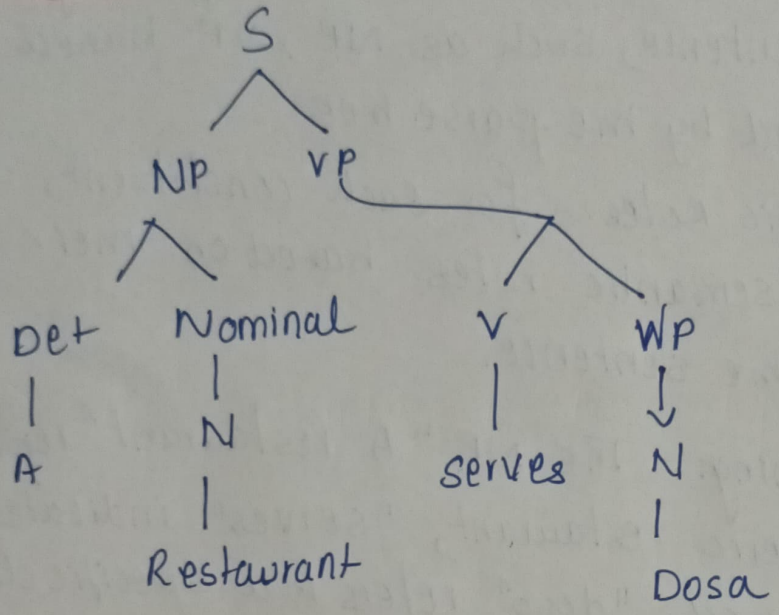
Module:-2

① "A restaurant serves dosa" → perform semantic analysis and show the semantic ~~representation~~ interpretations of the constituents.



→ Semantic Interpretations:- (SI)

. 'A restaurant':- This is a NP with a determiner "A" and nominal "Restaurant". The SI of this NP is that it refers to an unspecified or generic restaurant, w/o specifying any particular one.

. serves → This is a verb (V) that represents the action of serving. The verb "serves" indicates the activity or function performed by the subject.

. dosa → This is a NP that represents the direct object of the verb "serves". The SI of this NP is that it refers to a specific dish called "dosa".

→ Semantic Analysis Process

i) Syntactic Parsing:- start by analyzing the sentence's syntactic structure using the given parse tree. The

parse tree represents how the words are grouped together in phrases and how the phrases are related to each other.

ii) Identifying Constituents :- Next, we identify the grammatical constituents in the sentence, such as NP, VP based on the structure provided by the parse tree

iii) Assigning Semantic Roles :- for each constituent, we assign appropriate semantic roles based on their context & meaning in the sentence.

iv) Interpreting Meaning :- The NP "A restaurant" represents an unspecified or generic restaurant, "serves" indicates the action being performed, "dosa" refers to a specific dish

v) Semantic Representation :- We map the syntactic structure of its corresponding semantic representation, capturing the relationships b/w the constituents and their meanings. In this case, the semantic representation shows that a generic ~~representat~~ restaurant performs the action of serving the specific dish "dosa".

② Algorithm for Minimum edit Distance

Input: Two strings, X and Y

Output: The min edit distance b/w X & Y

```
m ← length(X)
n ← length(Y)
for i = 0 to m do
    dist[i, 0] ← i
```

```
for j=0 to n do
    dist[0,j] ← j
for i=0 to m do
    for j=0 to n do
        dist[i,j] = min {dist[i-1,j] + insert_cost,
                         dist[i-1,j-1] + subst_cost(Xi, Yj),
                         dist[i,j-1] + delet_cost}
```

→ The min edit distance b/w tutor and tumour

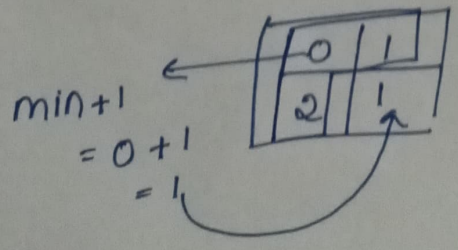
n \ m	#	t	u	m	o	u	r
#	0	1	2	3	4	5	6
t	1	0	1	2	3	4	5
u	2	1	0	1	2	3	4
t	3	2	1	1	2	3	4
o	4	3	2	2	1	2	3
r	5	4	3	3	2	2	2

Subs insertion
t u m o r
t u t o r

∴ 2

(6) if m = n → write me same no as diagonal

if m ≠ n → min of 3 boxes + 1



(2) Use Top down parsing for the sentence "The dogs cried" using the grammar

(Art → Article)

S → NP VP

NP → Art N

NP → Art Adj N

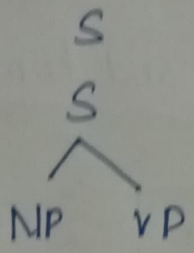
VP → V

VP → V NP

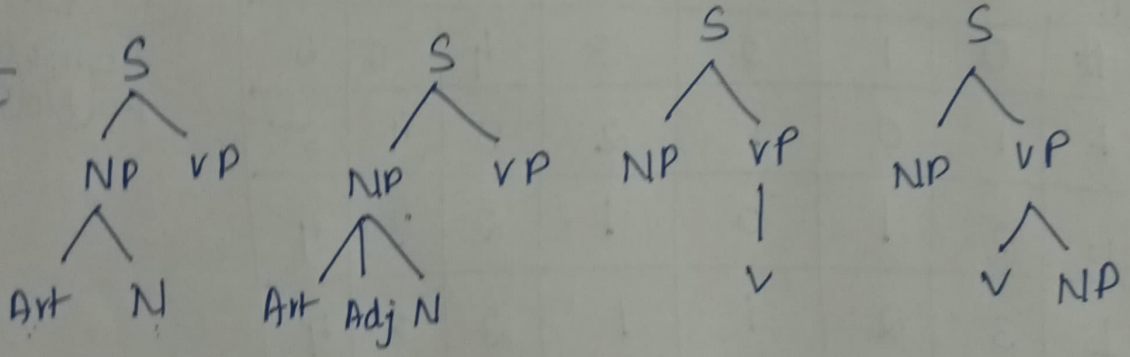
level I

S

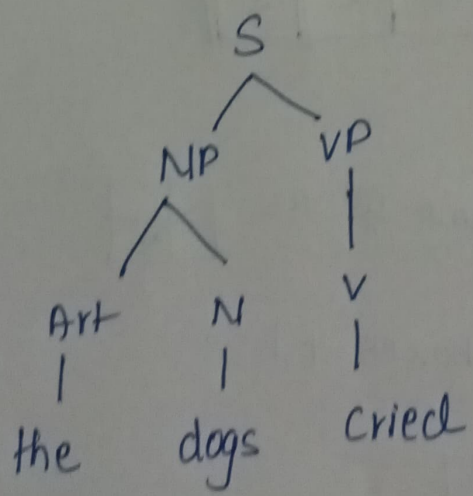
level II



level III



→ The correct parse tree



(4) Top down, depth-first, left to right parse tree for the given sentence "The angry bear chased the frightened little squirrel"

Grammar rules :- S → NP VP

NP → Det Nom

VP → V NP

Nom → Adj Nom/N

Det → the

Adj → little | angry | frightened

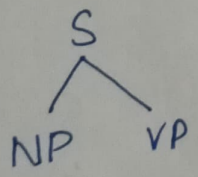
N → squirrel | bear

V → chased

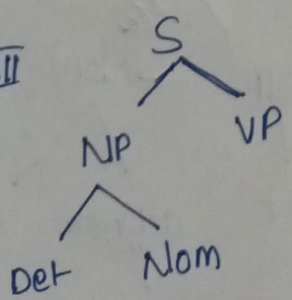
i) Top-down parsing:- Starts at the top of the sentence and recursively breaks down the sentence into its constituent parts

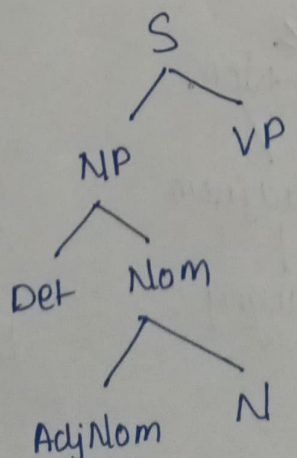
Level I S

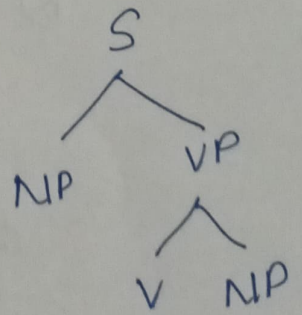
Level II



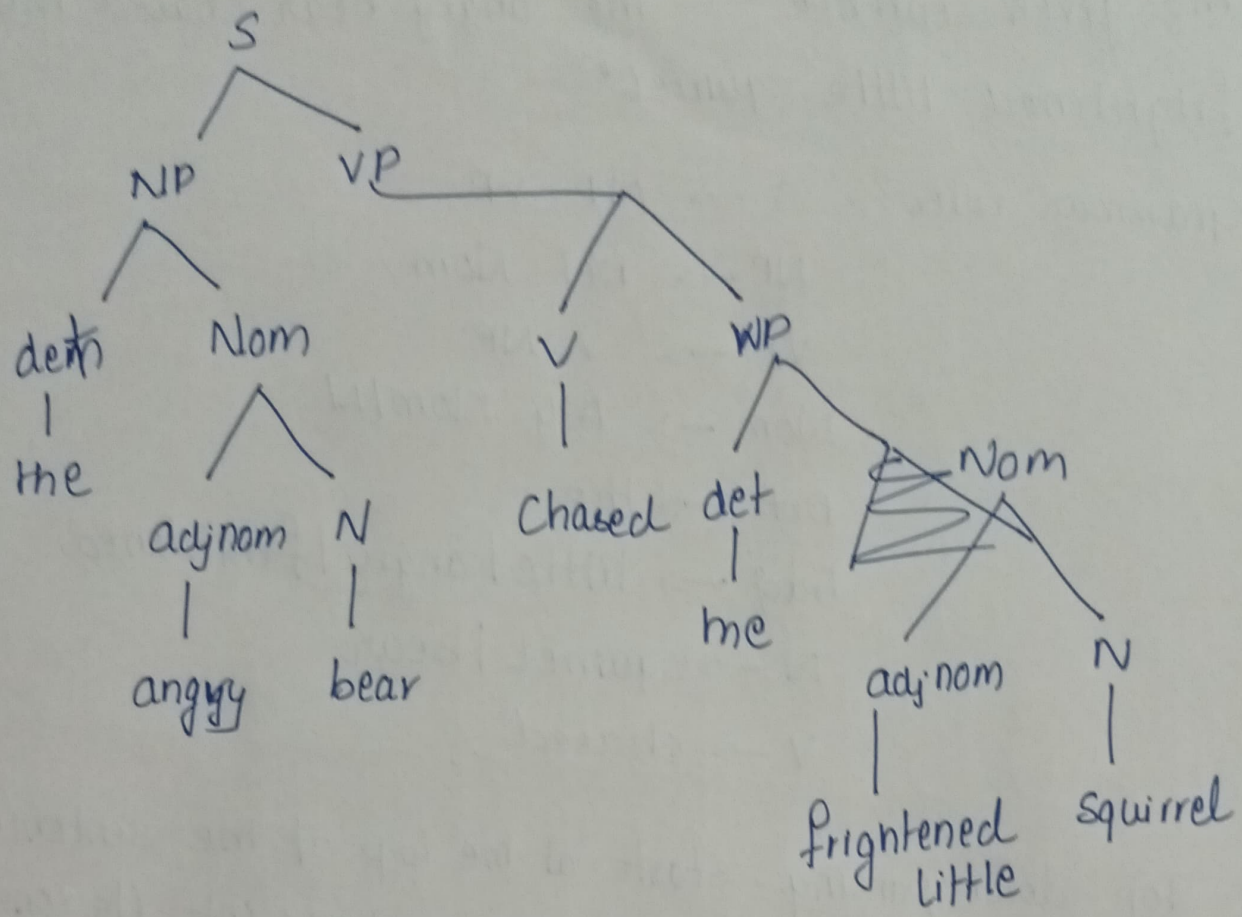
Level III







-> The correct parse tree



ii) Depth-first:- Similar to top-down parsing, but it only explores one branch of the search space at a time. If the branch has dead-end, the parser backtracks and explores another branch.

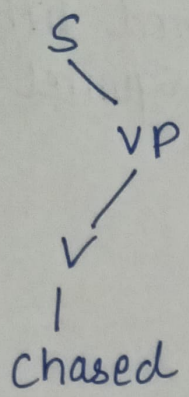
a) S
NP
det
the

b) S
NP — Nom
det
the
adjnom
angry

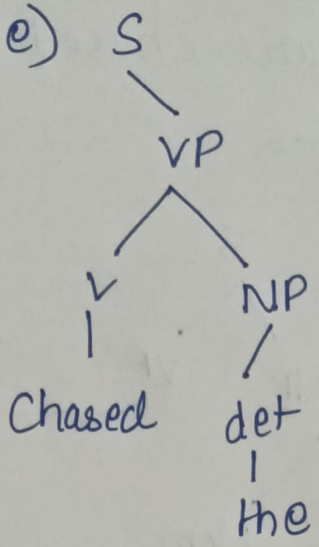
c) S
NP — Nom
det
the
adjnom
angry
N
bear

-> This is for left side

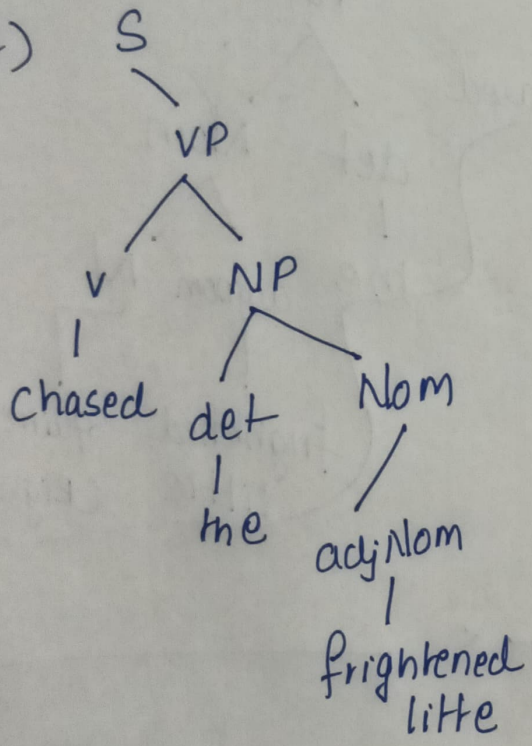
d)



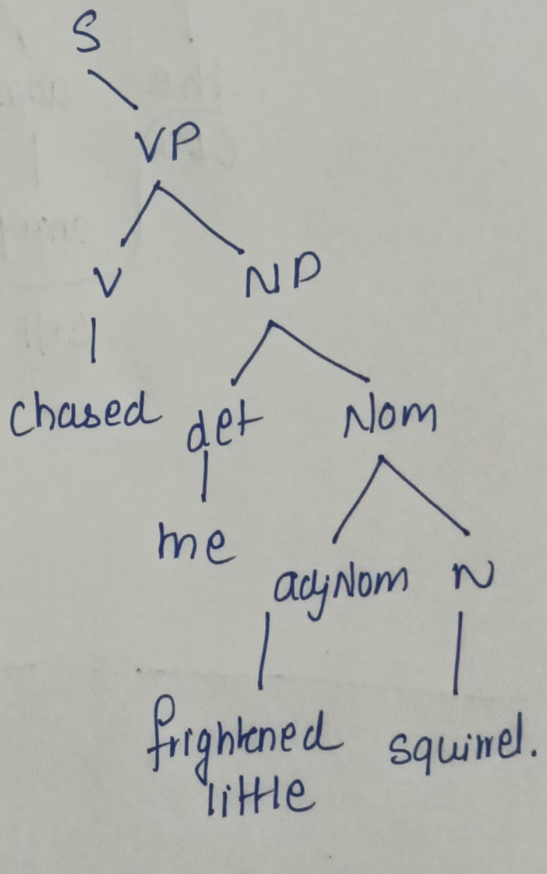
e)



f)



g)



—> Combine all the (g) steps we get our parse same like top-down, but here it only explores one branch of the search space at a time.

iii) left-to-right:- Starts at the beginning of the sentence and parses the sentence one word at a time.

The angry bear chased me frightened little squirrel