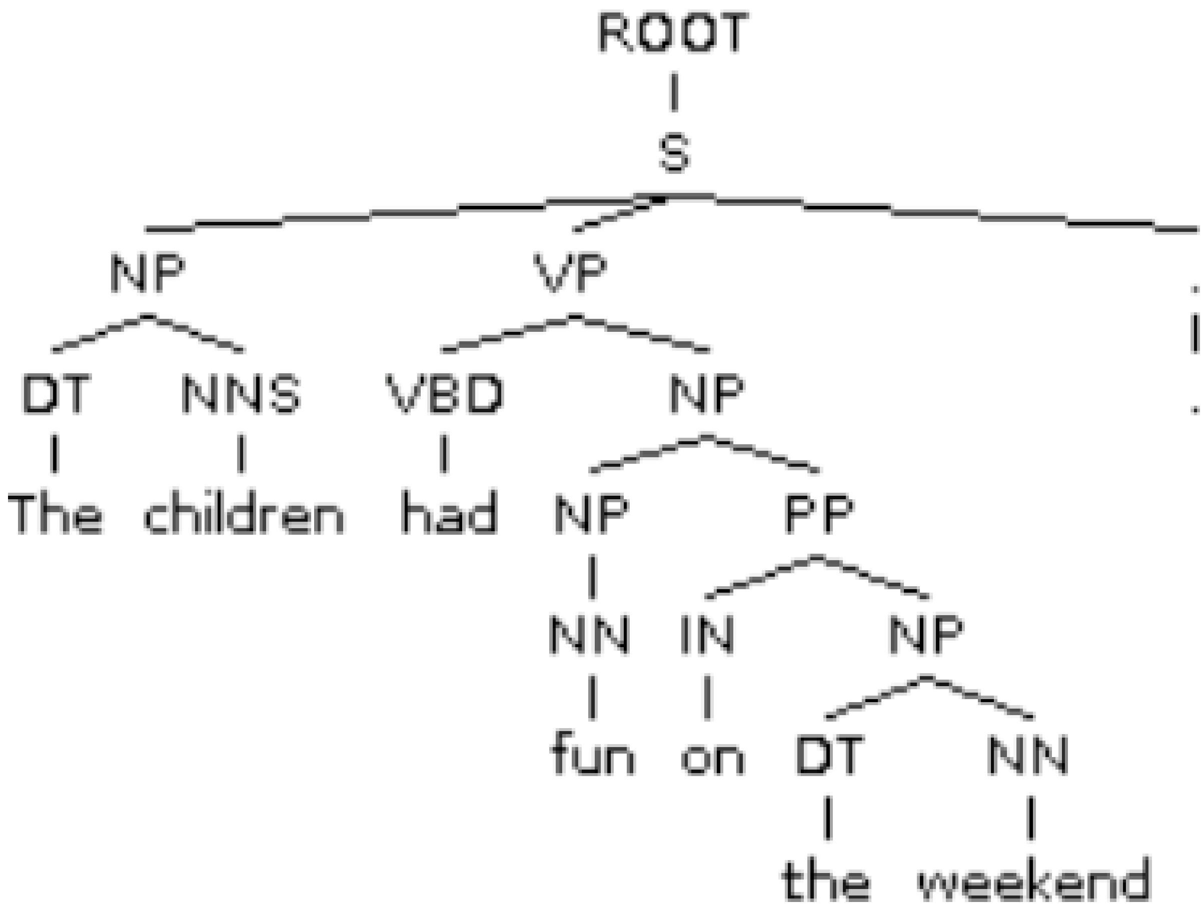
Phase 2: Syntactic Analysis



Phase 2: Syntactic Analysis



Phase 2: Syntactic Analysis



Phase 2: Syntactic Analysis

- Thus parsing converts flat list of words that form sentences into a hierarchical structure that defines the units that are represented by that flat list.

- These hierarchical structural units will correspond to meaning units when semantic analysis is performed.

- We can create a list of **Reference Markers (RM).** They provide a place to accumulate information about the entities as and when we get it.

- If there this syntactic processing is not performed, then the semantic processing should decide its own constituents.

- If parsing is done, it constraints the number of constituents that semantics can consider.

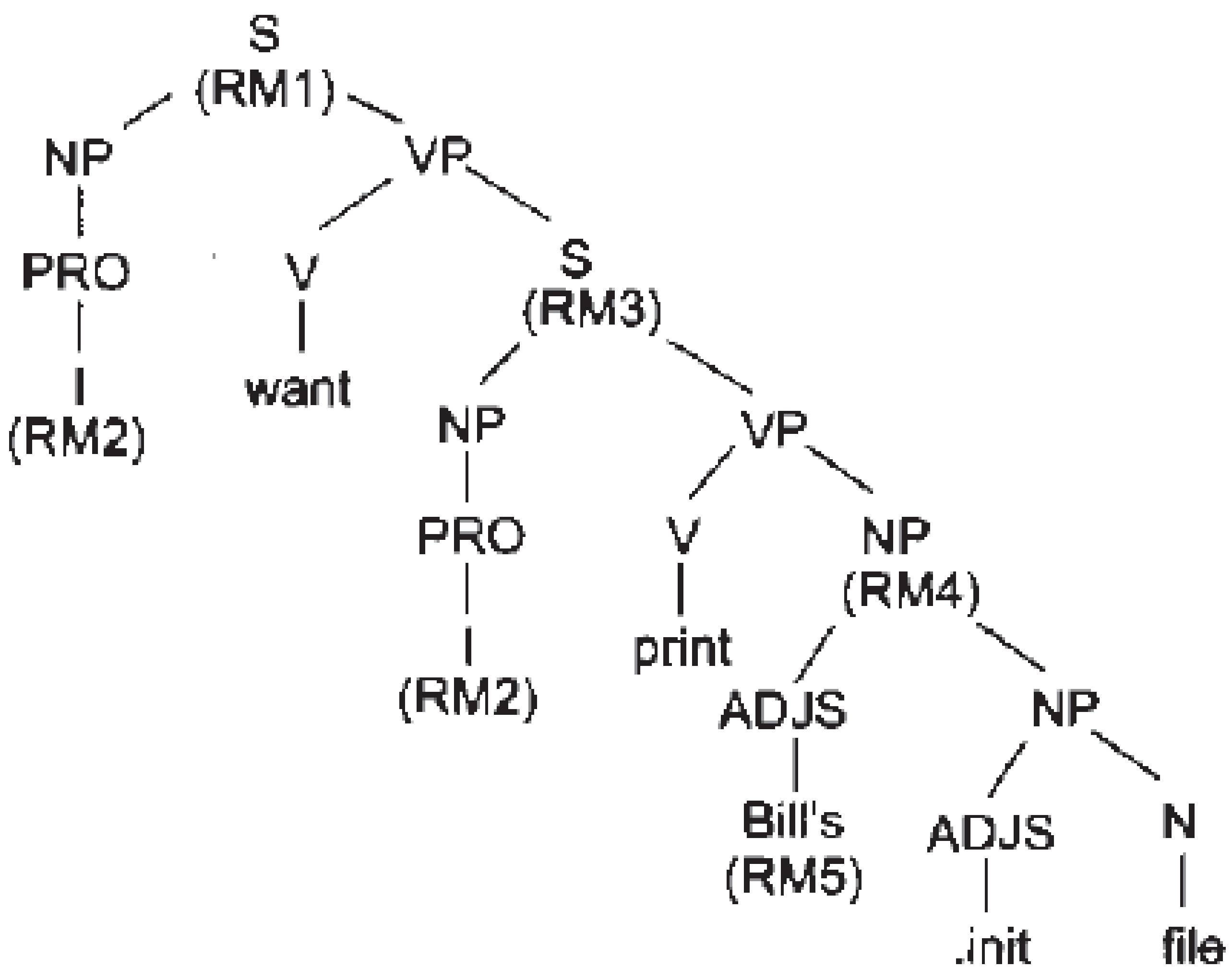
- Syntactic processing is computationally less expensive than is direct semantic processing (which may require substantial inference).

- Reduces overall system complexity.

Phase 2: Syntactic Analysis

- Example of syntactically correct but semantically wrong sentences.
  - ➢ The lion eats the rabbit.
  - ➢ The rabbit eats the lion.

Phase 2: Syntactic Analysis

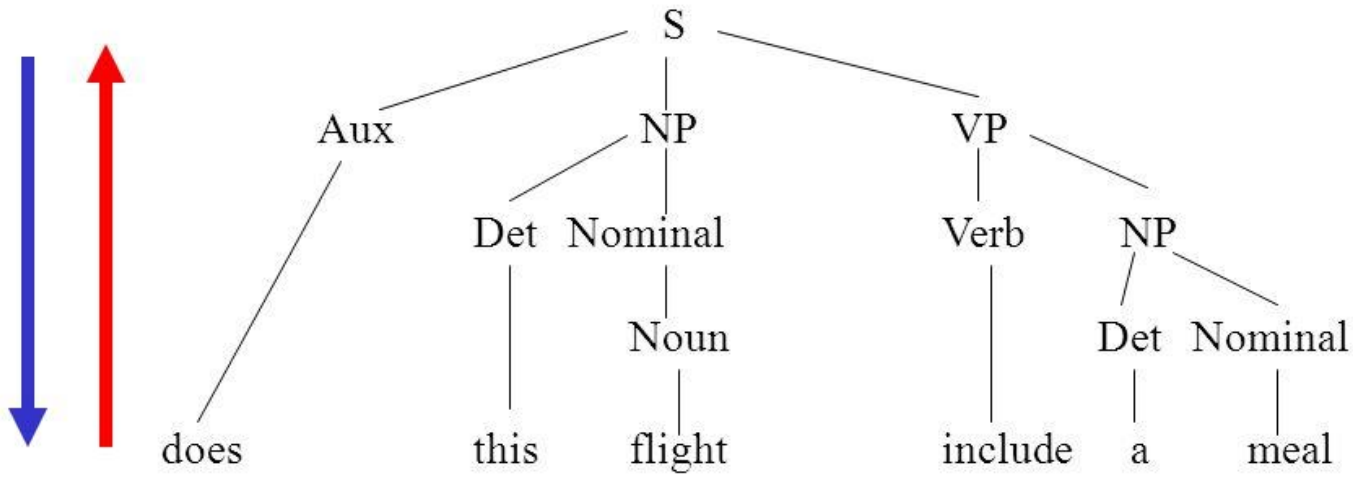


Phase 2: Syntactic Analysis

Bottom-up and Top-down Parsing

Bottom-up – from word-nodes to sentence-symbol

Top-down Parsing – from sentence-symbol to words



Phase 2: Syntactic Analysis

Concept of Derivation

In order to get the input string, we need a sequence of production rules. Derivation is a set of production rules. During parsing, we need to decide the non-terminal, which is to be replaced along with deciding the production rule with the help of which the non-terminal will be replaced.

Types of Derivation

Left-most Derivation

In the left-most derivation, the sentential form of an input is scanned and replaced from the left to the right. The sentential form in this case is called the left-sentential form.

Right-most Derivation

In the left-most derivation, the sentential form of an input is scanned and replaced from right to left. The sentential form in this case is called the right-sentential form.

Phase 2: Syntactic Analysis

Sentence: a and b or c

Derivation

N
⇒ N ‘or’ N
⇒ N ‘or c’
⇒ N ‘and’ N ‘or c’
⇒ N ‘and b or c’
⇒ ‘a and b or c’

Sentential Form: Rightmost derivation

- In this derivation each line is called a **sentential form**.
- Furthermore, each line of the derivation applies a rule from the CFG in order to show that the input can, in fact, be derived from the start symbol N.
- In the above derivation, we restricted ourselves to only expand on the rightmost non-terminal in each sentential form. This method is called the **rightmost derivation** of the input using a CFG.

Phase 2: Syntactic Analysis

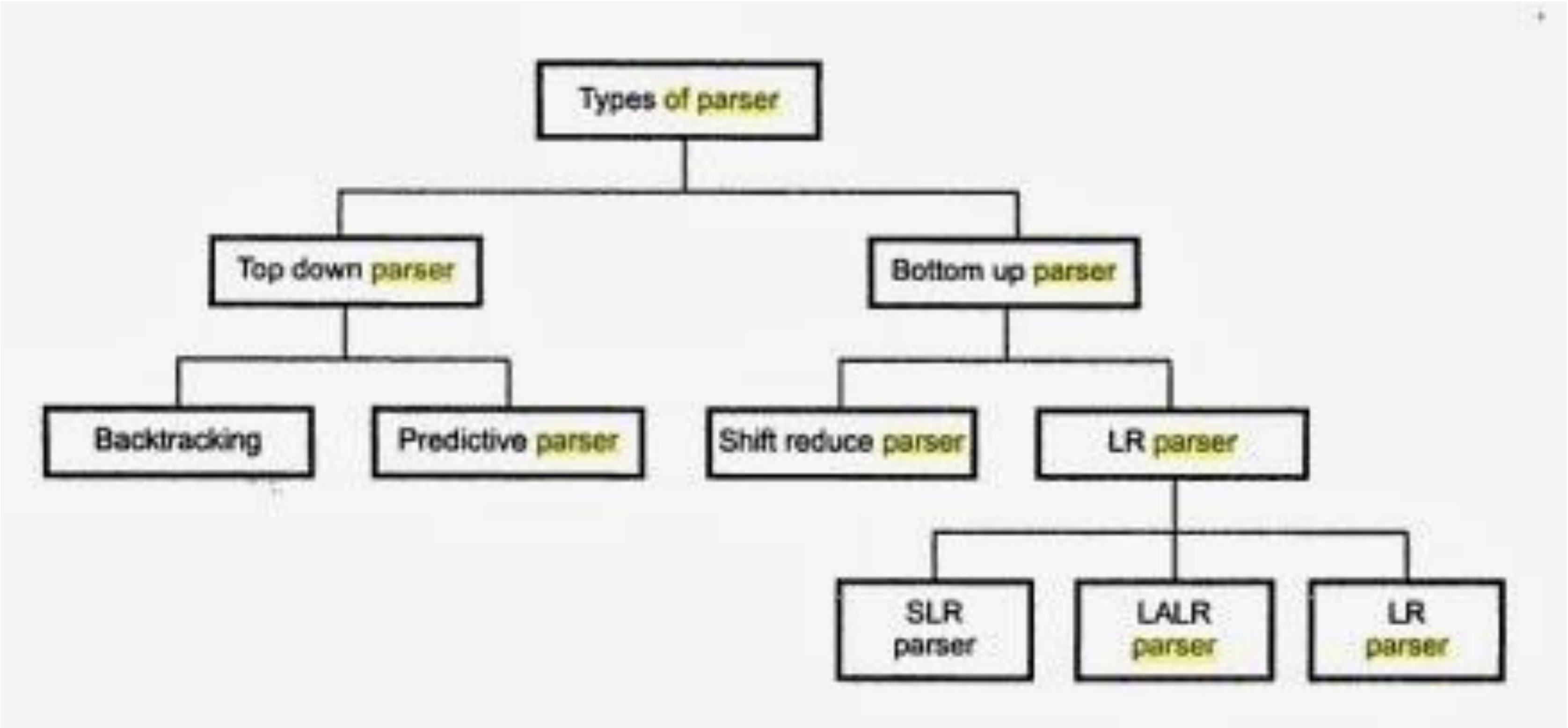
Rightmost Derivation Reverse Order

‘a and b or c’
⇒ N ‘and b or c’ # use rule N -> a
⇒ N ‘and’ N ’or c’ # use rule N -> b
⇒ N ‘or c’ # use rule N -> N and N
⇒ N ‘or’ N # use rule N -> c
⇒ N # use rule N -> N or N

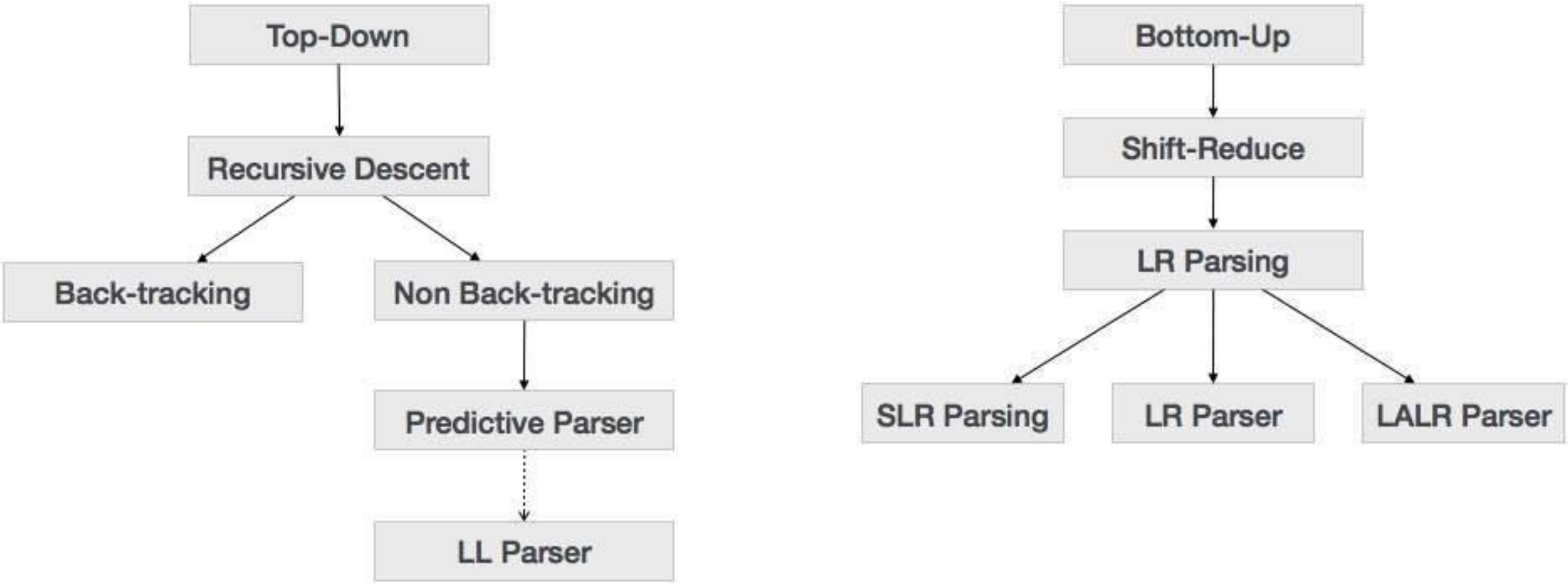
Phase 2: Syntactic Analysis

Parse tree	Stack	Input	Action
		a and b or c	Init
a	a	and b or c	shift a
(N a)	N	and b or c	reduce `N -> a`
(N a) and	N and	b or c	shift and
(N a) and b	N and b	or c	shift b
(N a) and (N b)	N and N	or c	reduce `N -> b`
(N (N a) and (N b))	N	or c	reduce `N -> a`
(N (N a) and (N b)) or	N or	c	shift or
(N (N a) and (N b)) or c	N or c		shift c
(N (N a) and (N b)) or (N c)	N or N		reduce `N -> c`
(N (N (N a) and (N b)) or (N c))	N		reduce `N -> N or N`
(N (N (N a) and (N b)) or (N c))	N		Accept!

Phase 2: Syntactic Analysis



Phase 2: Syntactic Analysis



Phase 2: Syntactic Analysis

Item	Top down	Bottom up
Definition	A parsing strategy that first looks at the highest level of the parse tree and works down the parse tree by using the rules of a formal grammar.	A parsing strategy that first looks at the lowest level of the parse tree and works up the parse tree by using the rules of a formal grammar.
Functionality	The parsing occurs from the starting symbol to the input string.	The parsing occurs from the starting symbol to the input string
Main decision	To select what production rule to use in order to construct the string	To select when to use a production rule to reduce the string to get the starting symbol
Method of construction	Uses leftmost derivation	Uses rightmost derivation

Phase 2: Syntactic Analysis

Transition Networks

My dog ate a frog.

```
   S <- NP VP
  VP <- VERB NP
  NP <- ART NOUN
  NP <- NAME
  NP <- POSS NOUN
NAME <- Jack
NOUN <- frog | dog
 ART <- a
VERB <- ate
POSS <- my
```

Parse Tree

```
                S
               / \
              /   \
             /     \
          NP         VP
         / \        / \
        /   \      /   \
     POSS  NOUN VERB    NP
      |     |    |     / \
      |     |    |    /   \
      |     |    |  ART  NOUN
      |     |    |   |     |
     my    dog  ate  a   frog
```

Derivation

```
(S (NP (POSS my)
       (NOUN dog))
   (VP (VERB ate)
       (NP (ART a)
           (NOUN frog))))
```

Phase 2: Syntactic Analysis

Transition Networks

My dog ate a frog.

```
   S  <- NP VP
  VP  <- VERB NP
  NP  <- ART NOUN
  NP  <- NAME
  NP  <- POSS NOUN
NAME  <- Jack
NOUN  <- frog | dog
 ART  <- a
VERB  <- ate
POSS  <- my
```

```
                                              pop
                                             /
         NP            VP                  /
S0  --------->  S1  --------->  S2
                                     /
         VERB              NP           /
VP0 --------->  VP1 --------->  VP2

                                              pop
                                             /
         ART              NOUN              /
NP0 --------->  NP1 --------->  NP2

                     pop
                    /
         NAME      /
NP3 --------->  NP4

                                              pop
                                             /
         POSS             NOUN              /
NP5 --------->  NP6 --------->  NP7
```

Phase 2: Syntactic Analysis

Transition Networks

```
                                   pop
                                  /
      NP           VP            /
 S0 ---------> S1 ---------> S2
                                 /
      VERB          NP          /
VP0 ---------> VP1 ---------> VP2

                                   pop
                                  /
      ART          NOUN           /
NP0 ---------> NP1 ---------> NP2

                    pop
                   /
      NAME        /
NP3 ---------> NP4

                                   pop
                                  /
      POSS         NOUN           /
NP5 ---------> NP6 ---------> NP7
```

```
                                   pop
                                  /
      NP           VP            /
 S0 ---------> S1 ---------> S2

                                   pop
                                  /
      VERB          NP           /
VP0 ---------> VP1 ---------> VP2
       ART          NOUN
    ---------> NP1 ---------
   /            ^           \
  /            /             \      pop
 /            /               \    /
/    POSS    /               _\|/
NP0 -----------                NP2
\                              ^
 \                            /
  \                          /
   \         NAME           /
    -----------------------
```

Phase 2: Syntactic Analysis

Transition Networks

My dog ate a frog.

```
        ART       NOUN                         ART       NOUN
      ------> S1 -------                     ------> S4 -------
     /       ^          \                    /       /         \
    /       /            \                  /      /            \    pop
   /       /              \                /      /              \    /
  /  POSS /               _\|  VERB       / POSS  /              _\| /
S0 -------                  S2 ------->  S3 -------                 S5
  \                        ^              \                        ^
   \                      /                \                      /
    \                    /                  \                    /
     \       NAME       /                    \       NAME       /
      ------------------                     -------------------
```

Phase 2: Syntactic Analysis

Recursive Transition Networks (RTN)

- Natural languages allow us to express an infinite range of ideas using a finite set of rules and symbols. We need the notion of recursion to do this.
- Not only are there an infinite number of sentences, but a single sentence can be infinitely long. Recursion enables us to represent this capability of language as well.
- While we can generate infinitely long sentences in English, such sentences will not necessarily be understandable given real constraints on human processing and memory, human lifespan, the age of the universe, and stuff like that

The boy drowned.

The boy with the raft drowned.

The boy with the raft near the island drowned.

The boy with the raft near the island in the ocean drowned.

The boy ....

Phase 2: Syntactic Analysis

Backtracking

- RTNs describe a process
- The RTN parser "knows" at each step of the parsing process exactly which choices are available, or what knowledge will be relevant. On the other hand, a CFG-based parser will have to search through the list of rules each step of the way to find the applicable ones.
- If there's more than one choice, how does the parser know which one is the right one? It doesn't, so we have to introduce the notion of backtracking.

The little orange ducks swallow flies.

```
                                                        pop
                                                        /
     ART        1  NOUN        VERB         NOUN       /
S ---------> S1 ---------> S2 ---------> S3 ---------> S4
             /\
            /  \ 2
            \  /
             \/
            ADJ
```

Phase 2: Syntactic Analysis

Backtracking

- You need to backtrack to the last decision point. In order to backtrack, your parser must record at every decision point:
- What word in the sentence it was looking at when it made the decision
- What node (or state) in the network it was at
- What other arcs (or paths) could have been chosen but weren't
- When the parser bumps into a dead end like that described above, it must reset the parsing process so that it's looking at the previously stored word, it's restarting from the previously stored node, and it's traversing the first of the previously stored arcs.
- (If there are more than one arc remaining, just push the information again, without the first arc, and proceed as usual.)

Phase 2: Syntactic Analysis

Augmented Transition Network (ATN)

- Recursive transition networks are interesting from a theoretical standpoint but are not of much use by themselves.
- From a computational perspective, a black box that accepts English input and just says "yes" or "no" doesn't buy us much.
- What we need is a black box that records the structure of the input as well as providing an evaluation of syntactic correctness. We do this by adding procedures to the arcs of the RTN.
- These procedures are then performed when the corresponding arcs are traversed. The resulting network is called an "augmented transition network" (ATN). (And sometimes, adding procedures to the arcs of a transition network is called "procedural attachment.")
- While we add procedures or augmentations in ATN, we store information in registers when arcs are traversed.

Phase 2: Syntactic Analysis

Augmented Transition Network (ATN)

- To record the structure of the input, we add an action to each arc which stores the word that was processed while traversing that arc in an appropriate register.

```
                                                         "the vicious dog"
                                        pop
                                         /
       ART          1   NOUN            /
NP ---------> NP1 ---------> NP2
     :         /\            :            ART  = the
     :        /  \ 2         :           ADJS  = vicious
     :        \  /           :           NOUN  = dog
     :         \/ ADJ        :
     :         :             :
     :         :             :
     :         :             :
  ART <- *     :      NOUN <- *          (NP (ART the)
               :                             (ADJS vicious)
      ADJS <- ADJS + *                       (NOUN dog))
```

Phase 2: Syntactic Analysis

Augmented Transition Network (ATN)

"the vicious dog ate the wimpy frog"

```
(lexicon
  (the      (CAT ART))
  (vicious (CAT ADJ))
  (dog      (CAT NOUN))
  (ate      (CAT VERB))
  (wimpy    (CAT ADJ))
  (frog     (CAT NOUN)))
```

```
(S (SUBJ (NP (ART the)
             (ADJS vicious)
             (NOUN dog)))
   (VERB ate)
   (OBJ  (NP (ART the)
             (ADJS wimpy)
             (NOUN frog))))
```

Phase 2: Syntactic Analysis

Augmented Transition Network (ATN) for feature tests:

Subject verb agreement: To perform agreement checks in the ATN framework, we attach "feature tests" to the arcs.

```
                                           pop
                                           /
     NP          VERB          NP         /
S ---------> S1 ---------> S2 ---------> S3
      :                :             :
      :                :            OBJ <- *
      :                :
   SUBJ <- *          VERB <- *

 NUM    <- NUM*         NUM <- NUM      intersect NUM*
    subj                          subj
```

Phase 2: Syntactic Analysis

Augmented Transition Network (ATN) for feature tests:

Subject verb agreement: To perform agreement checks in the ATN framework, we attach "feature tests" to the arcs.

```
                                      pop
                                       /
       ART          1   NOUN          /
 NP ----------> NP1 ----------> NP2
      :         /\            :
      :        /  \ 2         :
      :        \  /           :
      :         \/ ADJ        :
      :          :            :
  ART <- *       :      NOUN <- *
                 :
  NUM <- NUM*    :       NUM <- NUM intersect NUM*
                 :
                 :
       ADJS <- ADJS + *
```

```
(lexicon
  (the      (CAT ART)
            (NUM 3s 3p))
  (vicious  (CAT ADJ))
  (dog      (CAT NOUN)
            (NUM 3s))

  (ate      (CAT VERB)
            (NUM 1s 2s 3s 1p 2p 3p))
  (eat      (CAT VERB)
            (NUM 1s 2s 1p 2p 3p))
  (eats     (CAT VERB)
            (NUM 3s))
  (wimpy    (CAT ADJ))
  (frog     (CAT NOUN)
            (NUM 3s)))
```

Phase 2: Syntactic Analysis

Augmented Transition Network (ATN) for feature tests:

Subject verb agreement:

- Now, if we try to parse a sentence like "the dog eat the frog" our ATN parser will catch the agreement error when it performs the intersection on the number of "dog" and the number of "eat“.

- The result is the empty set.

- If the intersection between the set of all possible values for the number of the subject and the set of all possible values for the number of the verb is the empty set, then there is no agreement between the subject and the verb. The sentence must be syntactically incorrect.

Phase 2: Syntactic Analysis

Augmented Transition Network (ATN) for feature tests:

Sub categorization of the verb:

- Feature tests can also be used to check for agreement between auxiliary verbs and the main verb in a verb group.
- They can also be used to check for the correct verb complement structure.
- The feature tests can be used to see if the right number of noun phrases follow the verb.
- Remember that intransitive verbs expect no noun phrases, transitive verbs expect one noun phrase, and bitransitive verbs expect two noun phrases.
- Also note that any given verb might fall into more than one of these categories.
- Information about the complement structure is called the "subcategorization" of the verb.

Phase 3: Semantic Analysis

Semantic Analysis

- Generates meaning/representation of the sentence from its syntactic structures
- Represents the sentence in meaningful parts
- Uses possible syntactic structures and meaning
- Builds a parse tree with associated semantics
- Semantics typically represented with logic

- It must map individual words into appropriate objects in the knowledge base or database

- It must create the correct structures to respond to the way the meanings or the individual words combine with each other.

Phase 3: Semantic Analysis

Semantic Analysis

- Compositional semantics: meaning of the sentence from the meaning of its parts
  - Sentence - A tall man likes Mary
  - Representation - Ξx man(x) & tall(x) & likes(x,mary)
- Grammar + Semantics
  - Sentence (Smeaning)-> noun_phrase(NPmeaning), verb_phrase(VPmeaning), combine(NPmeaning,VPmeaning,Smeaning)
- Complications - Handling ambiguity
  - Semantic ambiguity - I saw the prudential building flying into Paris

Phase 3: Semantic Analysis

Lexical Processing

- Look up individual words in dictionary (lexicons) and extract their meanings

- Many words have multiple meanings, may not be possible to choose the correct one just by looking at the word itself.

- Lexical ambiguity:
  - Mean:
    - ❖ To signify
    - ❖ Unpleasant or cheap
    - ❖ Statistical average

- Lexical disambiguation: associating with each word in the lexicon, the information about the contexts in which each of the word’s senses may appear. Each of the words in a sentence can serve as part of the context in which h meanings o the other words must be determined.

Phase 3: Semantic Analysis

Lexical Processing

- Semantic markers:
  - ❖ Physical – object -→ I saw the diamond shimmering
  - ❖ Animate – object -→ My house loves the Sun.
  - ❖ Abstract – object -→ You are a gem.

Sentence Level Processing

- Semantic grammar: combines the following knowledge into a single set of rule in the form of grammar. The result of parsing with this grammar will be semantic parsing
  - ❖ Syntactic
  - ❖ Semantic
  - ❖ Pragmatic

- Case grammar : parsing output with semantic information + further interpretation

- Conceptual parsing: syntactic + semantic knowledge combined into single information system, guided by semantic knowledge.
- Approximately compositional semantic interpretation: semantic processing applied to results of syntactic parsing.

Phase 3: Semantic Analysis

Semantic Grammar

- Choice of non-terminals and production rules is governed by semantic as well as syntactic functions.
- Usually a semantic action associate with each grammar rule
- The result of parsing and applying all the associated semantic actions is the meaning of the sentence.

- The grammar rules themselves are designed around key semantic concepts.

- The proper constituents of semantic grammar are noun phrase (NP), verb phrase (VP), Noun (N), Verb (V), Preposition phrase (PP), Adverb (ADV) and so on.

- One of the successful applications of semantic grammar is the LIFER system (A database query system).

- Semantic grammars are suitable for use in systems with restricted grammars since its power of computation is limited.
- Semantic grammars are grammars whose non-terminals correspond to semantic concepts (e.g., [greeting] or [suggest.time]) rather than to syntactic constituents (such as Verb or WounPhrase).

Phase 3: Semantic Analysis

Advantages of Semantic Grammar

- They have the advantage that the semantics of a sentence can be directly read off its parse tree.

- When the parse is complete, the result can be used immediately without the additional stage of processing that would be required if a semantic interpretation had not already been performed during the parse.

- Many ambiguities that would arise during a strictly syntactic parse can be avoided since some of the interpretations do not make sense semantically and thus cannot be generated by a semantic grammar.

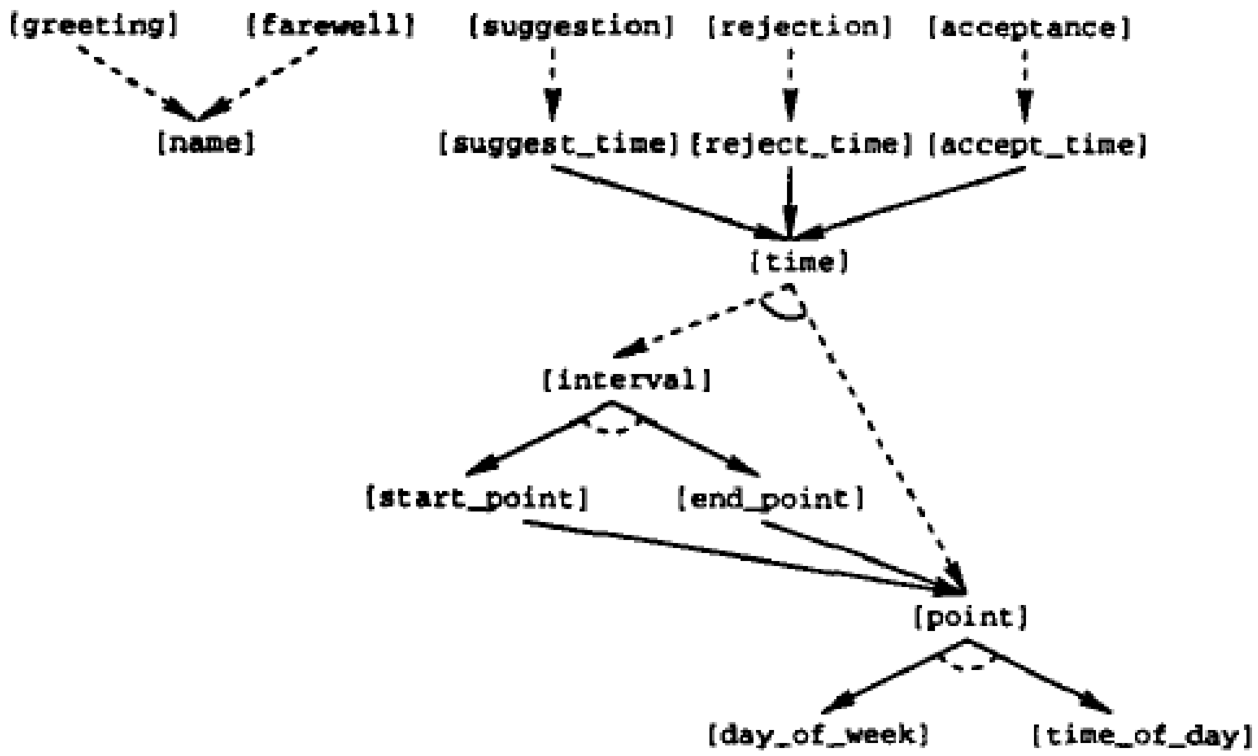
- Syntactic issues that do not affect the semantics can be ignored.

Disadvantages of Semantic Grammar

- A new grammar must be developed for each domain.
- The number of rules required can become very large since many syntactic generalizations are missed.
- Parsing process may be expensive due to large number of rules.

Phase 3: Semantic Analysis

Semantic Grammar



(1) [suggestion] ⟵ [suggest_time]
(2) [suggest_time] ⟵ *how about* [time]
(3) [time] ⟵ [point]
(4) [point] ⟵ **on* [day_of_week] *[time_of_day]
(5) [day_of_week] ⟵ *Tuesday*
(6) [time_of_day] ⟵ *afternoon*

Phase 3: Semantic Analysis

Case Grammar

- Case Grammars attempt to describe any given sentence in terms of a fixed frame of slots (called cases) which explicitly capture information about any activities described in the sentence, the instigators of those activities, positions, times, etc.
- Though there is no universally agreed set of cases or their names a common subset is outlined below.

Case	Meaning
Action	the action which takes place (usually related to the main verb of a sentence)
Actor	the instigator of the action (often an animate entity which/who *does* the action)
Object	the entity which is acted upon or is changed by the action
Source	the starting position for an entity (ie: its position before the action takes place)
Destination	the resulting position for an entity (after the action has completed)
Location	the location for the action
Instrument	an object/entity used in order to cause the event (eg: *key* in "*He unlocked the door with a key*"
Time	the time/date of the action

Phase 3: Semantic Analysis

Case Grammar

"Gary repaired the car in the garage on Sunday"

Action	repairs
Actor	Gary
Object	car
Location	garage
Time	Sunday

Phase 3: Semantic Analysis

Case Grammar

- With Case Grammars rules describe syntactic constraints but also describe manipulations geared towards producing case frames (sets of case slots).
- These may be flat (as in the example above) or nested hierarchical structures which can form the basis of semantic representations.
- To implement a Case Grammar the notation used to express rules must allow them to produce case frames (ie: structures without a pre-determined form)
- Grammar notations whose only output reflects the syntactic structure of their input are not suitable since they cannot be used to construct case frames.
- These rules can be applied in reverse by a parser to determine the underlying case structure from the superficial syntax.
- Parsing using a case grammar is usually expectation-driven. Once the verb of the sentence has been located, it can be sed to predict the noun phrases that will occur and to determine the relationship of those phrases to the rest of the sentence.

Phase 3: Semantic Analysis

Case Grammar

- ATNs provide a good structure for case grammar parsing.
- In traditional parsing algorithms, the output structure always mirrors the structure of the grammar rules that created it. However, in ATNs, the output structures can have arbitrary form.
- ATNs parsers can be used to translate English sentences into a semantic net representing the case structures of the sentences. These semantic nets can then be used to answer questions about the sentences.
- The result of parsing in a case representation I usually not a complete semantic description of a sentence.
- The constituents that fill the slots may still be English words rather than true semantic descriptions stated in the target representation.

Phase 3: Semantic Analysis

Conceptual Parsing

- Finds both structure and meaning of a sentence in one step.
- Driven by a dictionary hat describes the meanings of words as Conceptual Dependency (CD) structures.
- Similar to case grammar.
- Parsing process is heavily driven by a set of expectations that are set up on the basis of the sentence's main verb.
- However, here, the representation of a verb in CD is at a lower level than that of a verb in a case grammar where the representation is often identical to the English word that is used as verb.
- CD usually provides a greater degree of predictive power.
- The first step in mapping a sentence into its CD representation involves a syntactic processor that extracts the main noun and verb.

Phase 3: Semantic Analysis

Conceptual Parsing

- The grammar of the system is bipartite.
- The first part is a universal grammar exemplified by the conceptual rules employed by the system.
- The second part is language-specific and is made up of r e realization rules intended to map pieces of the conceptual network into linguistic items.
- The realization rules may be used for both parsing and generating. However, it is not necessary to use all the realization rules in order to parse.
- The system is capable of making sense of a piece of language containing only a few words that it knows since its conceptual framework is capable of making predictions.
- Thus, it can understand while using only a few realization rules, whereas it would need a great many more to map the same structure back into language. This phenomenon is similar to that observed in a man attempting to learn a foreign language.

Phase 3: Semantic Analysis

Conceptual Parsing

<u>Governing categories</u>

PP	An actor or object; corresponds roughly to a syntactic noun or pronoun.
ACT	In English, corresponds syntactically to verbs, verbal nouns (e.g., gerunds) and certain abstract nouns.
LOC	A noun denoting the location of a conceptualization.
T	Denotes the time of a conceptualization.

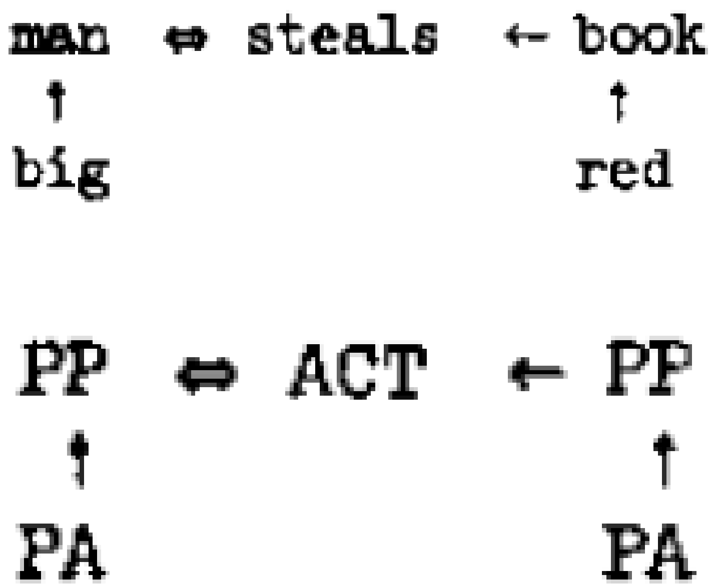
<u>Assisting categories</u>

PA	PP-assister; corresponds roughly to an adjective.
AA	ACT-assister; an adverb.

Phase 3: Semantic Analysis

Conceptual Parsing

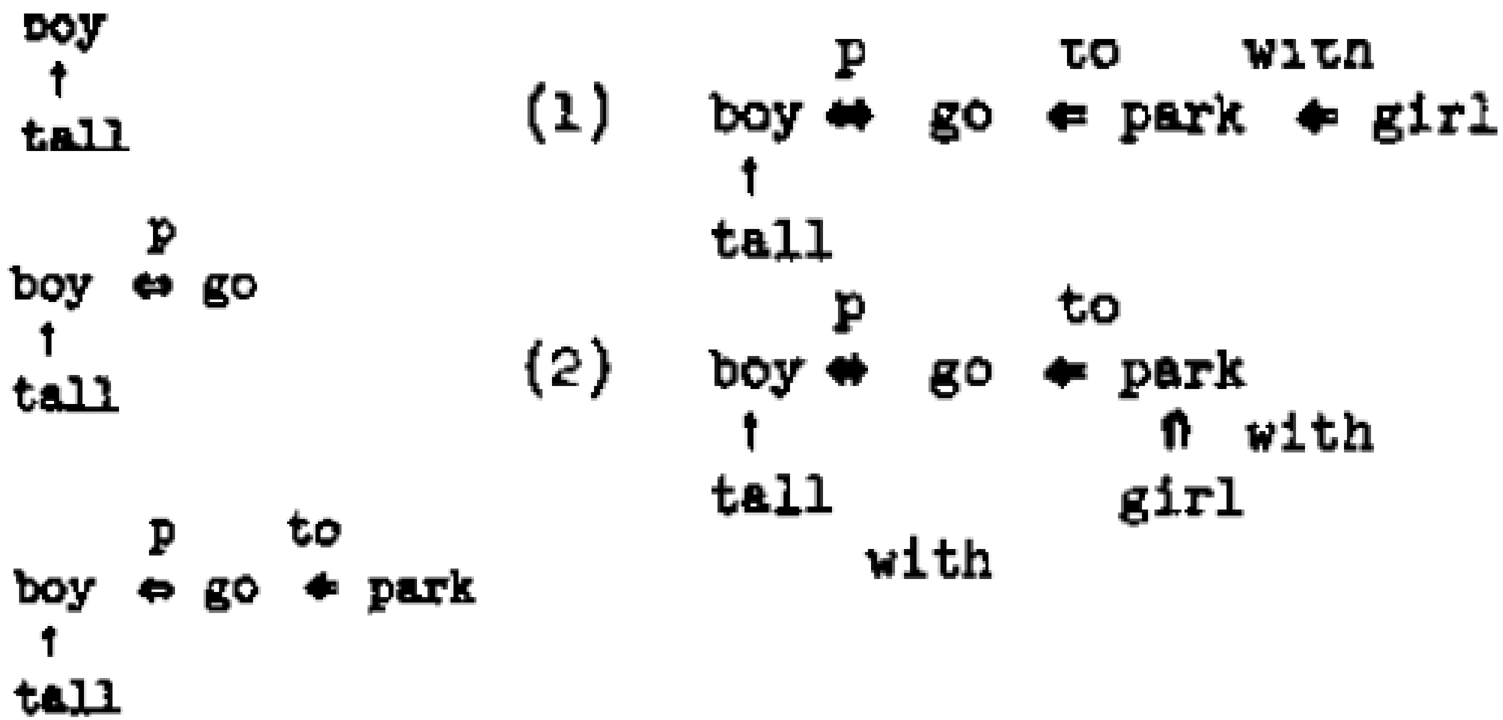
The big man steals the red book



Phase 3: Semantic Analysis

Conceptual Parsing

The tall boy went to the park with a girl



Phase 3: Semantic Analysis

Compositional Semantic Interpretation

- Here, syntactic parsing and semantic interpretation are treated as two separate steps.
- The two must mirror each other in well-defined ways.
- Steps to generate a semantic interpretation for a strict syntactic parsing of a sentence:
  - Look up for each word in a lexicon that contains one or more definitions for the word, each stated in terms of the chosen target representation. These definitions must describe how the idea that corresponds to the word is to be represented, and they may also describe how the idea represented by this word may combine with the ideas represented by other words in the sentence.
  - Use structure information in the output to provide additional constraints on how to combine individual words to form larger meaningful words.

Phase 3: Semantic Analysis

Compositional Semantic Interpretation: Montague Semantics

- Montague semantics is a theory of natural language semantics and of its relation with syntax.

- The most important features of the theory are its use of model theoretic semantics which is nowadays commonly used for the semantics of logical languages and its adherence to the principle of compositionality—that is, the meaning of the whole is a function of the meanings of its parts and their mode of syntactic combination.

- Important relationships are:

  - Entailment
  - Valid reasoning
  - Synonymy
  - Ambiguity

Phase 3: Semantic Analysis

Entailment

Whenever A is accepted as being true, B must also be accepted as being true, on grounds of meaning properties. Sentence (1) entails sentence (2)

- (1) Mary is singing and dancing.
- (2) Mary is singing.

Entailment with inverse relationship, 4 entails 3

- (3) No-one is singing and dancing.
- (4) No-one is singing.

Entailment with transitive and intransitive verbs: 5 entails 6

- (5) John cools the soup.
- (6) The soup cools

Phase 3: Semantic Analysis

Valid Reasoning

A ‘valid reasoning’ is a generalization of entailment that involves more sentences than two. If someone accepts (7) and (8), they may conclude correctly (9):

- (7) John sings.
- (8) John is a man.
- (9) A man sings.

- (10) John knows **whether** Mary comes.
- (11) Mary comes.
- (12) John knows **that** Mary comes

- (10) John knows whether Mary comes.
- (11a) Mary does not come.
- (12a) John knows that Mary does not come

Phase 3: Semantic Analysis

Synonymy

A special case of entailment is 'synonymy.' Sentences are called synonymous just in case they entail each other. For example, sentences (13) and (14) are synonymous:

- (13) John or Mary comes.
- (14) John comes or Mary comes.

- (15) Mary admires herself, and Sue **does too**.
- (16) Mary admires herself, and Sue admires herself

Ambiguity

- (17) Mary believes that she is ill, and Sue **does too**.
- (18) Mary believes that she is ill, and Sue believes that Mary is ill.

- (19) Sue believes that Mary is ill
- (20) Sue believes that she (= Sue) is ill.

- (21) Ten years ago, John met the president of the USA.
- (22) The president of the USA is Bush.

Phase 3: Semantic Analysis

Extended Reasoning with a knowledge base

- A significant amount of world knowledge is required to do semantic interpretation. The knowledge is required to enable the system to choose among competing interpretations.

- John made a huge wedding cake with chocolate icing
- John made a huge wedding cake with Bill's mixer.
- John made a huge wedding cake with a giant tower covered with roses.
- John made a cherry pie with a giant tower covered with roses.

- Knowledge enables the system to accept meanings that it has not been explicitly told about.

- John made a huge wedding cake with chocolate icing
- John made a huge wedding cake with Bill's mixer.
- John made a huge wedding cake with a giant tower covered with roses.
- John made a cherry pie with a giant tower covered with roses.

Phase 4: Discourse and pragmatic processing

- **Identical entities**

➢ Bill had a red balloon
➢ John wanted **it**. (It is anaphoric reference or anaphora)

- **Parts of entities**

➢ Ram opened the dress that he just bought.
➢ **The collar** had a damage.

- **Parts of actions**

➢ Jim went on a business trip to Delhi
➢ **He** left by the early morning flight.

- **Entities involved in actions**

➢ The bank was broken into last week.
➢ **They** took the cash and jewels.

Phase 4: Discourse and pragmatic processing

- **Elements of sets**

➢ The stickers we have in store are the lion, tiger, giraffe and octopus.
➢ I will take the **octopus**.

- **Names of individuals**

➢ Harry went to the party.

- **Casual chain**

➢ It was raining heavily yesterday.
➢ The schools were closed yesterday.

- **Planning sequences**

➢ Kiran wanted a new house.
➢ Kiran decided to get a better job.

Phase 4: Discourse and pragmatic processing

- **Illocutionary force**

➢ It sure is going to rain today/

- **Implicit presuppositions**

➢ Did the Delhi flight come late?

- **Prerequisites**

➢ The current focus of the dialogue.
➢ A model of each participant's current belief.
➢ The goal-driven character of the dialogue.
➢ The rules of the conversation shared by all participants.
➢ The schools were closed yesterday.

- **Planning sequences**

➢ Kiran wanted a new house.
➢ Kiran decided to get a better job.

Phase 4: Discourse and pragmatic processing

- **Using focus in understanding**

- Focus on relevant parts of the available knowledge base.
- Use that knowledge to resolve ambiguities and to make connections among things that were said.