CYK Parsing

CS6370:JAN-MAY 2021

Language & Grammar

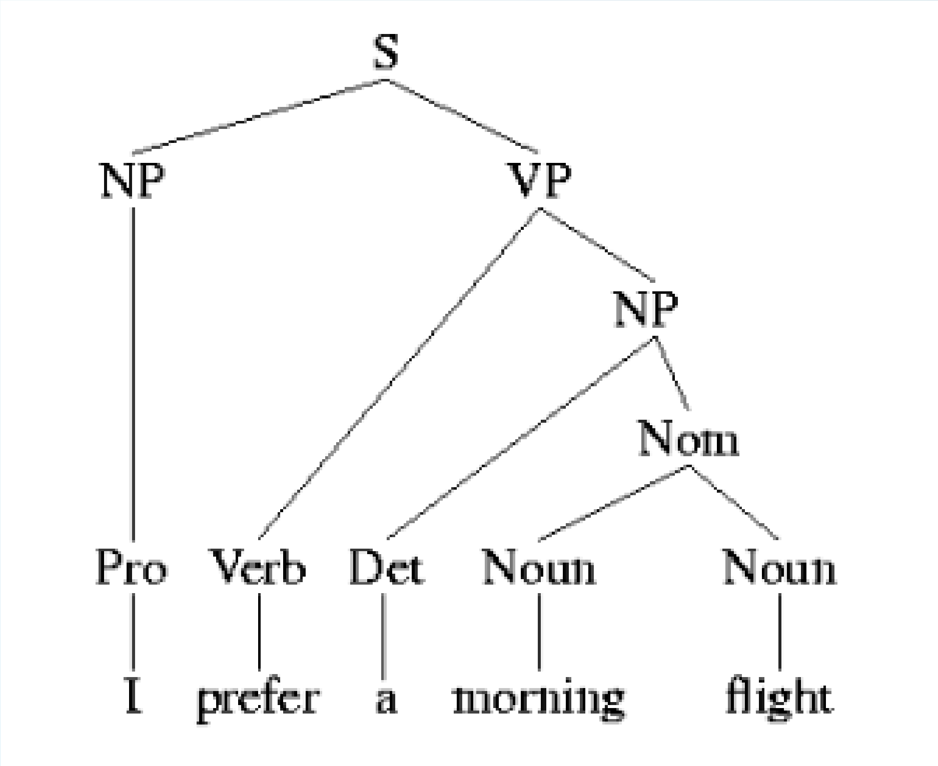
- Language - The set of all strings that can be derived from a grammar

Context Free Grammar

- Language - The set of all strings that can be derived from a grammar
- Context Free Grammar
    - Most commonly used mathematical system for modelling constituent structure in English and other NLs.

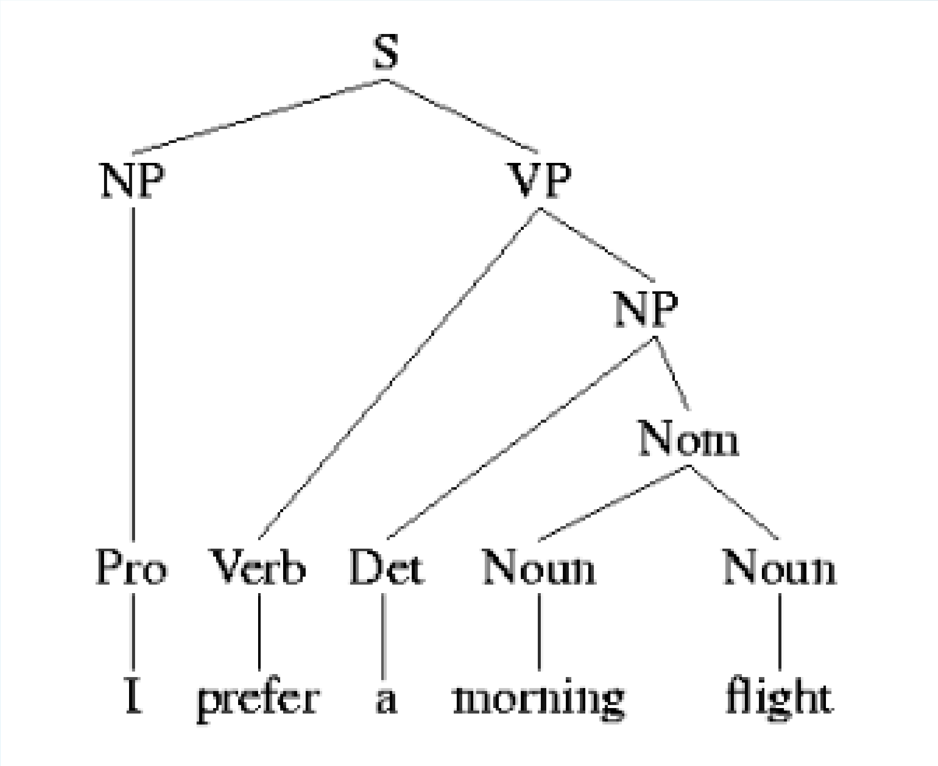
Natural Language - Constituent Structures

- In syntactic analysis, a constituent is a word or a group of words that function as a single unit within a hierarchical structure.



Natural Language - Constituent Structures

- In syntactic analysis, a constituent is a word or a group of words that function as a single unit within a hierarchical structure.
- Example: NP: “ a morning flight”
- VP: “prefer a morning flight”



Context Free Grammar

- Language - The set of all strings that can be derived from a grammar
- Context Free Grammar
  - Most commonly used mathematical system for modelling constituent structure in English and other NLs.
  - Powerful enough to express sophisticated relations among the words in a sentence, yet computationally tractable enough that efficient algorithms exist for parsing sentences with them.

Context Free Grammar

- A Context-free Grammar (CFG) is a 4-tuple:G= (N,T, P, S) where:
  - N: Non-Terminals (finite set of NT symbols)
  - T: Terminals (finite set of tokens – lexicon)
  - P: Production Rules (a->b where a is in N, b is in $(N \cup T)^*$)
  - S: Start symbol

Example Grammar 1

- Non-Terminals = {S, A ,B}
- Terminals = {a,b}
- Start symbol = {S}
- Production Rules = {$S \rightarrow AB$, $A \rightarrow a$, $B \rightarrow b$}

Example Grammar 1

- Non-Terminals (N) = {S, A ,B}
- Terminals (T) = {a,b}
- Start symbol = {S}
- Production Rules = {S → AB, A → a, B → b}
- **Language = {ab}**

Example Grammar 2

- Non Terminals = {S, A, B}
- Terminals = {a, b}
- Production Rules = {$S \rightarrow AB$, $A \rightarrow aA \mid a$, $B \rightarrow bB \mid b$}

Example Grammar 2

- Non Terminals = {S, A, B}
- Terminals = {a, b}
- Production Rules = {S → AB, A → aA | a, B → bB | b}

- L(G) = {ab, a^2b, ab^2, a^2b^2, ………}

ab

$S \rightarrow AB$

$A \rightarrow aA$

$A \rightarrow a$

aaA

aaaA

aaaa

$S \rightarrow aAb$

$\rightarrow aab$

Example Grammar 2

- Non Terminals = {S, A, B}
- Terminals = {a, b}
- Production Rules = $\{S \rightarrow AB,\ A \rightarrow aA \mid a,\ B \rightarrow bB \mid b\}$

- $L(G) = \{ab, a^2b, ab^2, a^2b^2, \ldots\ldots\ldots\} = \{a^m b^n, m>0; n>0\}$

CFG for English – An Example

- Set of Terminals

Noun → *flight* | *breeze* | *trip* | *morning* | ...
Verb → *is* | *prefer* | *like* | *need* | *want* | *fly* ...
Adjective → *cheapest* | *non-stop* | *first* | *latest* | *other* | *direct* | ...
Pronoun → *me* | *I* | *you* | *it* | ...
Proper-Noun → *Alaska* | *Baltimore* | *Los Angeles* | *Chicago* | *United* | *American* | ...
Determiner → *the* | *a* | *an* | *this* | *these* | *that* | ...
Preposition → *from* | *to* | *on* | *near* | ...
Conjunction → *and* | *or* | *but* | ...

The lexicon for L_0

CFG for English - An Example

- Non-terminals and Rules

$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Pronoun \\
&\quad \mid Proper\text{-}Noun \\
&\quad \mid Det\ Nominal \\
Nominal &\rightarrow Noun\ Nominal \\
&\quad \mid \quad Noun \\
VP &\rightarrow Verb \\
&\quad \mid Verb\ NP \\
&\quad \mid Verb\ NP\ PP \\
&\quad \mid Verb\ PP \\
PP &\rightarrow Preposition\ NP
\end{aligned}
$$

CFG for English - An Example

- Non-terminals and Rules

$S \rightarrow NP\ VP$	I + want a morning flight
$NP \rightarrow Pronoun$	I
$\mid Proper\text{-}Noun$	Los Angeles
$\mid Det\ Nominal$	a + flight
$Nominal \rightarrow Noun\ Nominal$	morning + flight
$\mid Noun$	flights
$VP \rightarrow Verb$	do
$\mid Verb\ NP$	want + a flight
$\mid Verb\ NP\ PP$	leave + Boston + in the morning
$\mid Verb\ PP$	leaving + on Thursday
$PP \rightarrow Preposition\ NP$	from + Los Angeles

Chomsky Normal Form

- A CFG(context free grammar) is in CNF(Chomsky normal form) if <u>all production rules</u> <u>satisfy **one** of the following conditions</u>:
  - Start symbol generating null string ε. For example, A → ε.
  - A non-terminal generating two non-terminals. For example, **S → AB.**
  - A non-terminal generating a terminal. For example, **S → a.**

Any CFG can be converted into an equivalent CNF

- Example:
- $S \rightarrow a \mid aA \mid B$
- $B \rightarrow Aa \mid b$
- $A \rightarrow a$

Reference

[Chapter 13 in J&M Book](https://www.geeksforgeeks.org/converting-context-free-grammar-chomsky-normal-form/)

https://www.geeksforgeeks.org/converting-context-free-grammar-chomsky-normal-form/

Problem definition

- Given a CFG and a sentence, determine if the sentence can be generated by the given grammar

- Assumption: CFG is expressed in Chomsky Normal Form

Bottom-up Parsing

- Brute force method to tackle the membership problem
  - Start from the input string, and construct all possible subtrees
  - Stop when the input is spanned by S (the start symbol), or when no more rules apply (the string is not in the language of G)

Bottom-up Parsing

- Brute force method to tackle the membership problem
  - Start from the input string, and construct all possible subtrees
  - Stop when the input is spanned by S (the start symbol), or when no more rules apply (the string is not in the language of G)
- Cons of the naive bottom-up parsing approach
  - Repeated sub-structures
  - Will explore subtrees that do not fit the entire sentence

We saw a DP solution to Edit distance earlier

- What does each cell in a edit distance table stand for?

			1	2	3	4
			P	A	R	K
		0	1	2	3	4
1	S	1	1	2	3	4
2	P	2	1	2	3	4
3	A	3	2	1	2	3
4	K	4	3	2	2	2
5	E	5	4	3	3	3

We saw a DP solution to Edit distance earlier

- What does each cell in a edit distance table stand for?

			1	2	3	4
			P	A	R	K
		0	1	2	3	4
1	S	1	1	2	3	4
2	P	2	1	2	3	4
3	A	3	2	1	2	3
4	K	4	3	2	2	2
5	E	5	4	3	3	3

We saw a DP solution to Edit distance earlier

- What does each cell in a edit distance table stand for?

			1	2	3	4
			P	A	R	K
		0	1	2	3	4
1	S	1	1	2	3	4
2	P	2	1	2	3	4
3	A	3	2	1	2	3
4	K	4	3	2	2	2
5	E	5	4	3	3	3

Edit distance of a sub-problem :

Edit distance(SP, PAR)

CYK Parsing (Cocke-Younger-Kasami Algorithm)

- For a string of length **n**, CYK algorithm uses a table of **n** rows and **n** columns
- However, not all cells of the table are filled by the algorithm

CYK Parsing (Cocke-Younger-Kasami Algorithm)

- For a string of length **n**, CYK algorithm uses a table of **n** rows and **n** columns
- However, not all cells of the table are filled by the algorithm

CYK Parsing

- For a string of length **n**, CYK algorithm uses a table of **n** rows and **n** columns
- However, not all cells of the table are filled by the algorithm

	1	2	3	4
1				
2				
3				
4				

CYK Parsing

- For a string of length **n**, CYK algorithm uses a table of **n** rows and **n** columns
- However, not all cells of the table are filled by the algorithm

	1	2	3	4
1				
2				
3				
4				

CYK Parsing

- For a string of length **n**, CYK algorithm uses a table of **n** rows and **n** columns

What does each cell stand for?

	1	2	3	4
1				
2				
3				
4				

CYK Parsing

Sentence: We love online classes

What does each cell stand for?

	1	2	3	4
1	We	love	online	classes
2				
3				
4				

CYK Parsing

Sentence: We love online classes

What does each cell stand for?

	1	2	3	4
1	We	love	online	classes
2	We love	love online	online classes	
3				
4				

CYK Parsing

Sentence: We love online classes

	1	2	3	4
1	We	love	online	classes
2	We love	love online	online classes	
3	We love online	love online classes		
4	We love online classes			

What does each cell stand for?

CYK Parsing

Sentence: We love online classes

Constituents in the syntactic structure corresponding to these **substrings**

	1	2	3	4
1	We	love	online	classes
2	We love	love online	online classes	
3	We love online	love online classes		
4	We love online classes			

Example

Grammar

S → NP VP

VP → VP PP

VP → eats   VP → V NP

PP → P NP

NP → D N

NP → she

V → eats

P → with

N → fish

N → fork

D → a

Example

She eats a fork with a fish

Grammar

S → NP VP

VP → VP PP

VP → eats , VP → V NP

PP → P NP

NP → D N

NP → she

V → eats

P → with

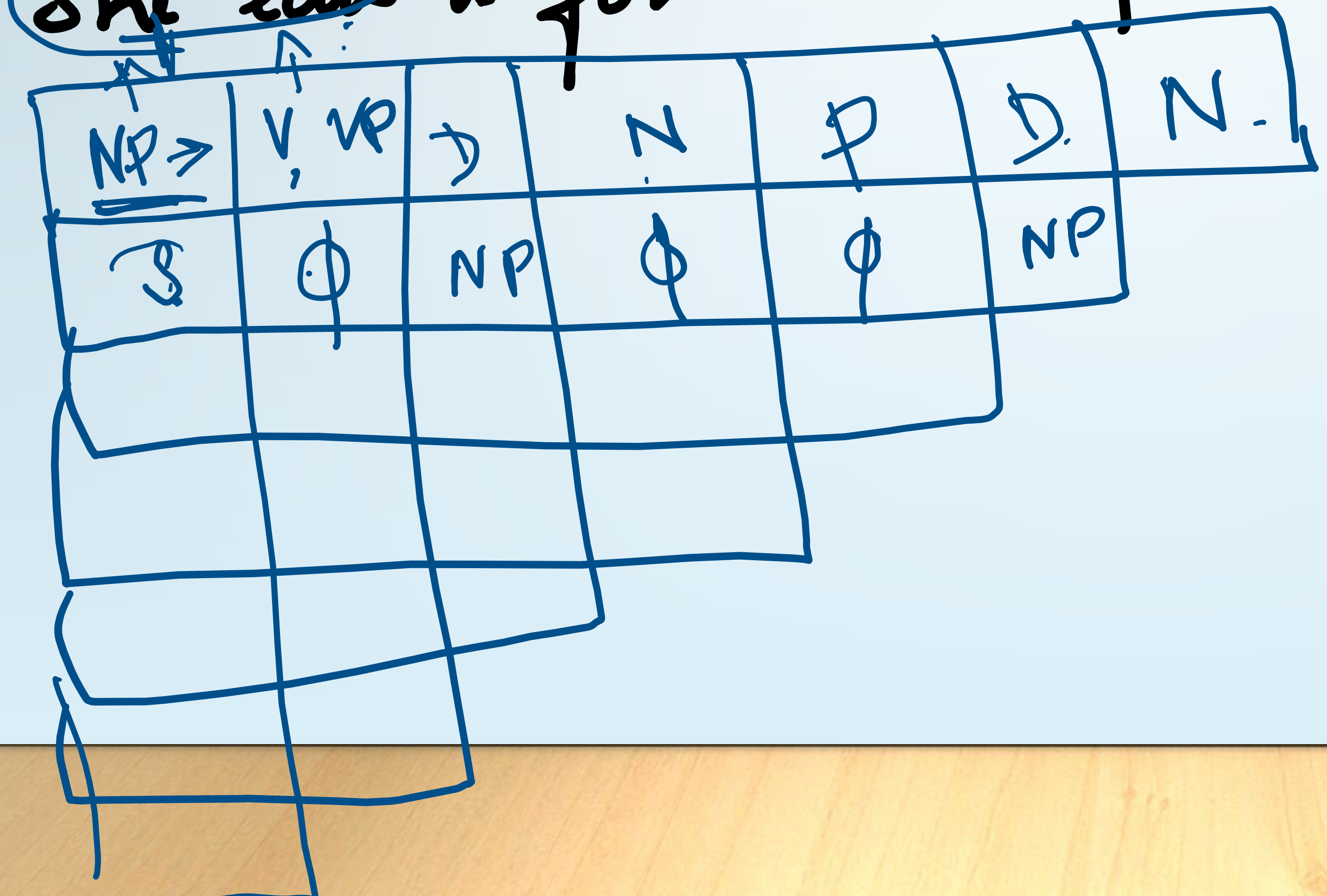
N → fish

N → fork

D → a

Example

She eats a fork with a fish



Grammar

S → NP VP

VP → VP PP

VP → eats , VP → V NP

PP → P NP

NP → D N

NP → she

V → eats

P → with

N → fish

N → fork

D → a

Example

She	eats	a	fork	with	a	fish
NP	V, VP	D	N	P	D	N
S	Φ	NP	Φ	Φ	NP	

Grammar

$S \rightarrow NP\ VP$

$VP \rightarrow VP\ PP$

$VP \rightarrow eats$, $VP \rightarrow V\ NP$

$PP \rightarrow P\ NP$

$NP \rightarrow D\ N$

$NP \rightarrow she$

$V \rightarrow eats$

$P \rightarrow with$

$N \rightarrow fish$

$N \rightarrow fork$

$D \rightarrow a$

Example

She	eats	a	fork	with	a	fish
NP	V, VP	D	N	P	D	N
S	ϕ	NP	ϕ	ϕ	NP	
ϕ	VP	ϕ	ϕ	PP		

Grammar

$S \to NP\ VP$

$VP \to VP\ PP$

$VP \to eats$, $VP \to V\ NP$

$PP \to P\ NP$

$NP \to D\ N$

$NP \to she$

$V \to eats$

$P \to with$

$N \to fish$

$N \to fork$

$D \to a$

Example

She	eats	a	fork	with	a	fish
NP	V, VP	D	N	P	D	N
S	ϕ	NP	ϕ	ϕ	NP	
ϕ	VP	ϕ	ϕ	PP		
S	ϕ	ϕ	ϕ			

Grammar

S → NP VP

VP → VP PP

VP → eats, VP → V NP

PP → P NP

NP → D N

NP → she

V → eats

P → with

N → fish

N → fork

D → a

Example

She	eats	a	fork	with	a	fish
NP	V, VP	D	N	P	D	N
S	ϕ	NP	ϕ	ϕ	NP	
ϕ	VP	ϕ	ϕ	PP		
S	ϕ	ϕ	ϕ			
ϕ	ϕ	ϕ				

Grammar

$S \rightarrow NP\ VP$

$VP \rightarrow VP\ PP$

$VP \rightarrow$ eats, $VP \rightarrow V\ NP$

$PP \rightarrow P\ NP$

$NP \rightarrow D\ N$

$NP \rightarrow$ she

$V \rightarrow$ eats

$P \rightarrow$ with

$N \rightarrow$ fish

$N \rightarrow$ fork

$D \rightarrow$ a

Example

She	eats	a	fork	with	a	fish
NP	V, VP	D	N	P	D	N
S	ϕ	NP	ϕ	ϕ	NP	
ϕ	VP	ϕ	ϕ	PP		
S	ϕ	ϕ	ϕ			
ϕ	ϕ	ϕ				
ϕ	VP					

Grammar

$S \rightarrow NP\ VP$

$VP \rightarrow VP\ PP$

$VP \rightarrow eats$, $VP \rightarrow V\ NP$

$PP \rightarrow P\ NP$

$NP \rightarrow D\ N$

$NP \rightarrow she$

$V \rightarrow eats$

$P \rightarrow with$

$N \rightarrow fish$

$N \rightarrow fork$

$D \rightarrow a$

Example

She	eats	a	fork	with	a	fish
NP	V, VP	D	N	P	D	N
S	ϕ	NP	ϕ	ϕ	NP	
ϕ	VP	ϕ	ϕ	PP		
S	ϕ	ϕ	ϕ			
ϕ	ϕ	ϕ				
ϕ	VP					
S						

Grammar

S → NP VP

VP → VP PP

VP → eats , VP → V NP

PP → P NP

NP → D N

NP → she

V → eats

P → with

N → fish

N → fork

D → a

Example 2

- Sentence : aaabb   Grammar:

$$
\begin{aligned}
S &\to AB \\
S &\to XB \\
T &\to AB \\
T &\to XB \\
X &\to AT \\
A &\to a \\
B &\to b
\end{aligned}
$$

CYK Parsing

a	a	a	b	b
{A}	{A}	{A}	{B}	{B}

$$S \to AB$$
$$S \to XB$$
$$T \to AB$$
$$T \to XB$$
$$X \to AT$$
$$A \to a$$
$$B \to b$$

CYK Parsing

a	a	a	b	b
{A}	{A}	{A}	{B}	{B}
‘aa’	‘aa’	‘ab’	‘ba’	

2nd row: To derive substrings of length 2

$$S \to AB$$
$$S \to XB$$
$$T \to AB$$
$$T \to XB$$
$$X \to AT$$
$$A \to a$$
$$B \to b$$

CYK Parsing

a	a	a	b	b
{A}	{A}	{A}	{B}	{B}
'aa'	'aa'	'ab'	'ba'	

2nd row: To derive substrings of length 2

$$S \to AB$$
$$S \to XB$$
$$T \to AB$$
$$T \to XB$$
$$X \to AT$$
$$A \to a$$
$$B \to b$$

CYK Parsing

a	a	a	b	b
{A}	{A}	{A}	{B}	{B}
'aa'	'aa'	'ab'	'ba'	

2nd row: To derive substrings of length 2

$$S \to AB$$
$$S \to XB$$
$$T \to AB$$
$$T \to XB$$
$$X \to AT$$
$$A \to a$$
$$B \to b$$

CYK Parsing

a	a	a	b	b
{A}	{A}	{A}	{B}	{B}
‘aa’	‘aa’	‘ab’	‘ba’	

2nd row: To derive substrings of length 2

$$\boxed{S \rightarrow AB}$$
$$S \rightarrow XB$$
$$\boxed{T \rightarrow AB}$$
$$T \rightarrow XB$$
$$X \rightarrow AT$$
$$A \rightarrow a$$
$$B \rightarrow b$$

CYK Parsing

a	a	a	b	b
{A}	{A}	{A}	{B}	{B}

‘aa’ ‘aa’ ‘ab’ ‘ba’

2^{nd} row: To derive substrings of length 2

$$S \to AB$$
$$S \to XB$$
$$T \to AB$$
$$T \to XB$$
$$X \to AT$$
$$A \to a$$
$$B \to b$$

CYK Parsing

a	a	a	b	b
{A}	{A}	{A}	{B}	{B}
∅	∅	{S, T}	∅	

$$S \to AB$$
$$S \to XB$$
$$T \to AB$$
$$T \to XB$$
$$X \to AT$$
$$A \to a$$
$$B \to b$$

CYK Parsing

a	a	a	b	b
{A}	{A}	{A}	{B}	{B}
∅	∅	{S, T}	∅	
'aaa'	'aab'	'abb'		

3rd row: To derive substrings of length 3

$$S \to AB$$
$$S \to XB$$
$$T \to AB$$
$$T \to XB$$
$$X \to AT$$
$$A \to a$$
$$B \to b$$

CYK Parsing

	a	a	a	b	b
length 1	{A}	{A}	{A}	{B}	{B}
length 2	∅	∅	{S, T}	∅	
	'aaa'	'aab'	'abb'		

3rd row: To derive substrings of length 3

$$S \to AB$$
$$S \to XB$$
$$T \to AB$$
$$T \to XB$$
$$X \to AT$$
$$A \to a$$
$$B \to b$$

CYK Parsing

	a	a	a	b	b
length 1	{A}	{A}	{A}	{B}	{B}
length 2	∅	∅	{S, T}	∅	
	'aaa'	'aab'	'abb'		

3rd row: To derive substrings of length 3

$$S \to AB$$
$$S \to XB$$
$$T \to AB$$
$$T \to XB$$
$$X \to AT$$
$$A \to a$$
$$B \to b$$

CYK Parsing

	a	a	a	b	b
length 1	{A}	{A}	{A}	{B}	{B}
length 2	∅	∅	{S, T}	∅	
	'aaa'	'aab'	'abb'		

3rd row: To derive substrings of length 3

$$S \to AB$$
$$S \to XB$$
$$T \to AB$$
$$T \to XB$$
$$\boxed{X \to AT}$$
$$A \to a$$
$$B \to b$$

CYK Parsing

	a	a	a	b	b
length 1	{A}	{A}	{A}	{B}	{B}
length 2	∅	∅	{S, T}	∅	
	‘aaa’	‘aab’	‘abb’		

3rd row: To derive substrings of length 3

$$S \to AB$$
$$S \to XB$$
$$T \to AB$$
$$T \to XB$$
$$X \to AT$$
$$A \to a$$
$$B \to b$$

CYK Parsing

a	a	a	b	b
{A}	{A}	{A}	{B}	{B}
∅	∅	{S, T}	∅	
∅	{X}	∅		

$S \to AB$

$S \to XB$

$T \to AB$

$T \to XB$

$X \to AT$

$A \to a$

$B \to b$

CYK Parsing

a	a	a	b	b
{A}	{A}	{A}	{B}	{B}
∅	∅	{S, T}	∅	
∅	{X}	∅		
∅	{S, T}			

$$S \to AB$$
$$\boxed{S \to XB}$$
$$T \to AB$$
$$\boxed{T \to XB}$$
$$X \to AT$$
$$A \to a$$
$$B \to b$$

CYK Parsing

a	a	a	b	b
{A}	{A}	{A}	{B}	{B}
∅	∅	{S, T}	∅	
∅	{X}	∅		
∅	{S, T}			
{X}				

$$S \to AB$$
$$S \to XB$$
$$T \to AB$$
$$T \to XB$$
$$\boxed{X \to AT}$$
$$A \to a$$
$$B \to b$$

CYK Parsing

a	a	a	b	b
{A}	{A}	{A}	{B}	{B}
Ø	Ø	{S, T}	Ø	
Ø	{X}	Ø		
Ø	{S, T}			
{X}				

$S \to AB$

$S \to XB$

$T \to AB$

$T \to XB$

$\boxed{X \to AT}$

$A \to a$

$B \to b$

The given grammar does not generate 'aaabb' because root of parse tree obtained is X and not the start symbol **S.**

Example 3

A very heavy orange book

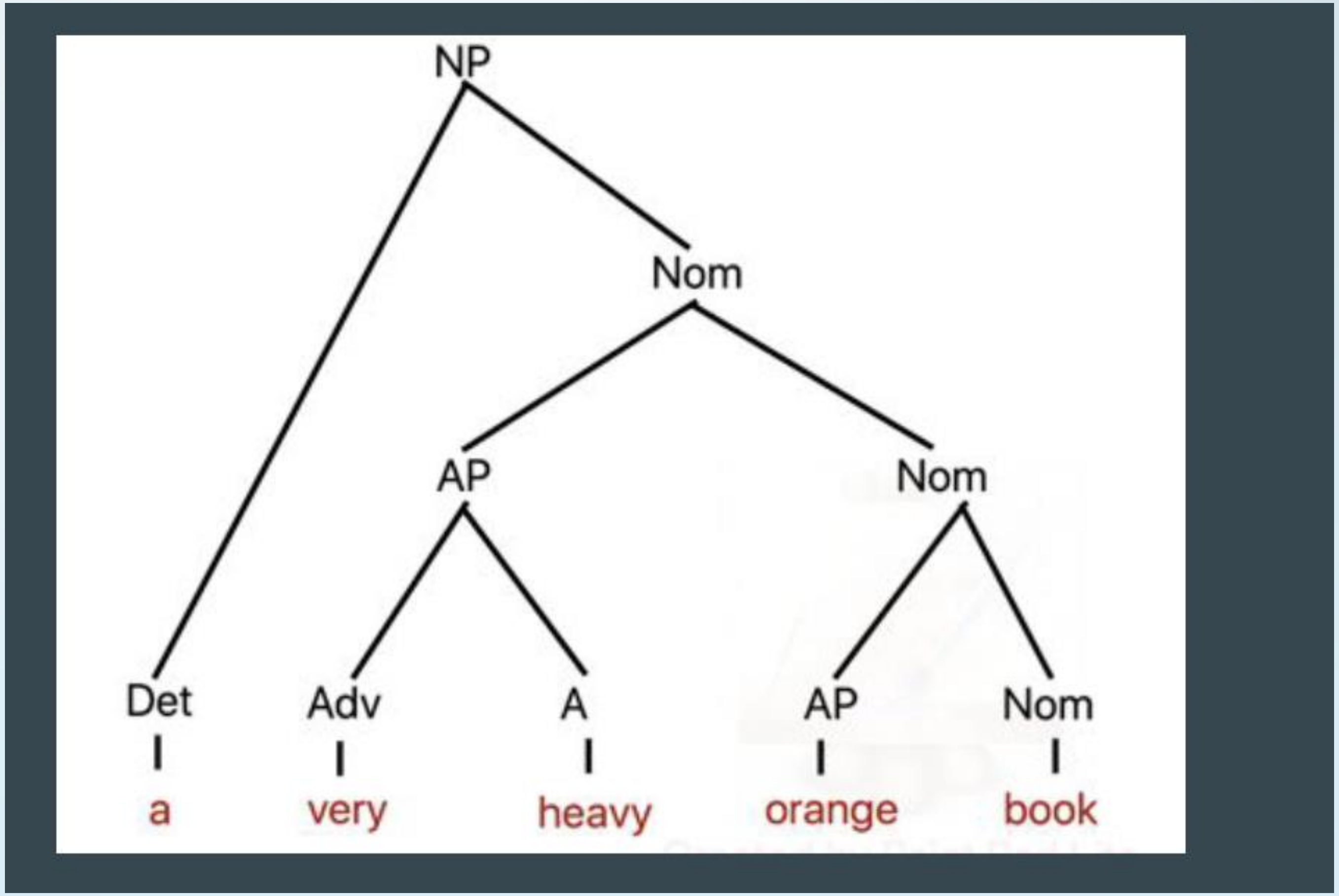
NP	→	Det Nom
Nom	→	AP Nom
AP	→	Adv A
Det	→	a | an
Adv	→	very | extremely
AP	→	heavy | orange | tall
A	→	heavy | orange | tall | muscular
Nom	→	book | orange | man

- A very heavy orange book

A	very	heavy	orange	book
Det	Adv	A, AP	AP, A, Nom	Nom
ϕ	AP	Nom	Nom	
ϕ	Nom	Nom		
ϕ	Nom			
NP				

NP	$\rightarrow$	Det Nom				
Nom	$\rightarrow$	AP Nom				
AP	$\rightarrow$	Adv A				
Det	$\rightarrow$	a | an				
Adv	$\rightarrow$	very | extremely				
AP	$\rightarrow$	heavy | orange | tall				
A	$\rightarrow$	heavy | orange | tall | muscular				
Nom	$\rightarrow$	book | orange | man				

One possible parse tree



HW

- A very tall muscular man
- A very tall extremely muscular man

Example 4

I eat sushi with chopsticks with you

Is this an ambiguous sentence?

1	S	$\longrightarrow$	NP	VP
2	NP	$\longrightarrow$	NP	PP
3	NP	$\longrightarrow$	sushi	
4	NP	$\longrightarrow$	I	
5	NP	$\longrightarrow$	chopsticks	
6	NP	$\longrightarrow$	you	
7	VP	$\longrightarrow$	VP	PP
8	VP	$\longrightarrow$	Verb	NP
9	Verb	$\longrightarrow$	eat	
10	PP	$\longrightarrow$	Prep	NP
11	Prep	$\longrightarrow$	with	

Example 3

I eat sushi with chopsticks with you

I	eat	sushi	with	chopsticks	with	you
NP	Verb	NP	Prep	NP	Prep	NP
ϕ	VP	ϕ	PP	ϕ	PP	
S	ϕ	NP	ϕ	NP		
ϕ	VP_1 VP_2	ϕ	PP			
S_1, S_2	ϕ	NP → NP PP				
ϕ	VP_1, VP_2 VP_3					
S_1, S_2, S_3						

1	S	→	NP	VP
2	NP	→	NP	PP
3	NP	→	sushi	
4	NP	→	I	
5	NP	→	chopsticks	
6	NP	→	you	
7	VP	→	VP	PP
8	VP	→	Verb	NP
9	Verb	→	eat	
10	PP	→	Prep	NP
11	Prep	→	with	

Are there multiple parse trees?

Can you draw them?

CYK Algorithm

```
function CKY-PARSE(words, grammar) returns table

  for j ← from 1 to LENGTH(words) do
      for all {A | A → words[j] ∈ grammar}
          table[j − 1, j] ← table[j − 1, j] ∪ A
      for i ← from j − 2 down to 0 do
          for k ← i + 1 to j − 1 do
              for all {A | A → BC ∈ grammar and B ∈ table[i,k] and C ∈ table[k, j]}
                  table[i,j] ← table[i,j] ∪ A
```

Figure 13.5 The CKY algorithm.

Time complexity O(n^3) Space complexity O(n^2)

References

- J&M Book “Speech and Language Processing” – Chapter 13
- Monisha’s slides on CYK Parsing from CS6370: Jan-May 2020
- https://www.geeksforgeeks.org/converting-context-free-grammar-chomsky-normal-form/
- https://courses.engr.illinois.edu/cs447/fa2018/Slides/Lecture09.pdf