Natural Language Processing (NLP)

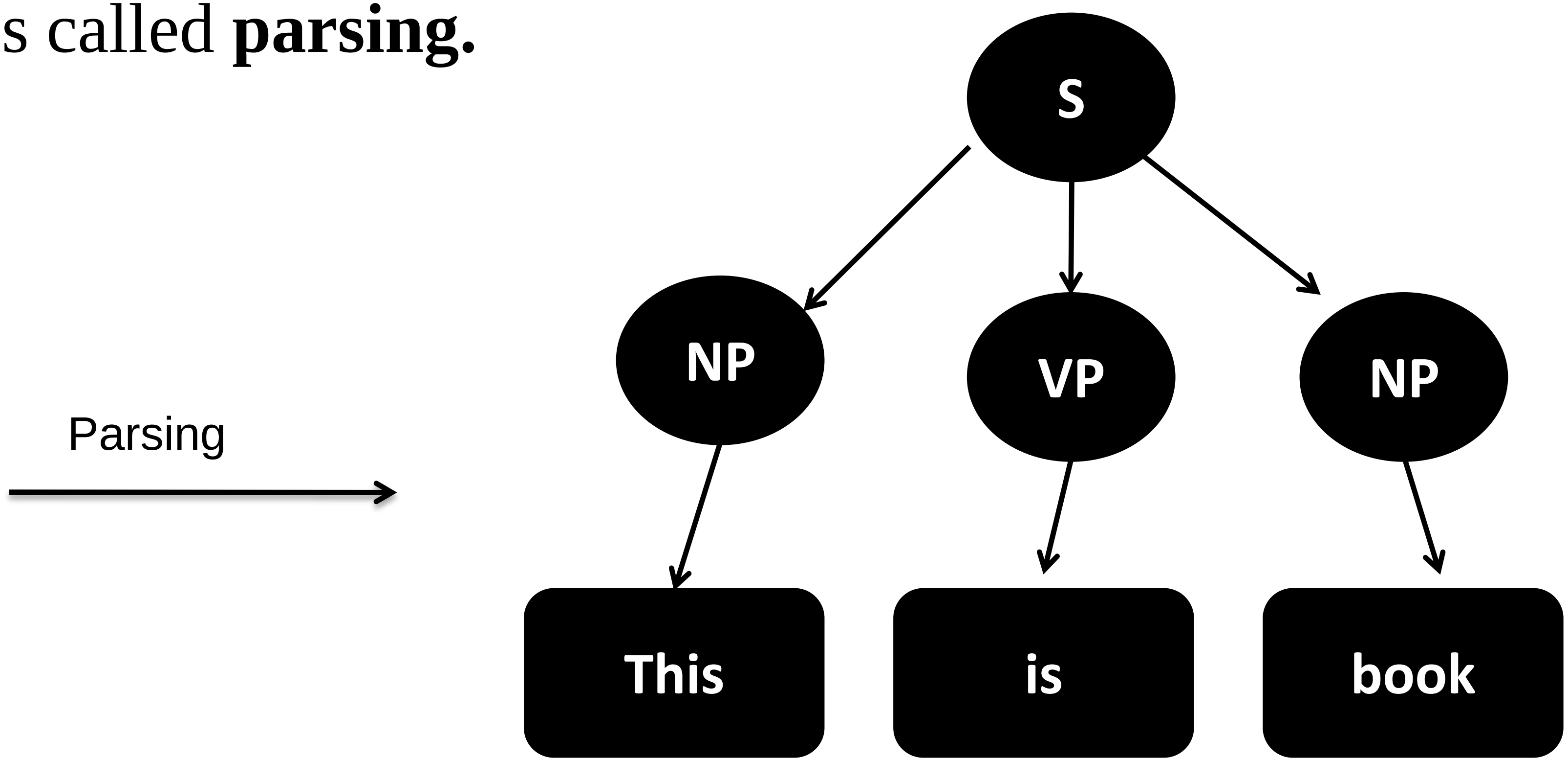
Phases of NLP ?

1. **Morphological Analysis**
2. **Syntactic Analysis**
3. **Semantic Analysis**
4. **Discourse integration**
5. **Pragmatic Analysis**

Syntactic Processing

- Syntactic Processing is the step in which a flat input sentence is converted into a hierarchical structure that **corresponds** to the **units of meaning** in the sentence.
- This process is called **parsing.**

This is book.



Syntactic Processing

- It plays an important role in natural language understanding systems for two reasons:
  - ***Semantic processing must operate on sentence "constituent"***.
  - Parsing **reduces** the number of constituents that semantics can consider.
  - **Syntactic parsing is computationally "less" expensive than is semantic processing.**

Syntactic Processing

– often possible to extract the meaning of a sentence without using grammatical facts, but not always

- The satellite orbited Mars
- Mars orbited the satellite

Note: Orbit, path described by one celestial body in its revolution about another

Syntactic Processing

- Syntactic processing have **two** main components:
  - A ***declarative representation, called a grammar***
  - A ***procedure, called parser,*** that compares the grammar against input sentences to produce parsed structures.

Grammars and Parsers

- Parsing does
  - Assign syntactic tags (subject, verb, object etc.)
  - Assign syntactic relation between words
- Two types of Parsing technique
  - ***rule-based parsing***
  - ***statistical parsing***.

Grammars and Parsers

Rule-based parsing

- syntactic structure of language is provided as **production rules** that are similar to context free rules.
- Production rules are defined using **non-terminal** and **terminal symbols.**

Grammars and Parsers

Statistical parsing

- Require
  - large corpora
  - and linguistic knowledge is represented as statistical parameters or probability.

Grammars and Parsers

- If sentence is syntactically **correct**. The Parser generates **parse tree**.
- Parsing can be done in two methods:
  - **top-down**
  - **bottom up**

Grammars and Parsers

- **Bottom-up parsing:**
  - we start with the words in the sentence
  - apply grammar rules in the backward direction until a single tree is produced whose root matches with start symbol.

Grammars and Parsers

- **Top-down parsing:**
  - we start with start symbol
  - apply grammar rules in forward direction until the terminal symbol of the parse tree corresponds to the word in the sentence.

Grammars and Parsers

- The choice between these two approaches is similar to the choice between **forward and backward** reasoning in other problem-solving tasks.
- The most important consideration is the **branching factor**. Is it greater going backward or forward?
- Sometimes these two approaches are combined to a single method called “**bottom-up parsing with top-down filtering**”.

Grammars and Parsers

- Consider the simple context free grammar for English language.

Rules	Dictionary Words
<S> -> <NP><VP>	<Det> -> a |an | the
<NP> -> <Det><Noun>	<Noun> -> girl | apple | song
<NP> -> <Det><Adj><Noun>	<Adj> -> cute | smart
<NP> -> <Adj><Noun>	<verb> -> sings | eats | ate
<VP> -> <Verb>	
<VP> -> <verb><NP>	

- The symbol $\longrightarrow$ is used for ‘defined as’.
- Vertical bar | for alternative definitions (OR)
- The S for sentence
- NP for noun phrase
- VP for verb phrase.

Rules	Dictionary Words
<s> -> <NP><VP>	<Det> -> a |an | the
<NP> -> <Det><Noun>	<Noun> -> girl | apple | song
<NP> -> <Det><Adj><Noun>	<Adj> -> cute | smart
<NP> -> <Adj><Noun>	<verb> -> sings | eats | ate
<VP> -> <Verb>	
<VP> -> <verb><NP>	

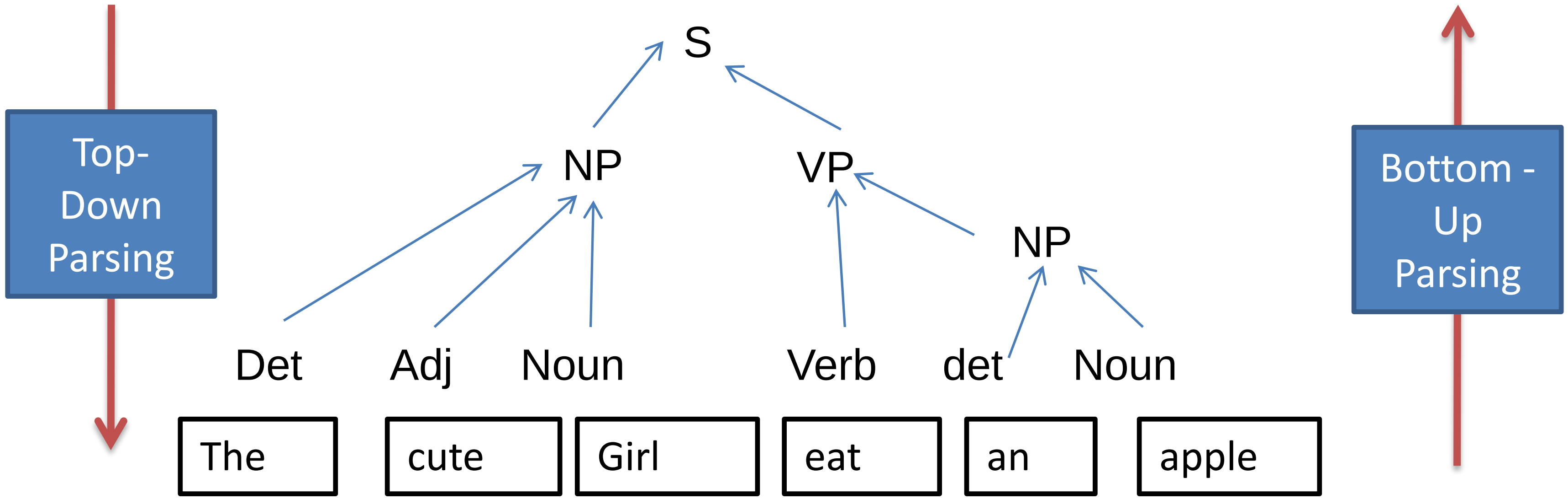
a) **the boy eats an apple**

Rules	Dictionary Words
<s> -> <NP><VP>	<Det> -> a |an | the
<NP> -> <Det><Noun>	<Noun> -> girl | apple | song
<NP> -> <Det><Adj><Noun>	<Adj> -> cute | smart
<NP> -> <Adj><Noun>	<verb> -> sings | eats | ate
<VP> -> <Verb>	
<VP> -> <verb><NP>	

b) The cute girl sings a song

Parse tree

- The cute girl ate an apple



Grammars and Parsers

- First rule can be read as " A sentence is composed of a noun phrase followed by Verb Phrase"; Vertical bar is OR ; ε represents empty string.
- Symbols that are further expanded by rules are called **non terminal symbols.**
- Symbols that correspond directly to strings that must be found in an input sentence are called **terminal symbols.**
- Pure context free grammar is **not** effective for describing natural languages.
- NLPs have **less** in common with computer language processing systems such as compilers.

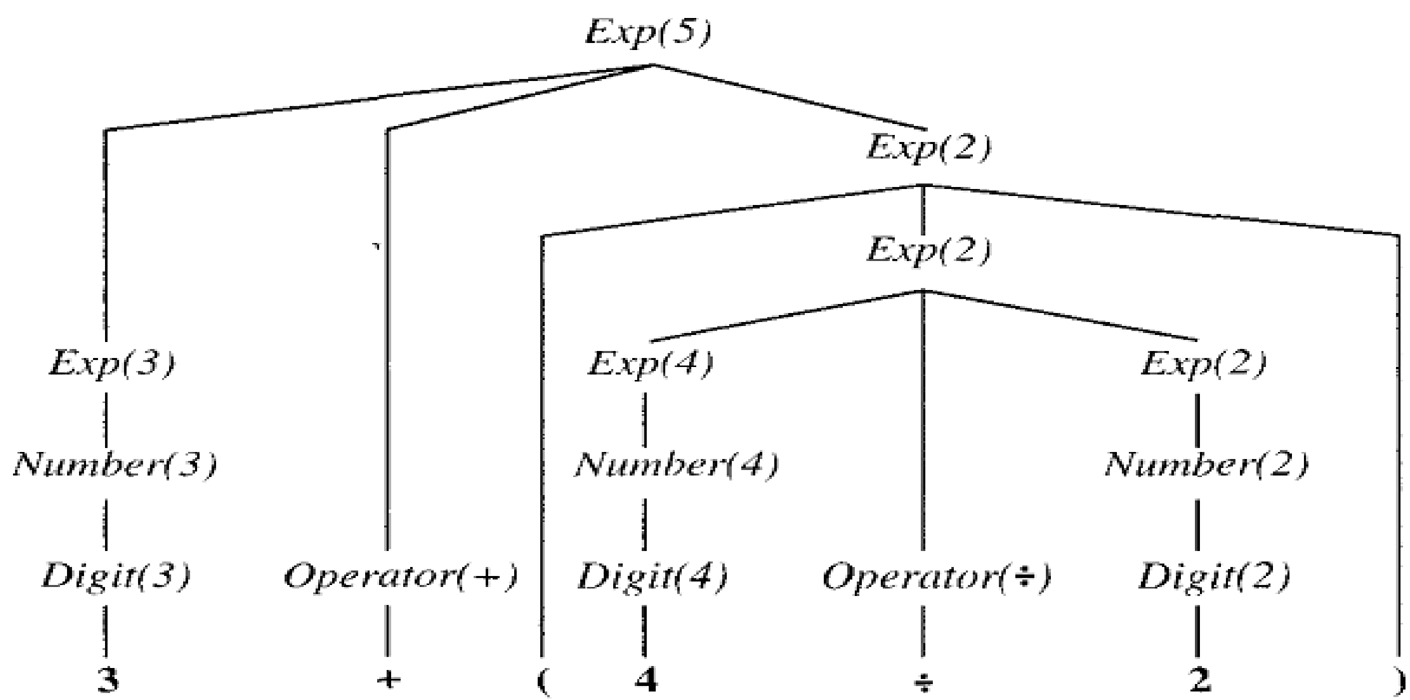
A grammar for arithmetic expressions, with semantics.

$$Exp(sem) \rightarrow Exp(sem_1)\ Operator(op) Exp(sem_2)\ \{sem = Apply(op, sem_1, sem_2)\}$$
$$Exp(sem) \rightarrow (\ Exp(sem)\ )$$
$$Exp(sem) \rightarrow Number(sem)$$
$$Digit(sem) \rightarrow sem\ \{0 < sem < 9\}$$
$$Number(sem) \rightarrow Digit(sem)$$
$$Number(sem) \rightarrow Number(sem_1) Digit(sem_2)\ \{sem = 10\ \text{x}\ sem_1 + sem_2\}$$
$$Operator(sem) \rightarrow sem\ \{sem\ \text{G}\ \{+, -, \div, \text{x}\}\}$$

Parse tree with semantic interpretations for the string "3 + (4 ÷ 2)".

A grammar for arithmetic expressions, with semantics.

$$Exp(sem) \rightarrow Exp(sem_1)\ Operator(op)\ Exp(sem_2)\ \{sem = Apply(op, sem_1, sem_2)\}$$
$$Exp(sem) \rightarrow (\ Exp(sem)\ )$$
$$Exp(sem) \rightarrow Number(sem)$$
$$Digit(sem) \rightarrow sem\ \{0 < sem < 9\}$$
$$Number(sem) \rightarrow Digit(sem)$$
$$Number(sem) \rightarrow Number(sem_1)\ Digit(sem_2)\ \{sem = 10 \times sem_1 + sem_2\}$$
$$Operator(sem) \rightarrow sem\ \{sem\ \mathrm{G}\ \{+, -, \div, \times\}\}$$



Parse tree with semantic interpretations for the string "3 + (4 ÷ 2)".