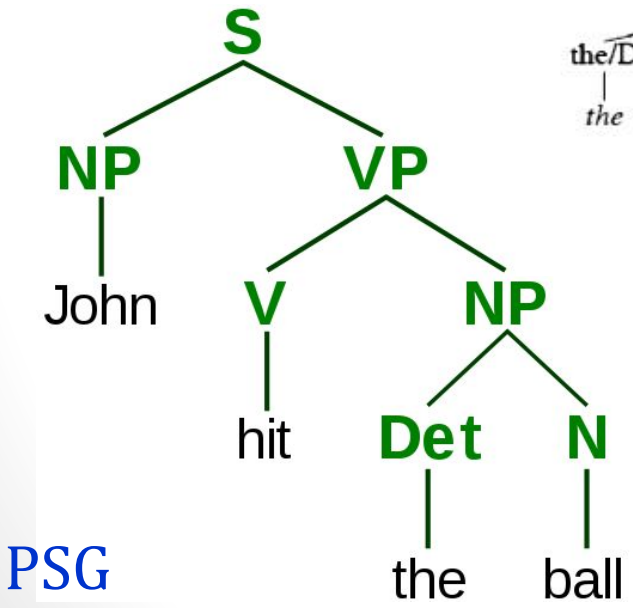
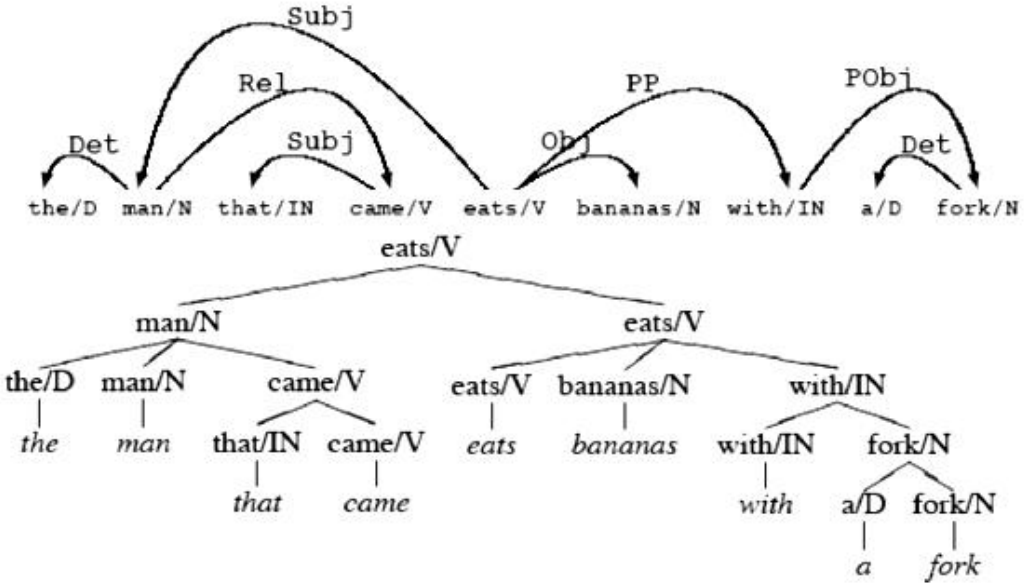
Topic 9 (Pt 1) :
Syntactic Parsing



PSG



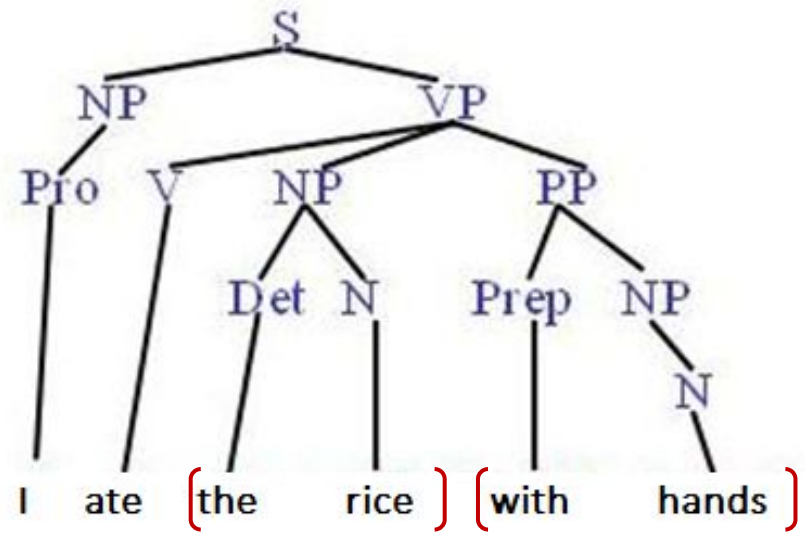
DG

Phrase Structure Grammar

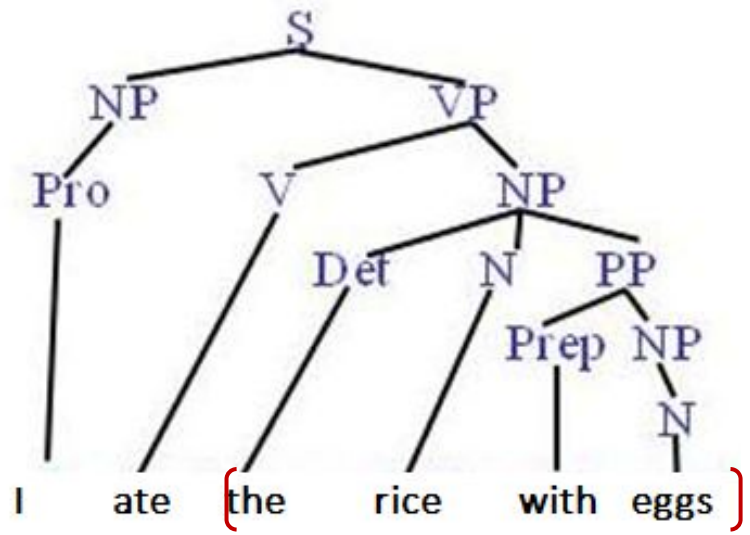
- *Phrase structure grammar (PSG)* is a type of generative grammar in which constituent structures are represented by *phrase structure rules* or *rewrite rules*.
- A phrase structure (or constituent) functions as the base component in the classic form of transformational grammar introduced by Noam Chomsky in the late 1950s.

Syntactic Parsing

- Produce the correct syntactic parse tree for a sentence based on PSG



How we ate the rice



What we ate with the rice

SVO - SUBJECT VERB OBJECT (ENG)

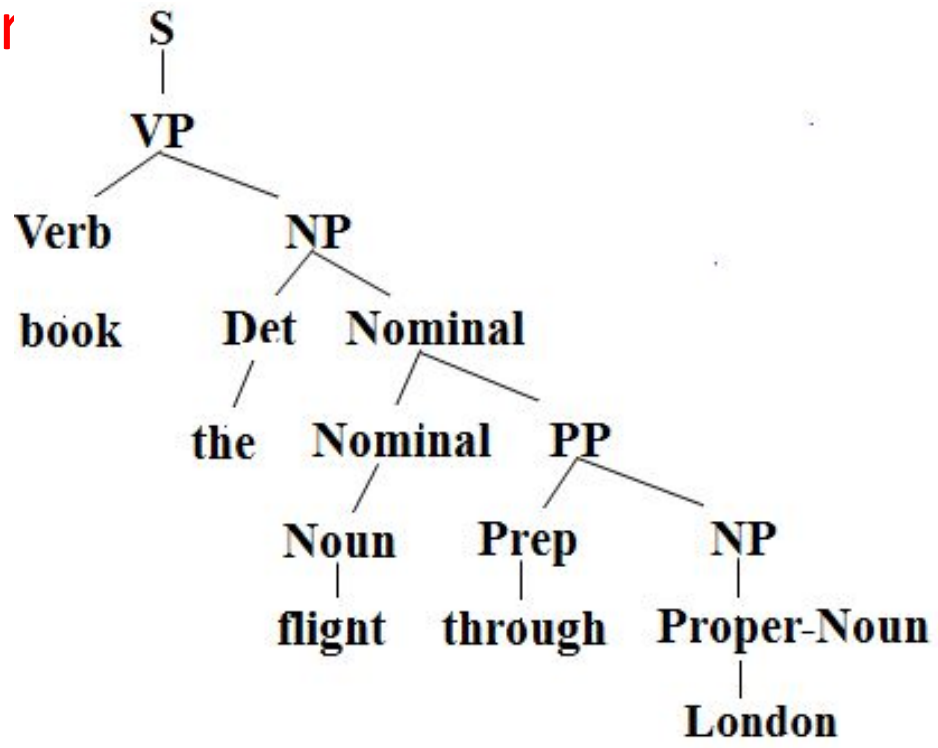
私は手でご飯を食べました

Watashi wa te de gohan o tabemashita (I with hands the rice ate)

SOV - SUBJECT OBJECT VERB (JAP)

Sentence Generation

- Sentences are generated by recursively rewriting the start symbol using the productions until only terminals symbols remain producing a derivation or par



Parsing as Search

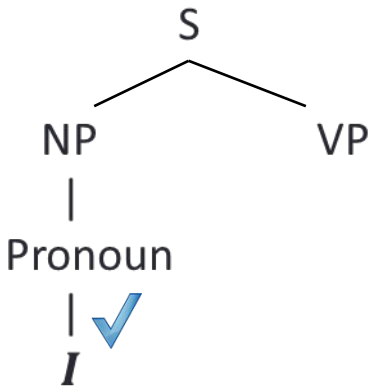
- Parsing searches for space of derivations that derives the given string
- In **Finite State Automata (FSA)**, the parser searches through the space of all possible paths defined by the **structure of the FSA**, through the automaton.
- In **syntactic parsing**, the parser can be viewed as searching through the space of all possible parse trees defined by the **grammar** to find the correct parse tree for the sentence.

Parsing as Search

- Parsing searches for space of derivations that derives the given string
- Two approaches of parsing:
  - **Top-down parsing**
    - Start searching space of derivations from the start symbol.
    - Grammar rules are applied from left to right
  - **Bottom-up parsing**
    - Start searching space of reversed derivations from terminal symbols (i.e., words) in the string
    - Look for places in parse tree where right hand-side of some rule fit.

Top Down Parsing

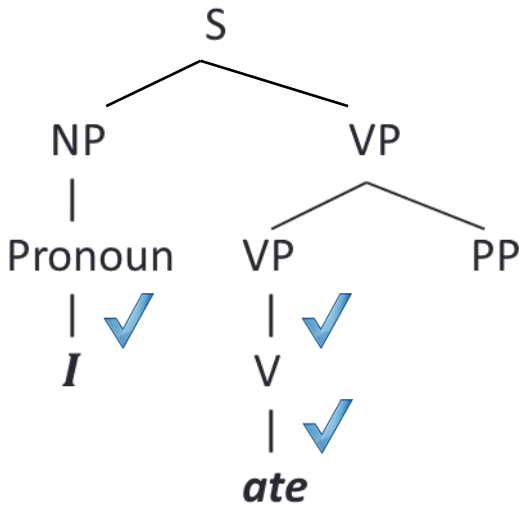
- Example : "*I ate the rice with eggs.*"



S (Sentence)
NP (Noun Phrase)
VP (Verb Phrase)
Pronoun

Top Down Parsing

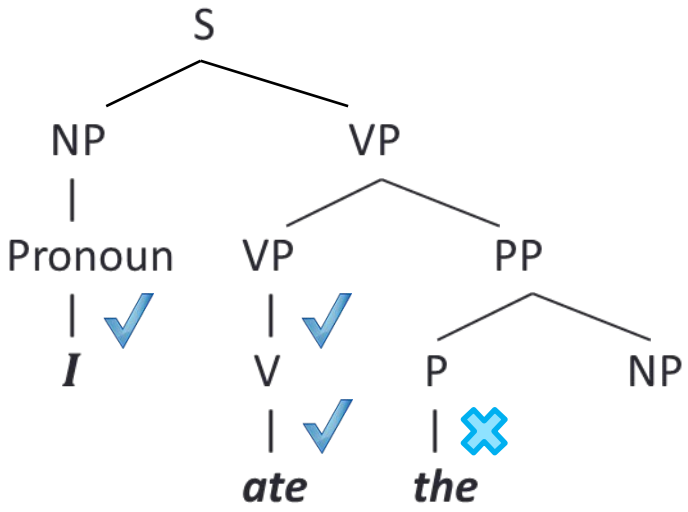
- Example : "*I ate the rice with eggs.*"



S (Sentence)
NP (Noun Phrase)
VP (Verb Phrase)
PP (Prepositional Phrase)
Pronoun
V(Verb)

Top Down Parsing

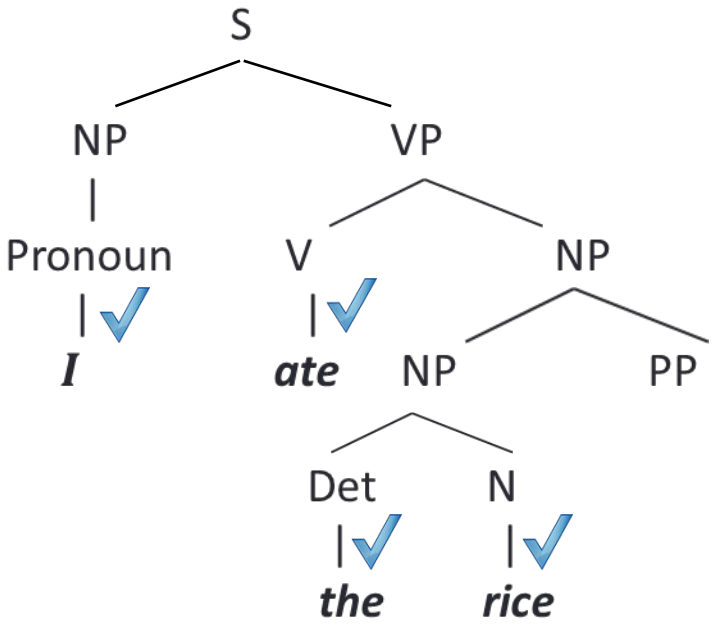
- Example : "*I ate the rice with eggs.*"



S (Sentence)
NP (Noun Phrase)
VP (Verb Phrase)
PP (Prepositional Phrase)
Pronoun
V(Verb)
P (Preposition)

Top Down Parsing

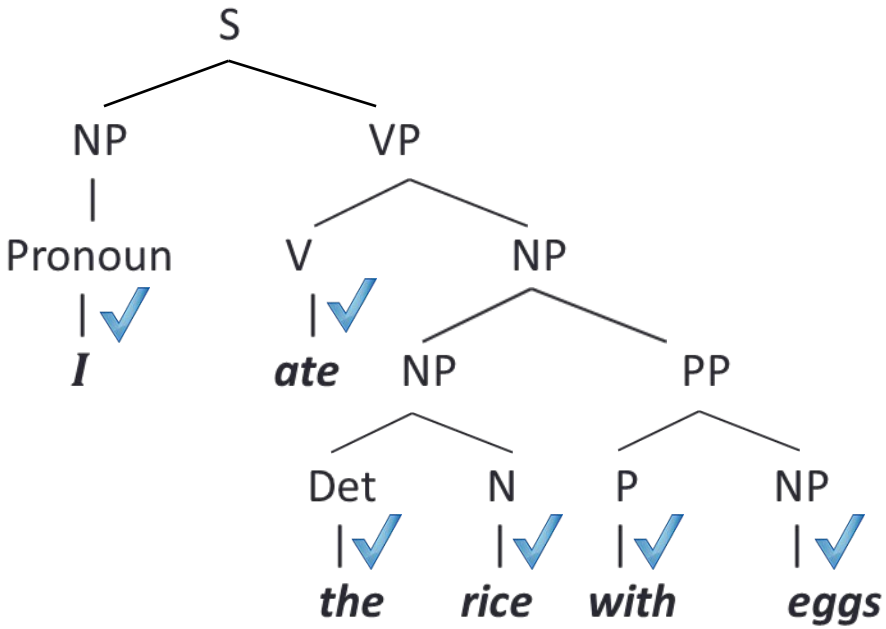
- Example : "*I ate the rice with eggs.*"



S (Sentence)
NP (Noun Phrase)
VP (Verb Phrase)
PP (Prepositional Phrase)
Pronoun
V (Verb)
Det (Determiner)
N (Noun)

Top Down Parsing

- Example : "*I ate the rice with eggs.*"



S (Sentence)
NP (Noun Phrase)
VP (Verb Phrase)
PP (Prepositional Phrase)
Det (Determiner)
Pronoun
V(Verb)
N (Noun)
P (Preposition)

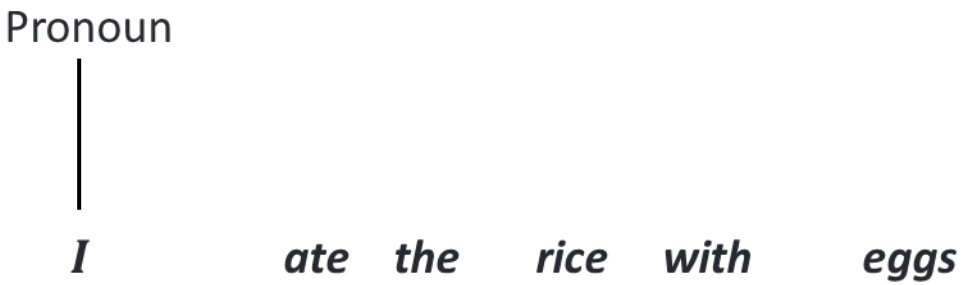
Bottom Up Parsing

- Example : *"I ate the rice with eggs."*

I *ate* *the* *rice* *with* *eggs*

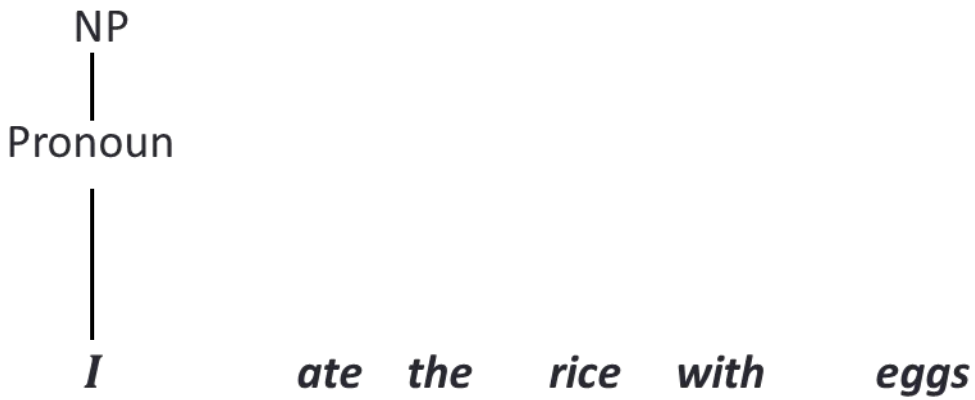
Bottom Up Parsing

- Example : "*I ate the rice with eggs.*"



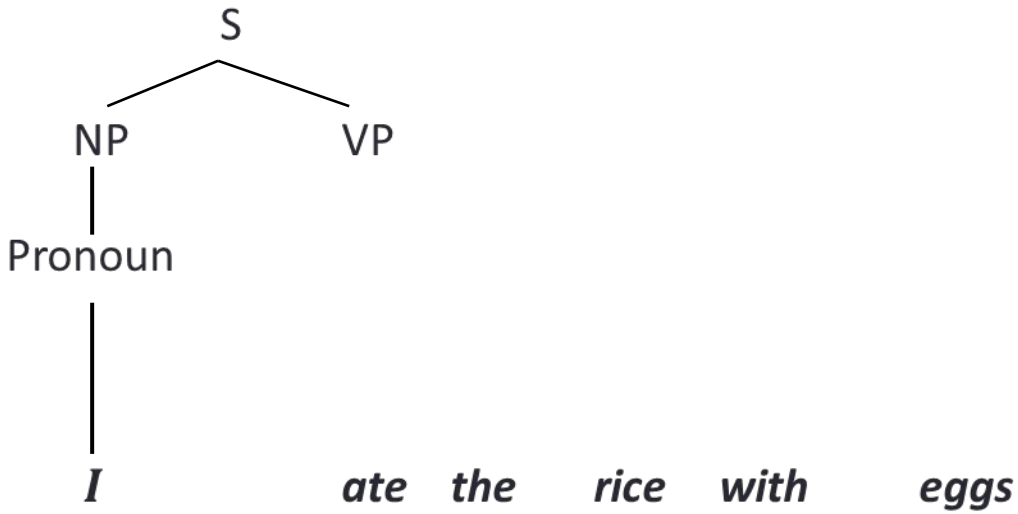
Bottom Up Parsing

- Example : "*I ate the rice with eggs.*"



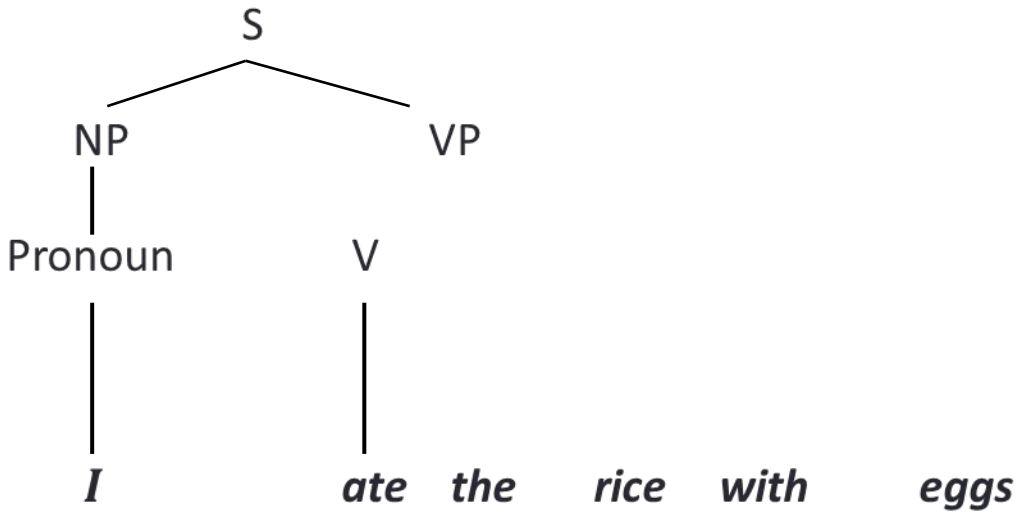
Bottom Up Parsing

- Example : "*I ate the rice with eggs.*"



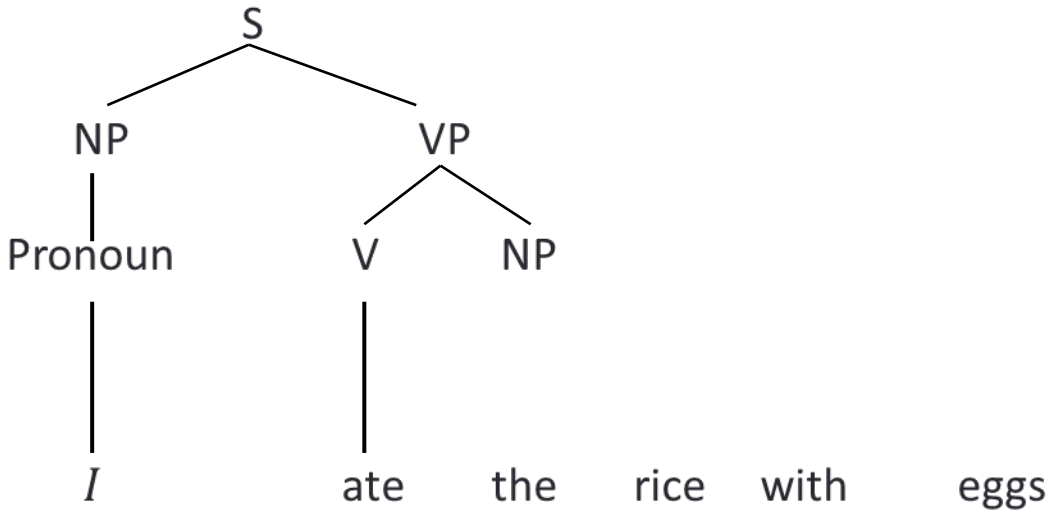
Bottom Up Parsing

- Example : "*I ate the rice with eggs.*"



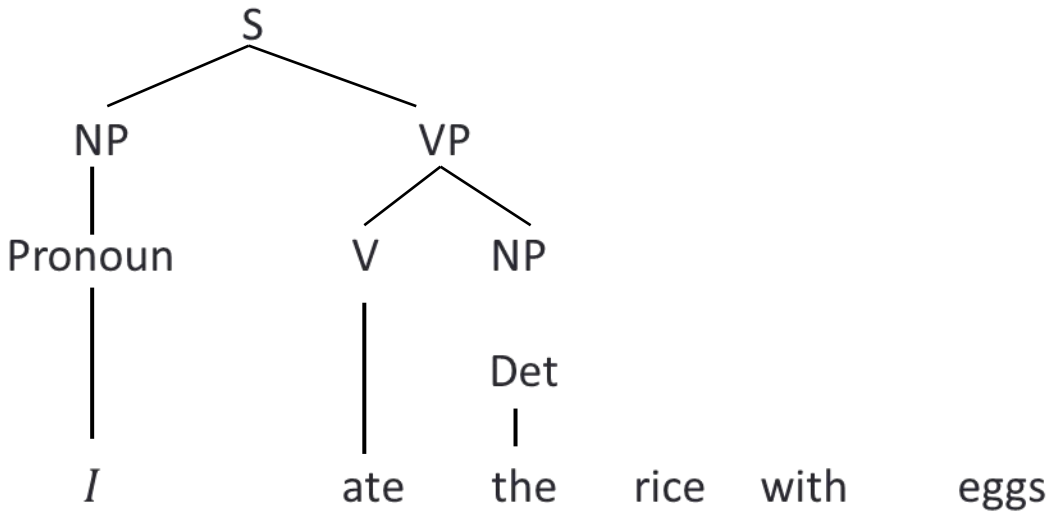
Bottom Up Parsing

- Example : "*I ate the rice with eggs.*"



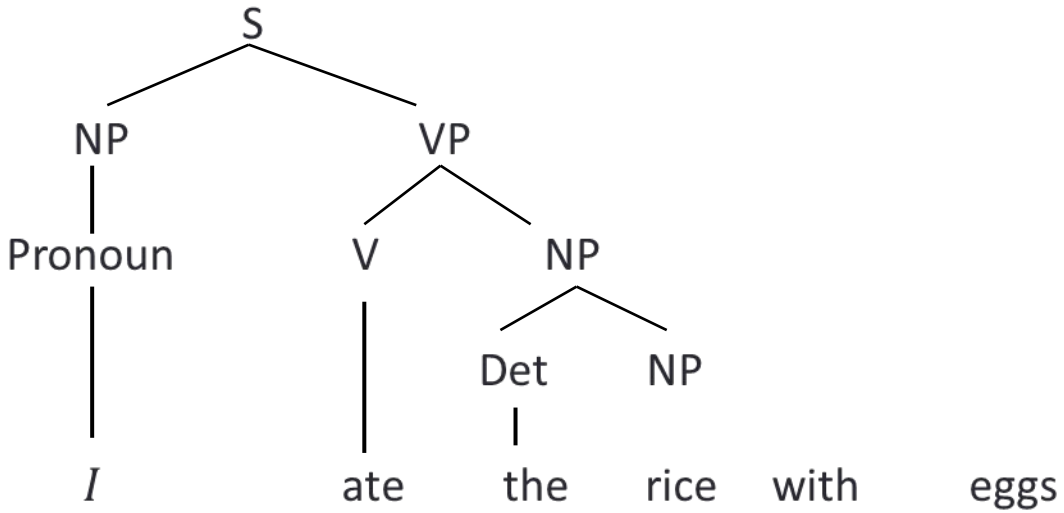
Bottom Up Parsing

- Example : "*I ate the rice with eggs.*"



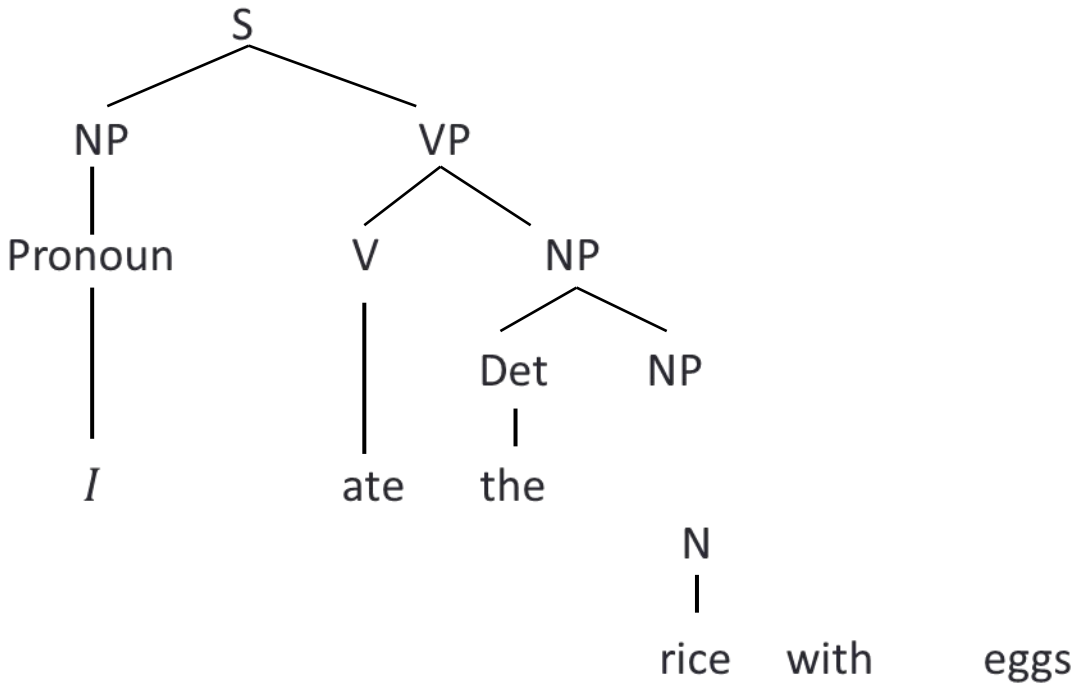
Bottom Up Parsing

- Example : "*I ate the rice with eggs.*"



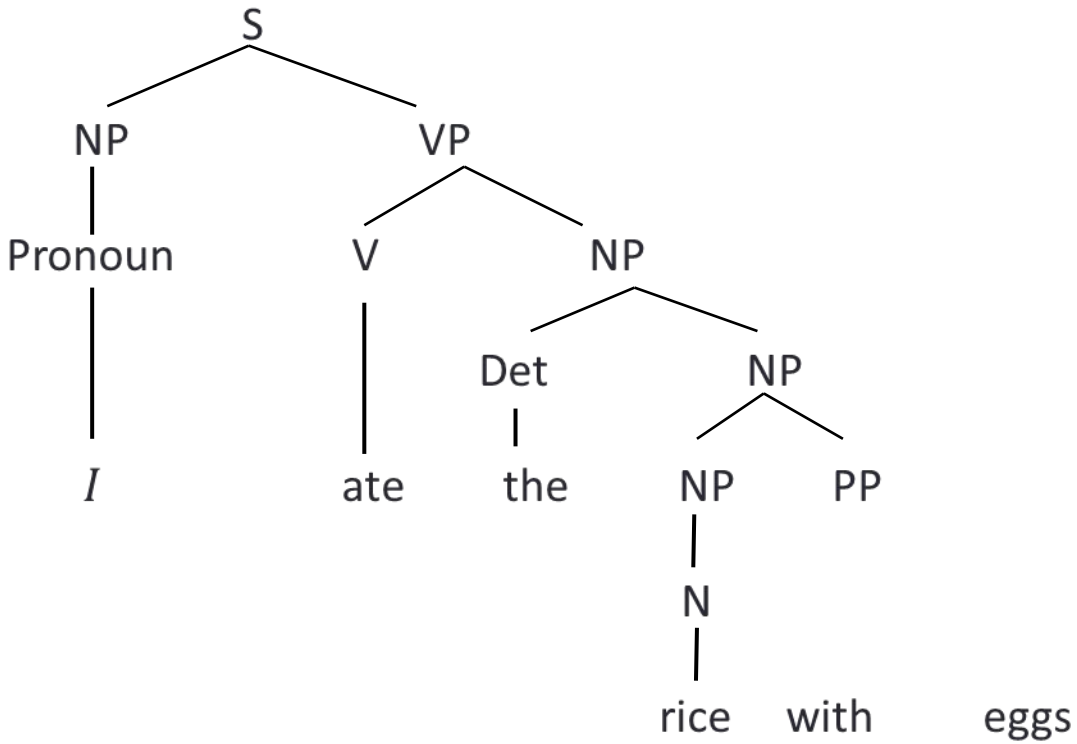
Bottom Up Parsing

- Example : "*I ate the rice with eggs.*"



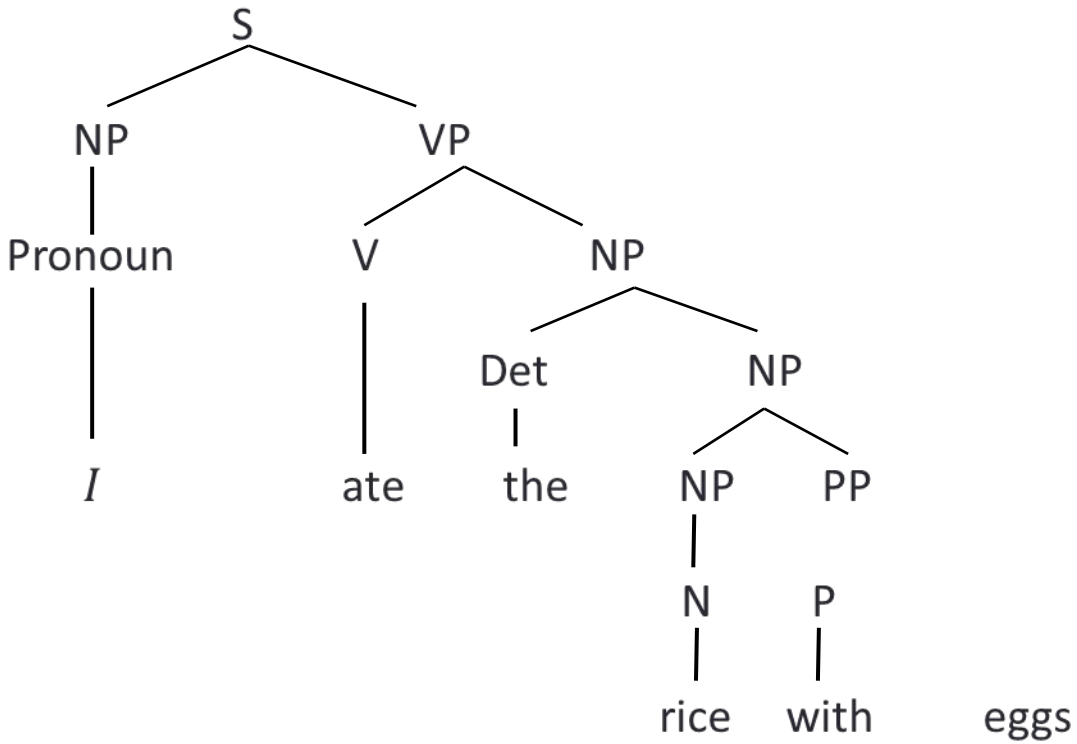
Bottom Up Parsing

- Example : "*I ate the rice with eggs.*"



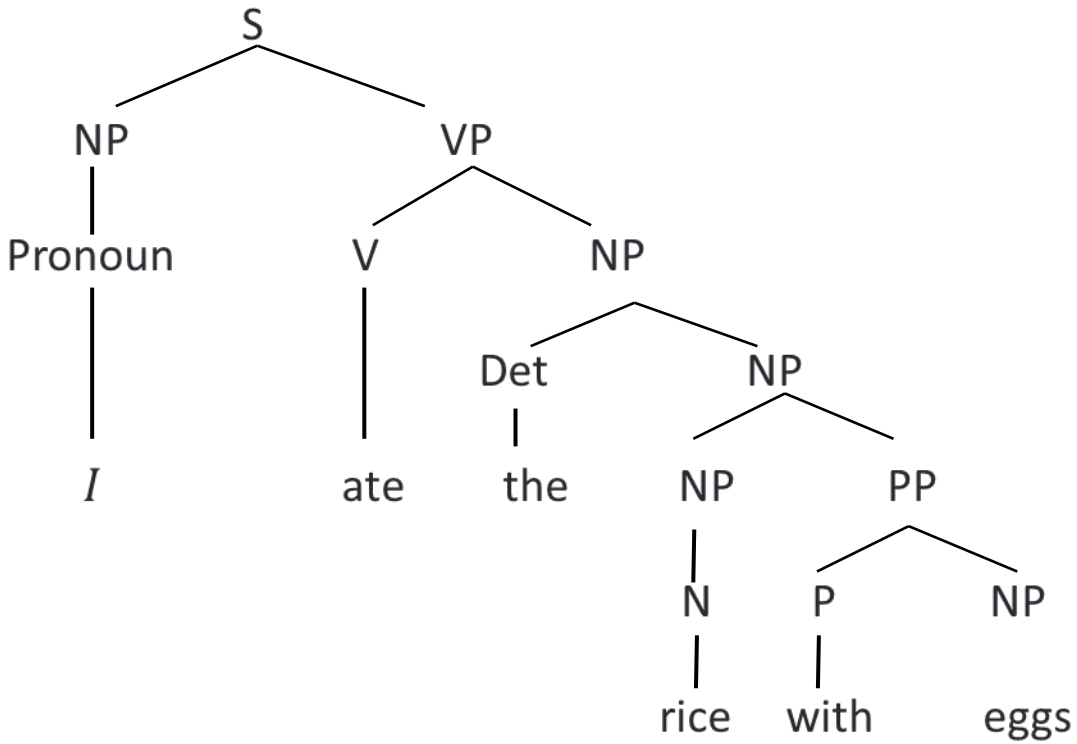
Bottom Up Parsing

- Example : "*I ate the rice with eggs.*"



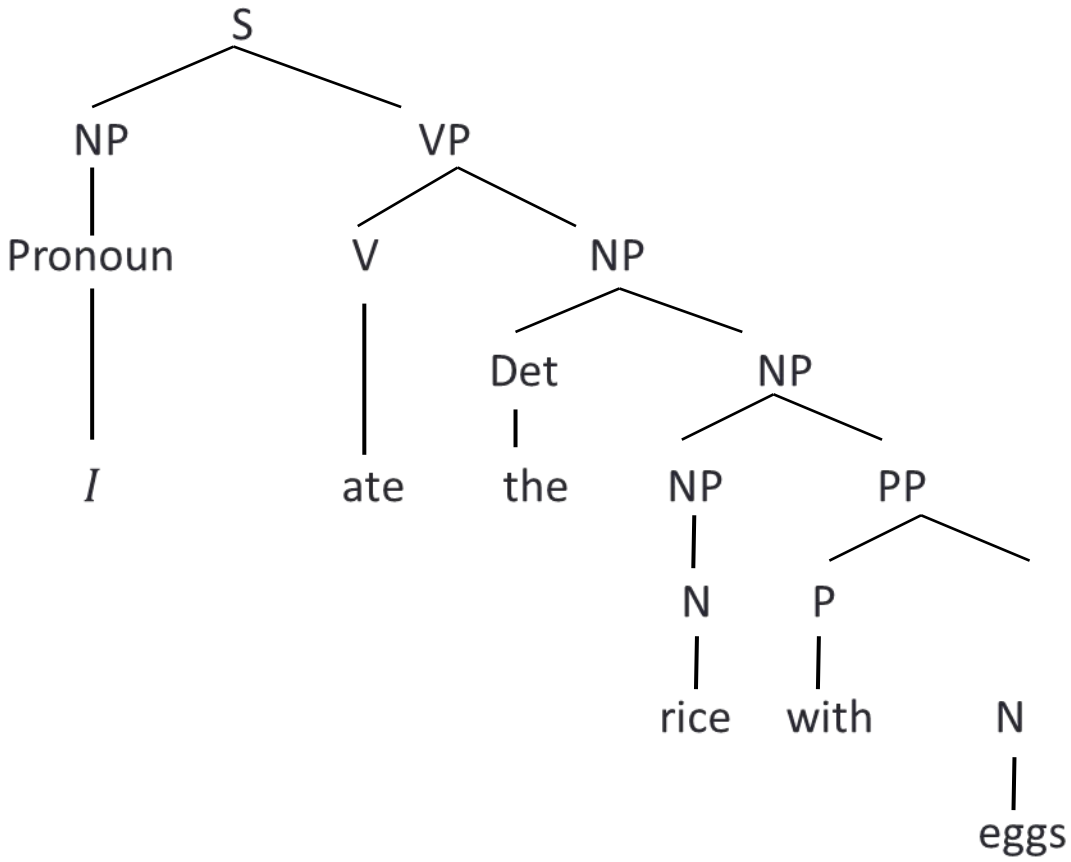
Bottom Up Parsing

- Example : "*I ate the rice with eggs.*"



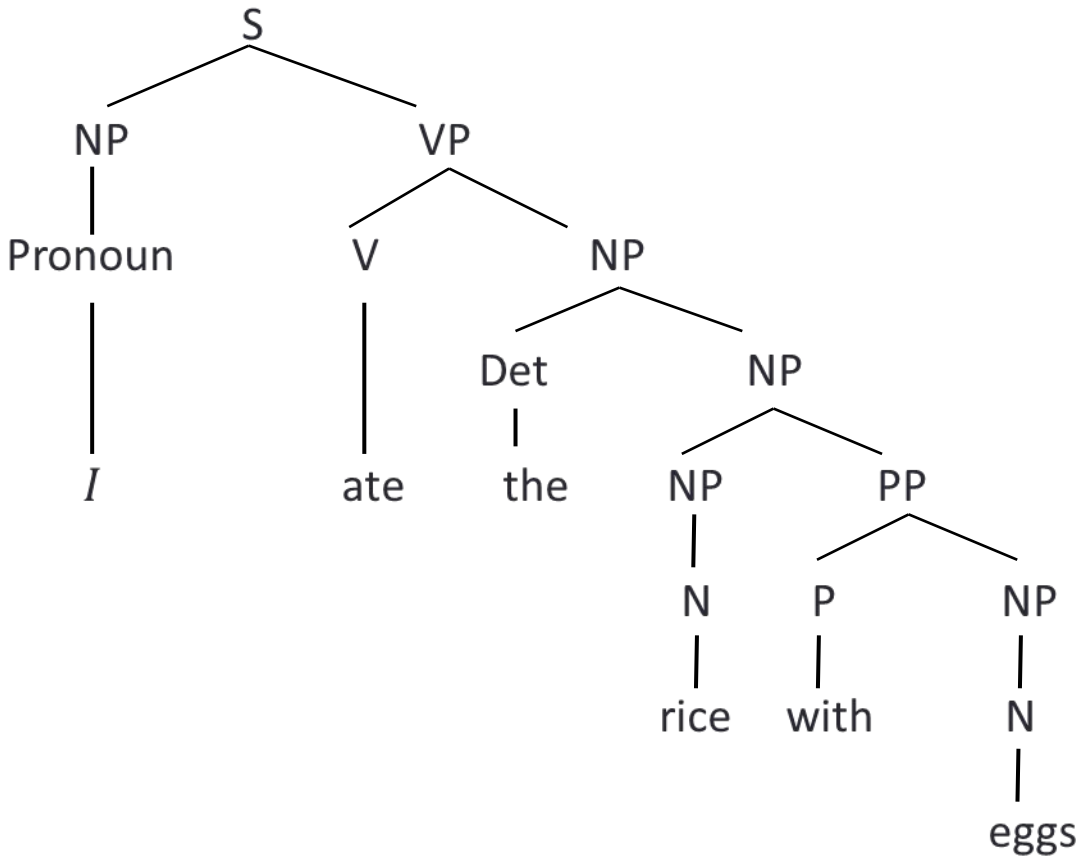
Bottom Up Parsing

- Example : "*I ate the rice with eggs.*"



Bottom Up Parsing

- Example : "*I ate the rice with eggs.*"



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

Top Down vs Bottom Up

- Top down parsers spend considerable effort on *S* trees that are not consistent with the input.
- Top down parsers can generate trees before ever examining the input
- Bottom-up never suggest trees that are not at least locally grounded in the actual input.

Parsing with NLTK

- Sentence: “I shot the elephant in my pajamas”
  - Define context free grammar

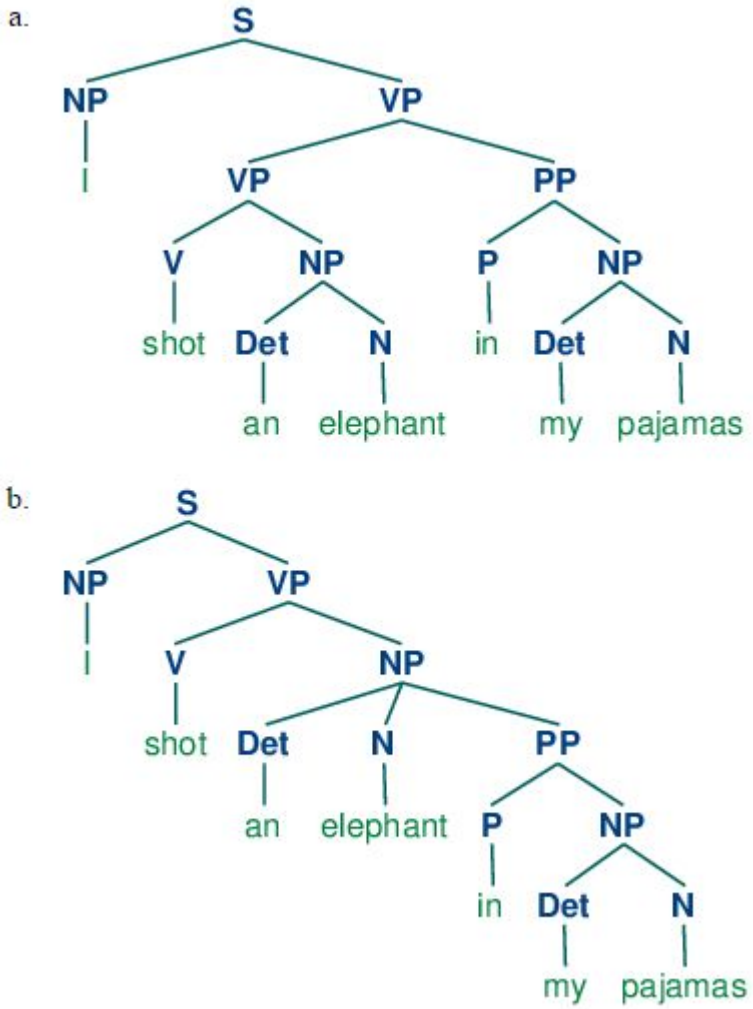
```
>>> grammar = nltk.CFG.fromstring("""
S -> NP VP
PP -> P NP
NP -> Det N | Det N PP | 'I'
VP -> V NP | VP PP
Det -> 'an' | 'my'
N -> 'elephant' | 'pajamas'
V -> 'shot'
P -> 'in'
""")
```

Parsing with NLTK

- Parse the sentence

```
>>> sent = ['I', 'shot', 'an', 'elephant', 'in', 'my', 'pajamas']
>>> parser = nltk.ChartParser(groucho_grammar)
>>> for tree in parser.parse(sent):
...     print(tree)
...
(S
  (NP I)
  (VP
    (VP (V shot) (NP (Det an) (N elephant)))
    (PP (P in) (NP (Det my) (N pajamas)))))
(S
  (NP I)
  (VP
    (V shot)
    (NP (Det an) (N elephant) (PP (P in) (NP (Det my) (N pajamas))))))
```

Parse Tree


a.
S
NP
VP
I
VP
PP
V
NP
P
NP
shot
Det
N
in
Det
N
an
elephant
my
pajamas
b.
S
NP
VP
I
V
NP
shot
Det
N
PP
an
elephant
P
NP
in
Det
N
my
pajamas

Exercise

- Parse the following sentences with NLTK:
  1. *Fighting animals could be dangerous.*
  2. *Visiting relatives can be tiresome.*
- Steps:
  - Create the CFG for the sentences
  - Parse the sentence