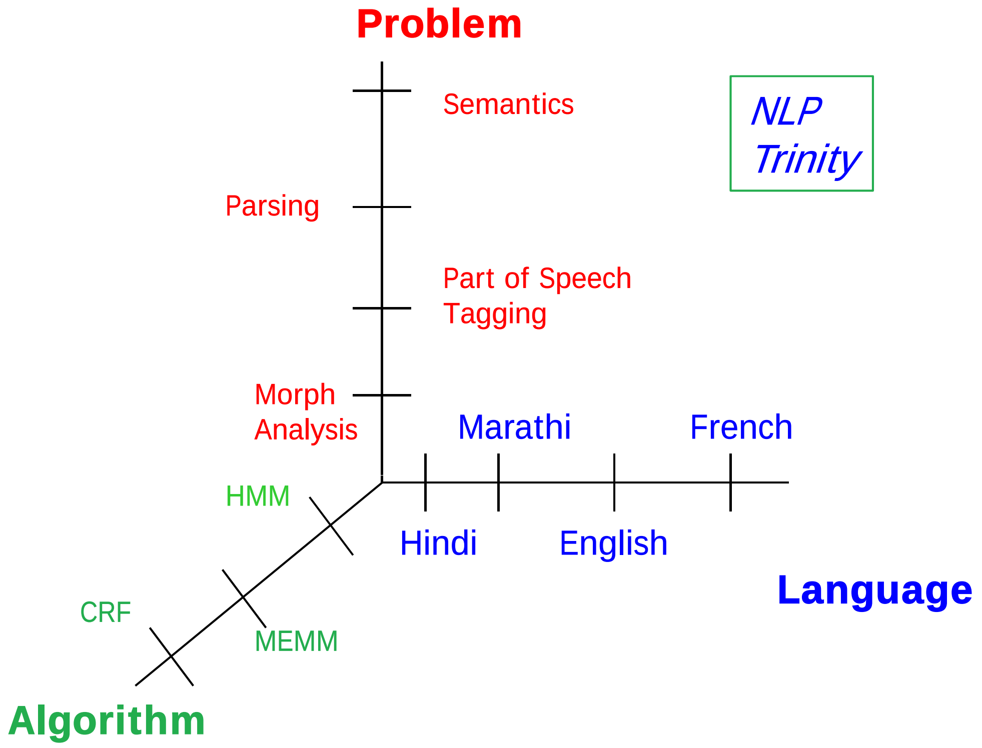
CS626: NLP, Speech and the Web

Pushpak Bhattacharyya
CSE Dept.,
IIT Bombay

Lecture 12,13: Parsing

23rd and 27th August, 2012

NLP Trinity



Problem

Semantics

Parsing

Part of Speech Tagging

Morph Analysis

Language

Hindi

Marathi

English

French

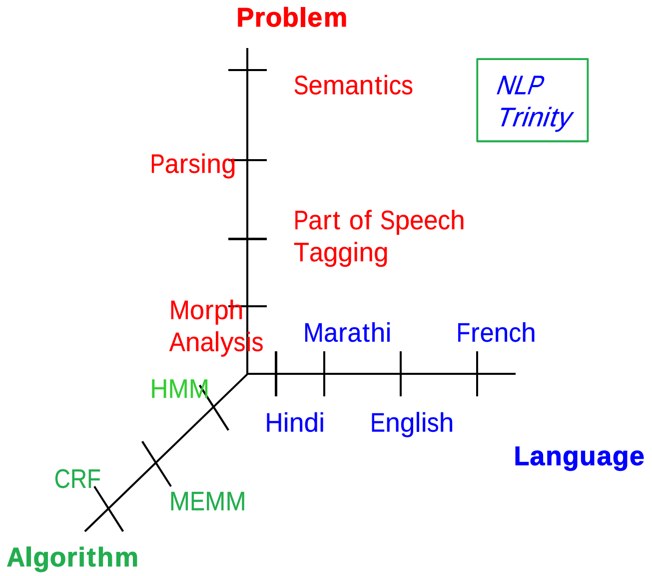
Algorithm

HMM

MEMM

CRF

Need for parsing



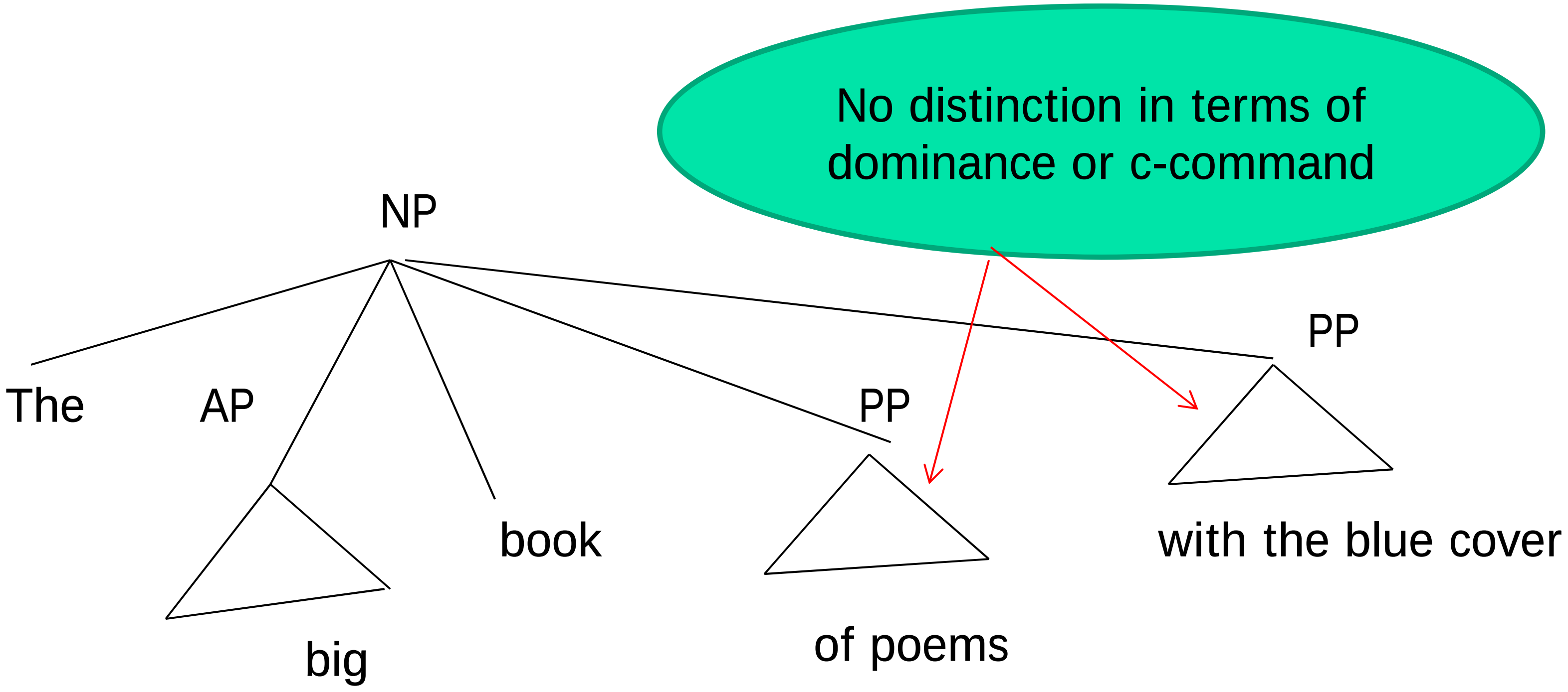
Sentences are linear structures

But there is a *hierarchy- a* ***tree-*** hidden behind the linear structure

There are constituents and branches

(Note: things to come back to- POS tagset; MEMM and CRF for POS)

PPs are at the same level: *flat with respect to the head word "book"*



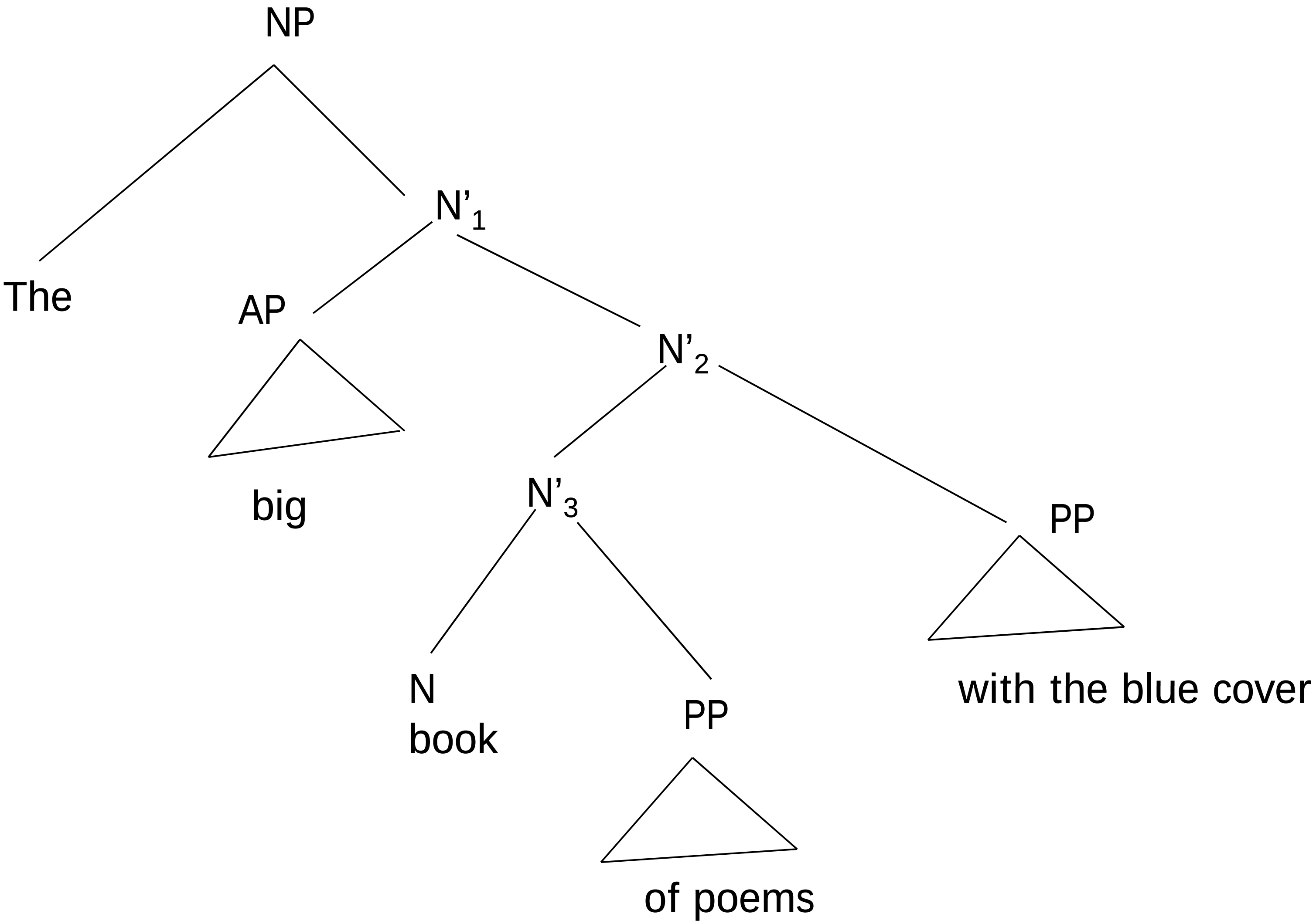
No distinction in terms of dominance or c-command

[*The big book of poems with the Blue cover*] is on the table.

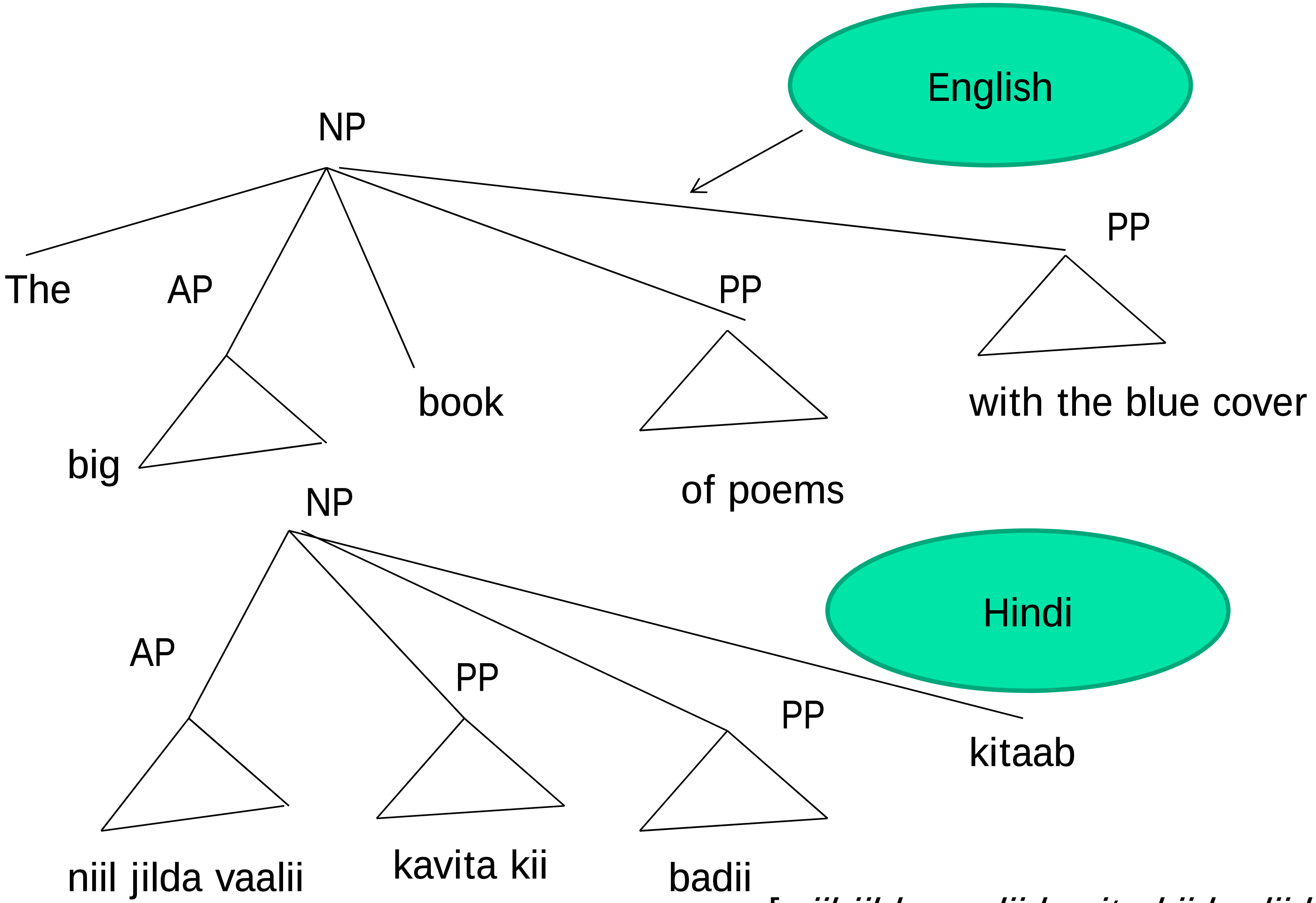
“Constituency test of Replacement” runs into problems

- One-replacement:
  - *I bought the big [book of poems with the blue cover] not the small [one]*
  - *One-replacement* targets *book of poems with the blue cover*
- Another one-replacement:
  - *I bought the big [book of poems] with the blue cover not the small [one] with the red cover*
  - *One-replacement* targets *book of poems*

More deeply embedded structure

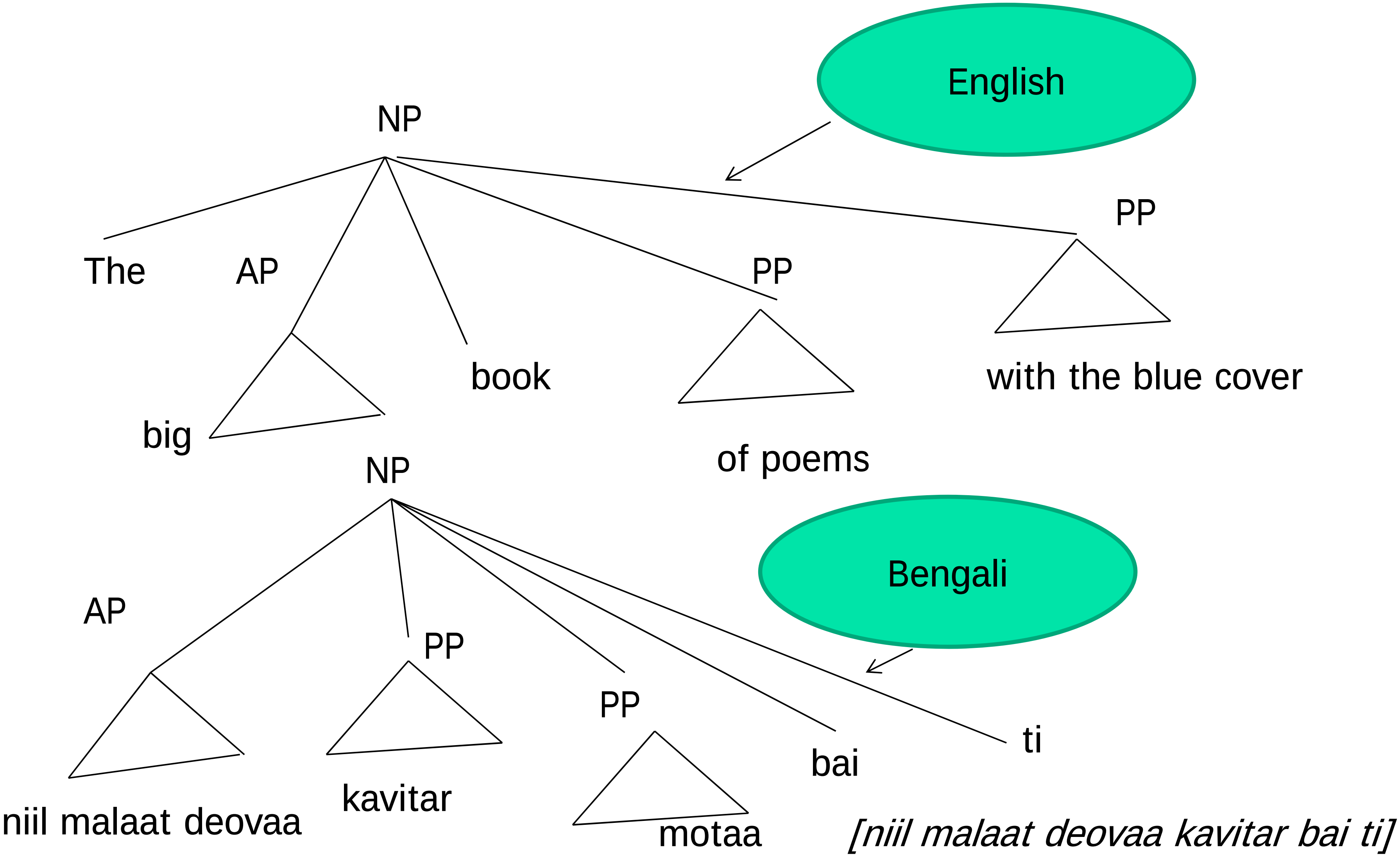


Other languages



[*niil jilda vaalii kavita kii badii kitaab*]

Other languages: contd



English

NP

The

AP

big

book

PP

of poems

PP

with the blue cover

Bengali

NP

AP

niil malaat deovaa

PP

kavitar

PP

motaa

bai

ti

[niil malaat deovaa kavitar bai ti]

Structure Dependency: more cases

- **Interrogative Inversion**

(1) John will solve the problem.
Will John solve the problem?

	Declarative	*Interrogative*
(2) a.	Susan must leave.	Must Susan leave?
b.	Harry can swim.	Can Harry swim?
c.	Mary has read the book.	Has Mary read the book?
d.	Bill is sleeping.	Is Bill sleeping?

……………………………………………………………..

*The section, "**Structure dependency a case study**" here is adopted from a talk given by Howard Lasnik (2003) in Delhi university.*

Credit: next few slides are from Dr. Bibhuti Mahapatra's lecture
On linguistics at CFILT, 2011

Interrogative inversion
Structure Independent (1st attempt)

(3) **Interrogative inversion process**

Beginning with a declarative, invert the first and second words to construct an interrogative.

Declarative	Interrogative
(4) a. The woman must leave.	*Woman the must leave?
b. A sailor can swim.	*Sailor a can swim?
c. No boy has read the book.	*Boy no has read the book?
d. My friend is sleeping.	*Friend my is sleeping?

Interrogative inversion
correct pairings

- Compare the incorrect pairings in (4) with the correct pairings in (5):

Declarative	**Interrogative**
(5) a. The woman must leave.	Must the woman leave?
b. A sailor can swim.	Can a sailor swim?
c. No boy has read the book.	Has no boy read the book?
d. My friend is sleeping.	Is my friend sleeping?

Interrogative inversion
Structure Independent (2nd attempt)

(6) Interrogative inversion process:

- Beginning with a declarative, move the auxiliary verb to the front to construct an interrogative.

Declarative	Interrogative
(7) a. Bill could be sleeping.	*Be Bill could sleeping?
	Could Bill be sleeping?
b. Mary has been reading.	*Been Mary has reading?
	Has Mary been reading?
c. Susan should have left.	*Have Susan should left?
	Should Susan have left?

Structure independent (3rd attempt):

(8) Interrogative inversion process

- Beginning with a declarative, move the first auxiliary verb to the front to construct an interrogative.

	Declarative	Interrogative
(9) a.	The man who is here can swim.	*Is the man who here can swim?
b.	The boy who will play has left.	*Will the boy who play has left?

Structure Dependent Correct Pairings

- For the above examples, fronting the second auxiliary verb gives the correct form:

	Declarative	Interrogative
(10) a.	The man who is here can swim.	Can the man who is here swim?
b.	The boy who will play has left.	Has the boy who will play left?

Natural transformations are structure dependent

(11) Does the child acquiring English learn these properties?

(12) We are not dealing with a peculiarity of English. No known human language has a transformational process that would produce pairings like those in (4), (7) and (9), repeated below:

(4) a. The woman must leave. *Woman the must leave?

(7) a. Bill could be sleeping. *Be Bill could sleeping?

(9) a. The man who is here can swim. *Is the man who here can swim?

Interrogative inversion: some more complicated facts

(22) The man left.

(23) Mary sleeps.

- Sentences, e.g. (22)-(23), with no auxiliary at all do have interrogative counterparts, but ones that initially seem to fall under entirely different mechanisms.

	Declarative	Interrogative
(24)	a. Mary will sleep.	a`. Will Mary sleep?
	b. Mary sleeps.	b`. Does Mary sleep?

- Comparing (24a) and (24a`), we see just the familiar inversion alternation.
- But comparing (24b) and (24b`), instead we see a change in the form of the main verb (from ***sleeps*** to ***sleep***), and the addition of a form of the auxiliary verb *do* in the pre-subject position

Need for Abstract underlying structure.

- Implementation of the above insight requires a notion of abstract underlying structure.
- Apart from interrogative inversion process there are three other phenomena displaying the same abstract pattern; such as: *Negation, Emphasis* and *Verb phrase Ellipsis:*

NEGATION

(25)	John left	John didn’t leave.
	John should leave.	John shouldn’t leave.
	John has left.	John hasn’t left.
	John is leaving.	John isn’t leaving.

Emphasis and Verb Phrase Ellipsis

EMPHASIS

(26)	John left.	John **did** leave.
	John should leave.	John **should** leave.
	John has left.	John **has** left.
	John is leaving.	John **is** leaving.

VERB PHRASE ELLIPSIS

(27)	John left.	Mary did too.
	John should leave.	Mary should too.
	John has left.	Mary has too.
	John is leaving.	Mary is too.

An even more hidden cause

(28) a. She worked.

b. She works.

(29) a. They worked.

b. They work.

- In the present tense, except for the third person singular form, there is no apparent morpheme on the verb at all. The verb in (29b) is indistinguishable from the uninflected citation form.

Constituency Vs. Dependency

- What is more important?
  - Noun or Verb
  - Sanskrit Tradition (Dhatujamah - धातुजमाह

    i.e. everything is derived from verbal root)

Dependency Parsing

- Dependency approach is suitable for free word-order language

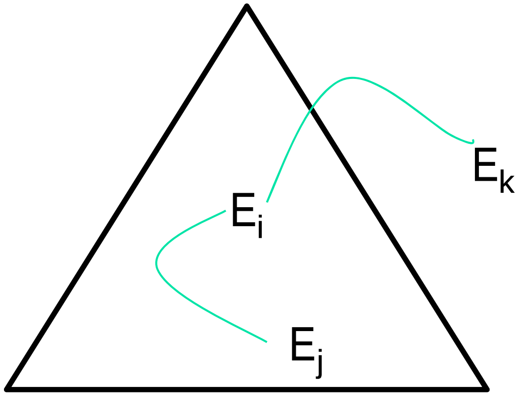
- Example : Hindi
  - राम ने शाम को देखा (Ram ne Shyam ko dekha)
  - शाम को राम ने देखा (Shyam ko Ram ne dekha)

- One step closer to **Semantics**

- Parsing Structural Ambiguity

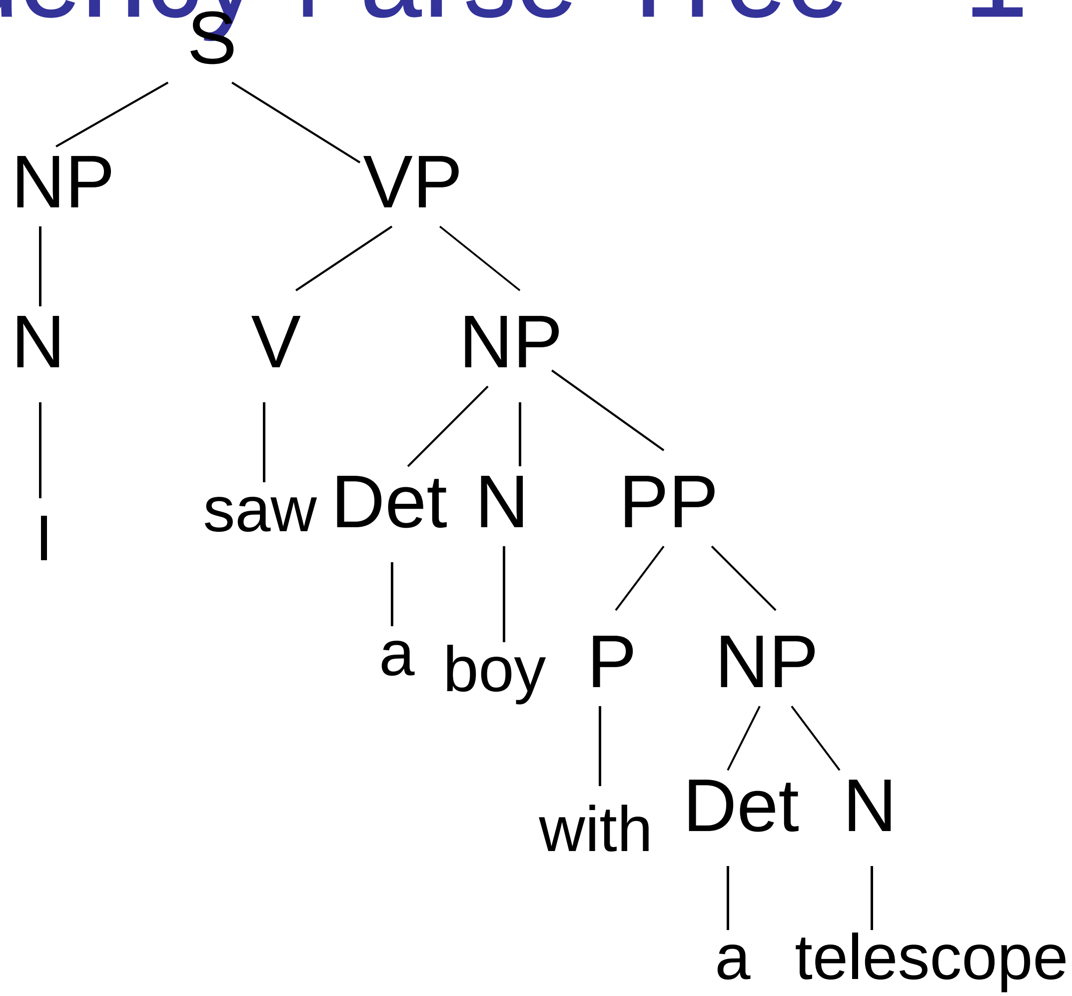
Parse Tree

- Within a sub-tree entities bind together than they do with entities outside the sub-tree

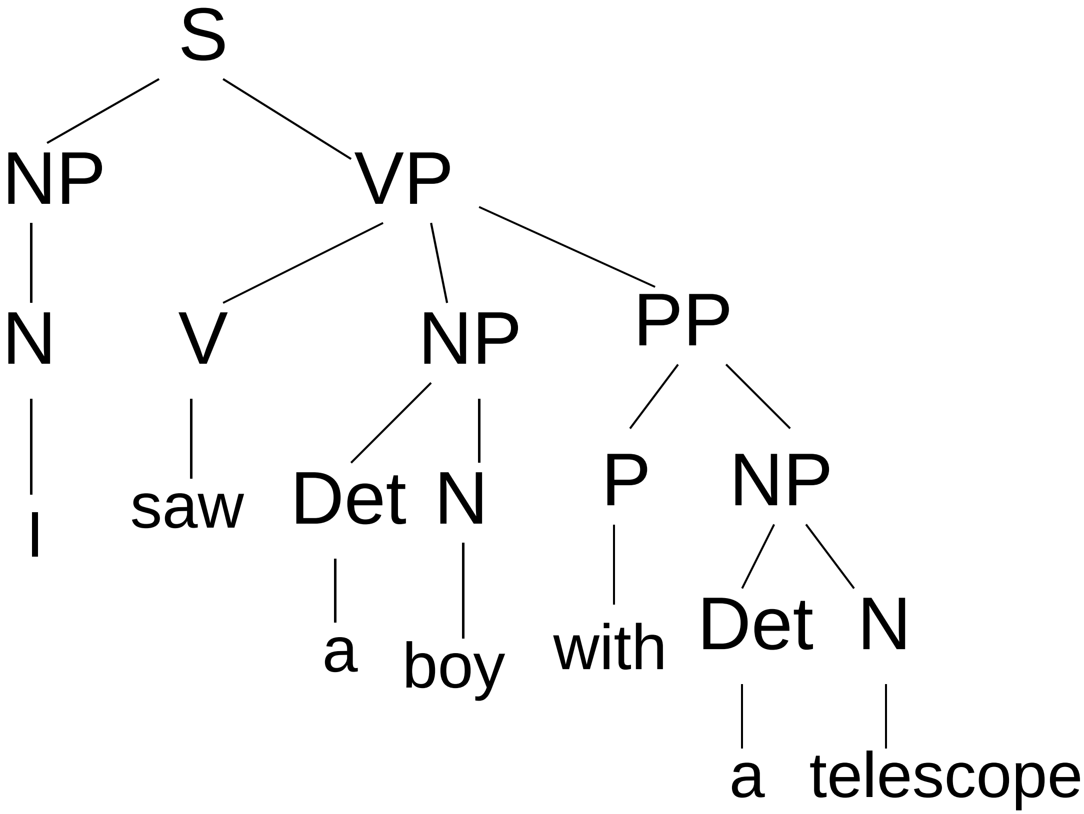


- Strength (E_i, E_j) > Strength (E_i, E_k)

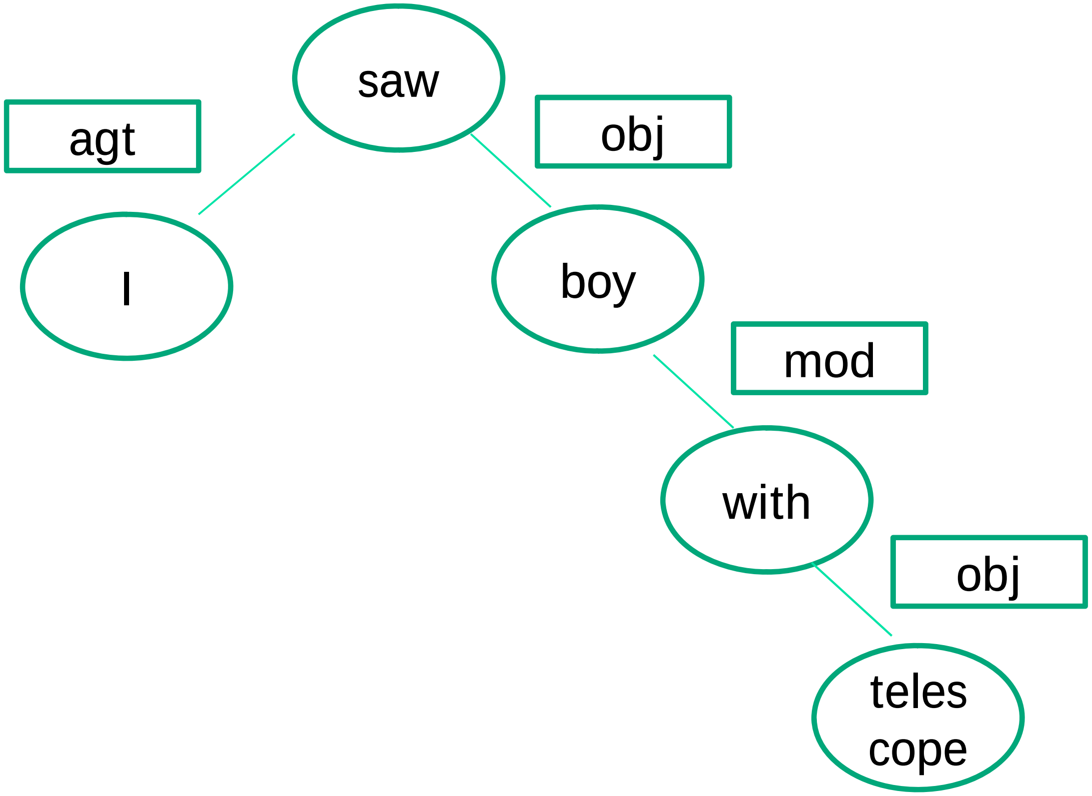
Constituency Parse Tree - 1



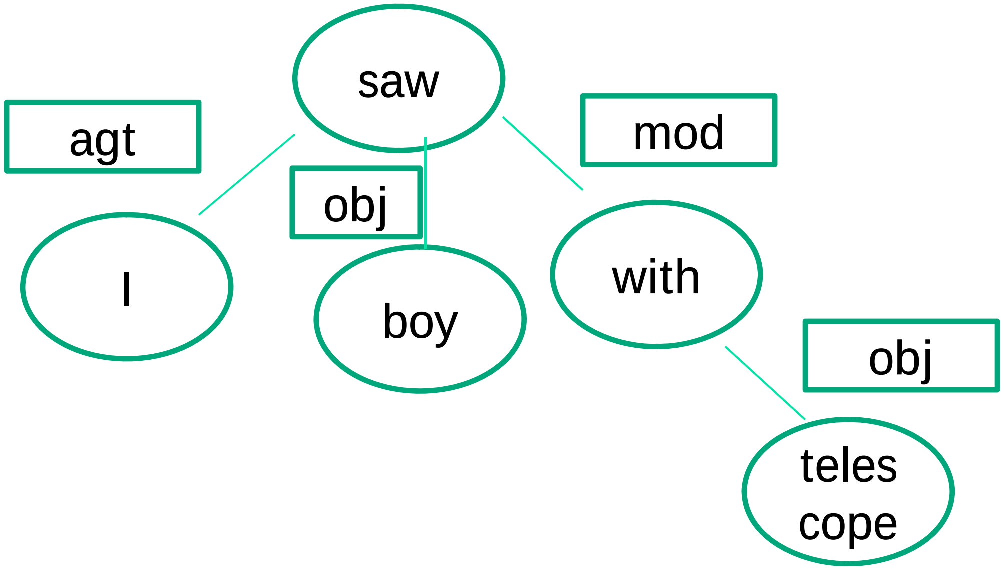
Constituency Parse Tree -2



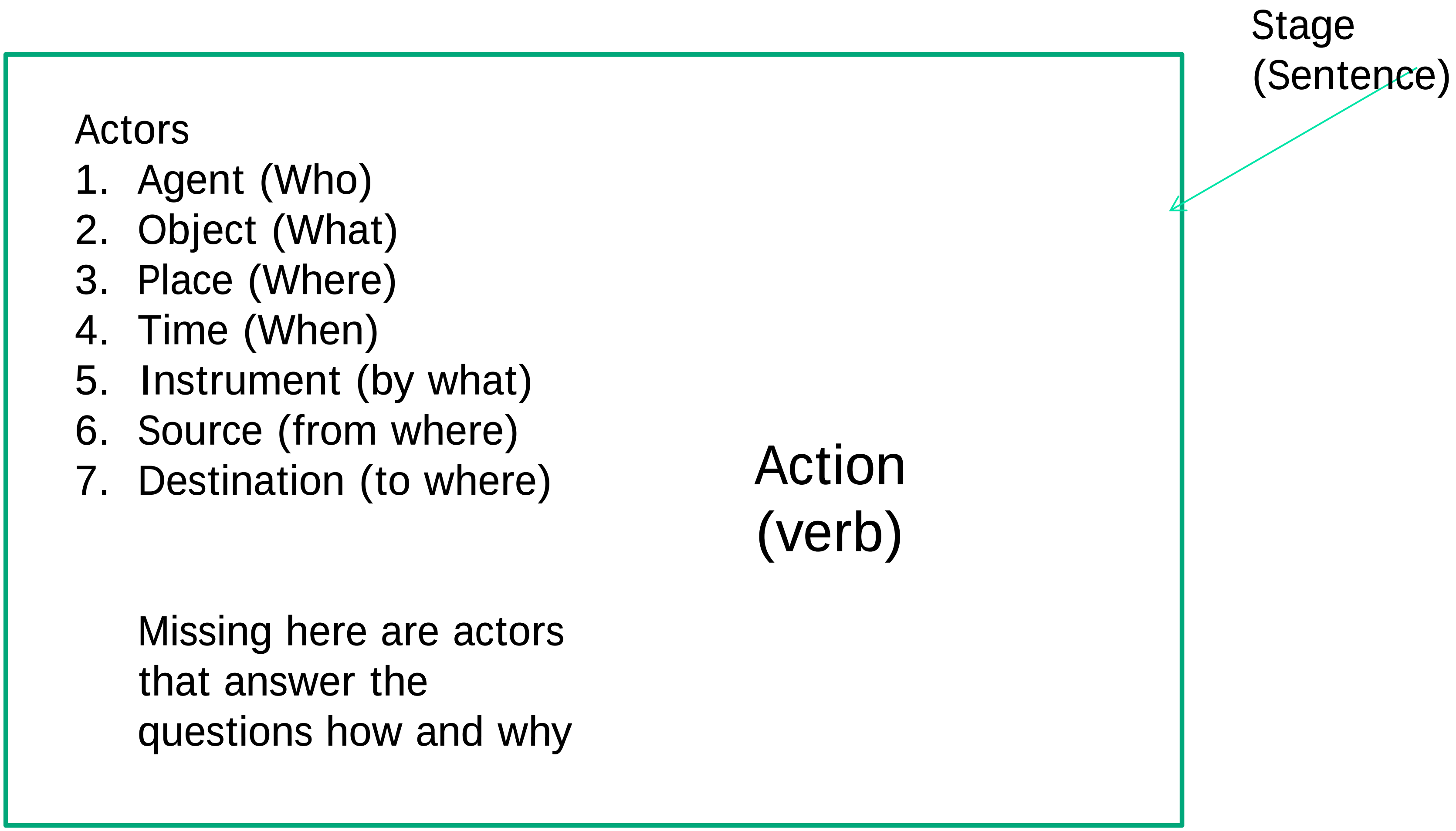
Dependency Parse Tree - 1



Dependency Parse Tree - 2



Verb centric view of Sentence



Stage
(Sentence)

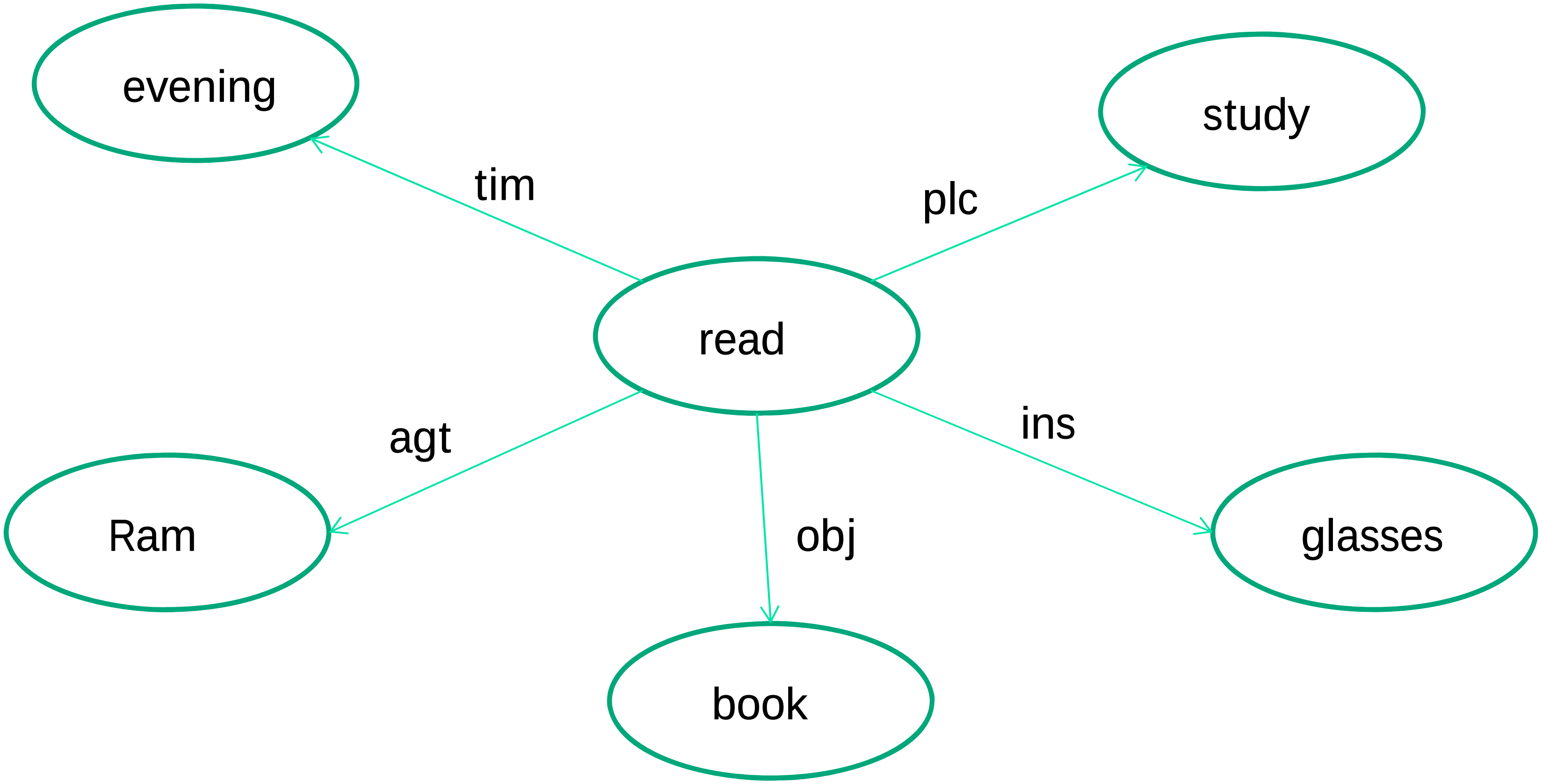
Actors

1. Agent (Who)
2. Object (What)
3. Place (Where)
4. Time (When)
5. Instrument (by what)
6. Source (from where)
7. Destination (to where)

Action
(verb)

Missing here are actors that answer the questions how and why

- Ram reads a book with his glasses in the evening in his study.



- The labels on the arcs are semantic roles and the task is Semantic Role Labeling.

Which is important?

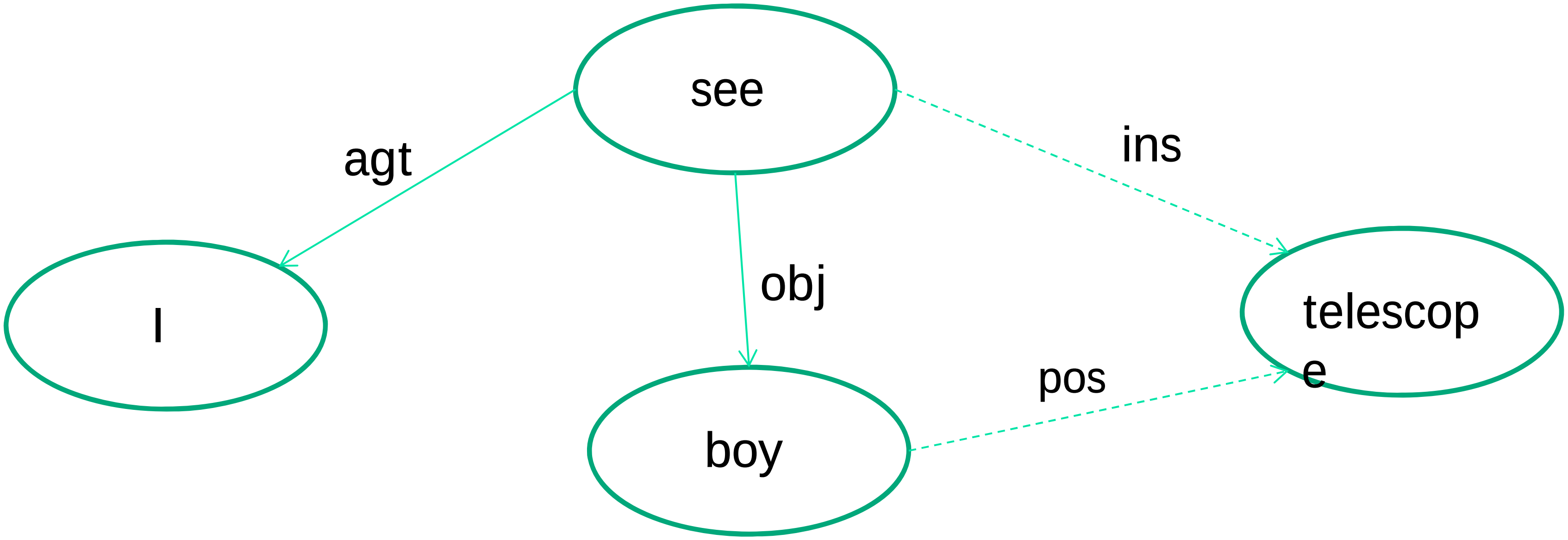
- Bill shot at the President
  - Emphasizes Bill – agent
- The president was shot at by Bill
  - Emphasizes the President – object
- The President was shot at
  - Emphasizes shooting – action
- कर्तरी प्रयोग - Agent is important
- कर्मणी प्रयोग – Object is important
- भावे प्रयोग – Action is important

Register

- Way of writing or speaking that is situation specific
- Friend to friend communication
  - Informal
- Application for leave
  - Formal communication

Dependency Parsing

- I saw the boy with a telescope.



Speech Acts

- Noun
  - Definite/Indefinite
- Verb
  - Tense
  - Number
  - Person

A note on Language Modeling

- Example sentence
  - *“ ^ The tortoise beat the hare in the race.”*

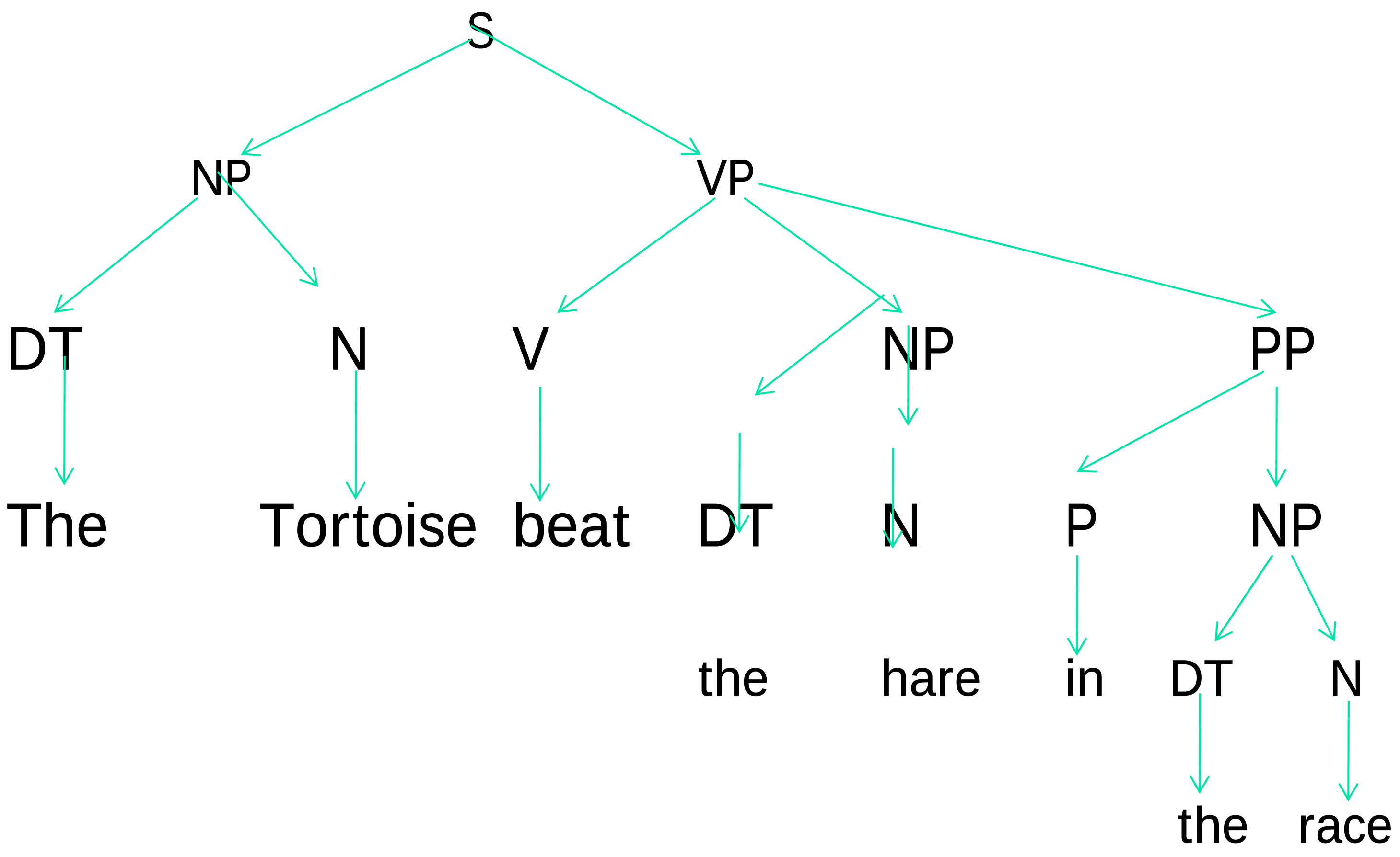
Guided by frequency

Guided by Language Knowledge

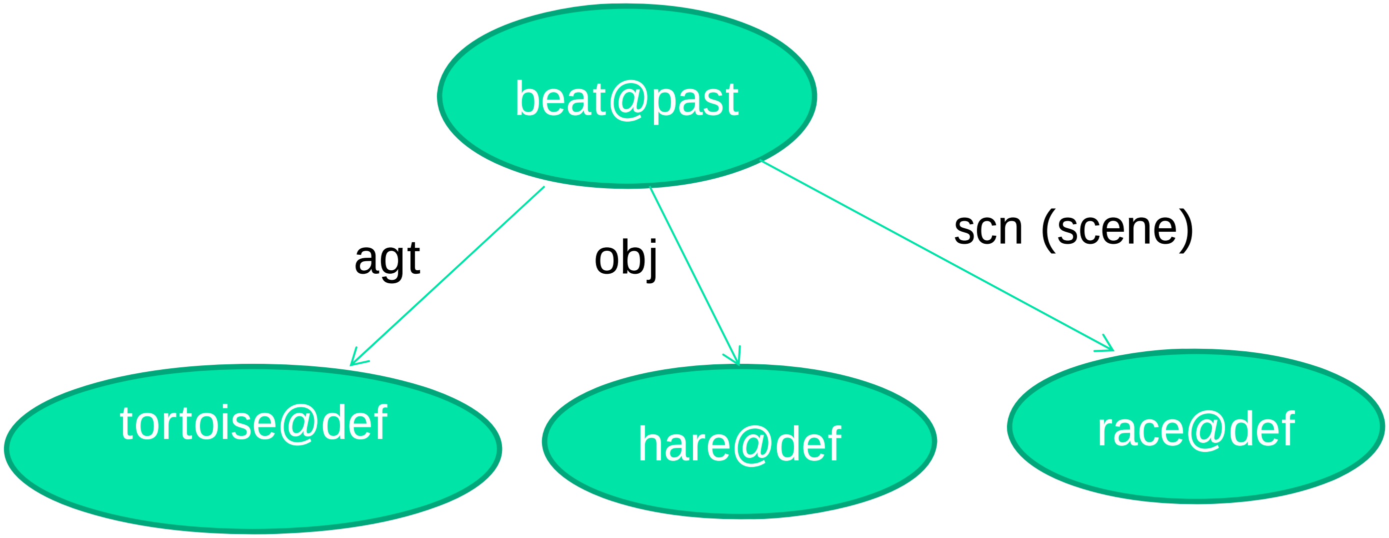
Guided by world Knowledge

N-gram (n=3)	**CFG**	**Probabilistic CFG**	**Dependency Grammar**	**Prob. DG**
^ the tortoise $5*10^{-3}$	S-> NP VP	S->NP VP 1.0	Semantic Roles *agt, obj, sen, etc.* Semantic Rules are always between *“Heads”*	Semantic Roles with probabilities
the tortoise beat $3*10^{-2}$	NP->DT N	NP->DT N 0.5		
tortoise beat the $7*10^{-5}$	VP->V NP PP	VP->V NP PP 0.4		
beat the hare $5*10^{-6}$	PP-> P NP	PP-> P NP 1.0		

Parse Tree


S
NP
VP
DT
N
V
NP
PP
The
Tortoise
beat
DT
N
P
NP
the
hare
in
DT
N
the
race

UNL Expression



Purpose of LM

- Prediction of next word (Speech Processing)
- Language Identification (for same script)
- Belongingness check (parsing)
- P(NP->DT N) means what is the probability that the ‘YIELD’ of the non terminal NP is DT N

Computation of Parsing

Parsing

- Essentially a Yes/No problem (belongingness)
- Parse Tree is a side effect
- 2 general ways – Expand from grammar or Resolve through data

Language Representation

Intrinsic Representation

Grammar for STR:

$\mathbf{S} \rightarrow a\mathbf{S}b \mid \epsilon$

-Compact representation
-Less Kolomogorov Complexity

Extrinsic Representation

STR: $\{\epsilon, ab, aabb, \dots, a^nb^n\}$

-Enumerated Representation

- Points to consider:
  - Should POS Tagging precede parsing?
  - Is POS Tagging necessary for Parsing?

→ POS Tagging increases implementation efficiency

Data	*People*	*laugh*
Lexicon	People- Noun, Verb	laugh- Noun, Verb
Grammar		

- Going back again and again to the lexicon isn’t required when POS Tagging has been done before Parsing

- Two issues are at the crux of parsing:
  - Ambiguity in Data
  - Ambiguity in Grammar

- Parsing Algorithms:
  - Top-Down Parsing
    - Predictive Parsing, Expectation Driven Parsing, Theory Driven Parsing, Grammar Driven Parsing
    - Suffers from Left-recursion
  - Bottom-Up Parsing
    - Data Driven parsing
    - Ambiguity on POS Tags can lead to useless steps while parsing

  - Chart Parsing

- Example sentence:

  $_1$ *People* $_2$ *laugh* $_3$ *loudly* $_4$

  - Multiple parse trees possible for ambiguous sentences
    - The man saw a boy with a telescope
  - Partial parses are also important
    - Text entailment
    - Question Answering
      - People laugh loudly → Who laughs? People laugh

Grammar and Parsing Algorithms

A simplified grammar

- S $\rightarrow$ NP VP
- NP $\rightarrow$ DT N | N
- VP $\rightarrow$ V ADV | V

Example Sentence

People laugh

1 2 3

These are positions

Lexicon:

People - N, V

Laugh - N, V

This indicate that both Noun and Verb is possible for the word "People"

Top-Down Parsing

State	Backup State	Action
1. ((S) 1)	-	-
2. ((NP VP)1)	-	-
3a. ((DT N VP)1)	((N VP) 1)	-
3b. ((N VP)1)	-	-
4. ((VP)2)	-	Consume “People”
5a. ((V ADV)2)	((V)2)	-
6. ((ADV)3)	((V)2)	Consume “laugh”
5b. ((V)2)	-	-
6. ((.)3)	-	Consume “laugh”

Position of input pointer

Termination Condition : All inputs over. No symbols remaining.

Note: Input symbols can be pushed back.

Discussion for Top-Down Parsing

- This kind of searching is goal driven.
- Gives importance to textual precedence (rule precedence).
- No regard for data, a priori (useless expansions made).

Bottom-Up Parsing

Some conventions:

N_{12}

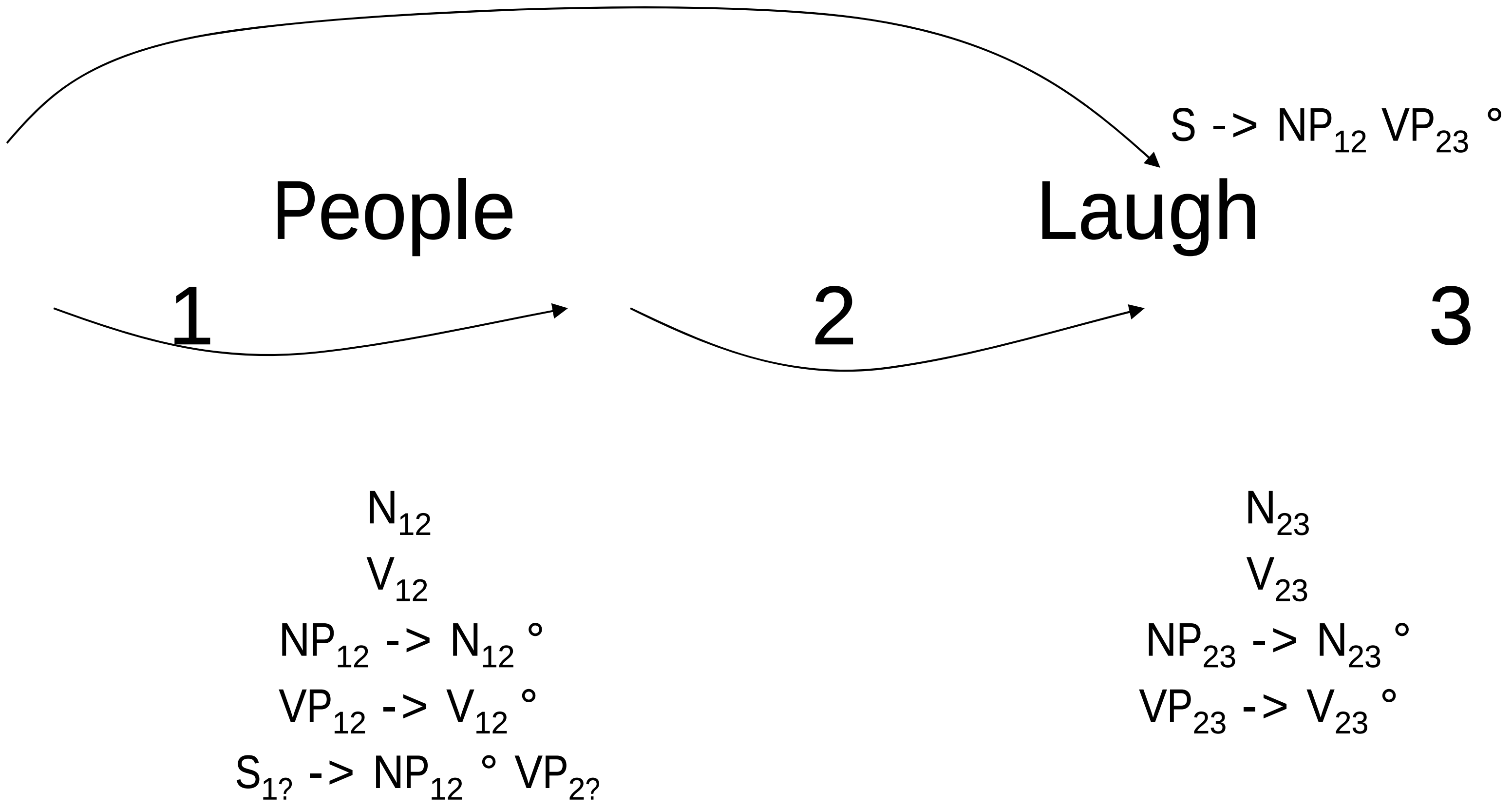
Represents positions

$S_{1?} -> NP_{12} \circ VP_{2?}$

End position unknown

Work on the LHS done, while the work on RHS remaining

Bottom-Up Parsing (pictorial representation)



$S \text{ -> } NP_{12}\ VP_{23}$ °

People Laugh

1 2 3

N_{12}

V_{12}

$NP_{12} \text{ -> } N_{12}$ °

$VP_{12} \text{ -> } V_{12}$ °

$S_{1?} \text{ -> } NP_{12}$ ° $VP_{2?}$

N_{23}

V_{23}

$NP_{23} \text{ -> } N_{23}$ °

$VP_{23} \text{ -> } V_{23}$ °

Problem with Top-Down Parsing

- Left Recursion
  - Suppose you have A-> AB rule.

    Then we will have the expansion as follows:
    - ((A)K) -> ((AB)K) -> ((ABB)K) ....

Combining top-down and bottom-up strategies

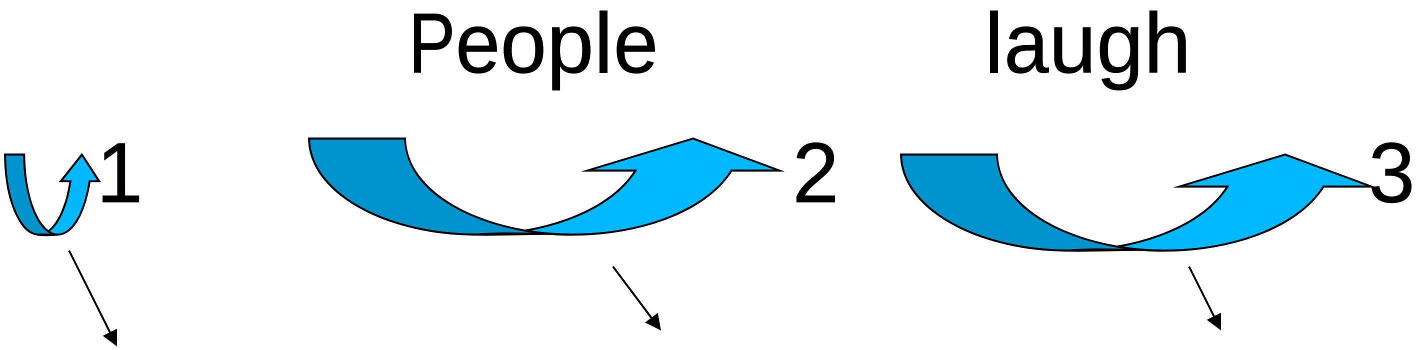
Top-Down Bottom-Up Chart Parsing

- Combines advantages of top-down & bottom-up parsing.
- Does not work in case of left recursion.
  - *e.g.* – “People laugh”
    - People – noun, verb
    - Laugh – noun, verb
  - Grammar – S → NP VP

    NP → DT N | N

    VP → V ADV | V

Transitive Closure



S →•NP VP

NP →•DT N

NP →•N

NP →N•

S → NP•VP

VP →•V ADV

VP →•V

VP → V •

S → NP VP •

success

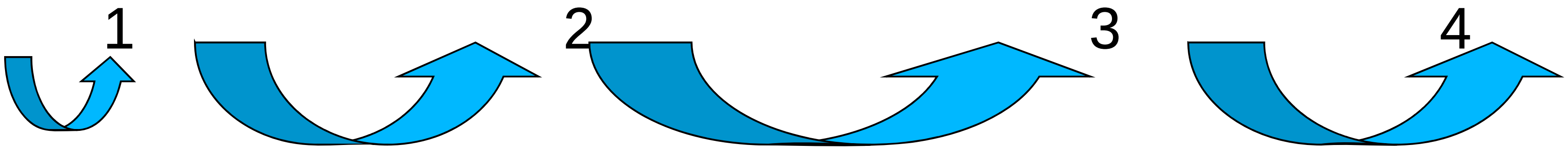
Arcs in Parsing

- Each arc represents a chart which records
  - Completed work (left of •)
  - Expected work (right of •)

Example

People laugh loudly

1 2 3 4



S →• NP VP NP →• DT N NP →• N	NP → N• S → NP•VP VP → •V ADV VP → •V	VP → V• VP → V•ADV S → NP VP•	VP → V ADV• S → NP VP•

Dealing With Structural Ambiguity

- Multiple parses for a sentence
  - The man saw the boy with a telescope.
  - The man saw the mountain with a telescope.
  - The man saw the boy with the ponytail.

At the level of syntax, all these sentences are ambiguous. But semantics can disambiguate 2nd & 3rd sentence.

Prepositional Phrase (PP) Attachment Problem

$V – NP_1 – P – NP_2$

(Here P means preposition)

NP_2 attaches to NP_1 ?

or NP_2 attaches to V ?

Parse Trees for a Structurally Ambiguous Sentence

Let the grammar be –

$S \rightarrow NP\ VP$

$NP \rightarrow DT\ N \mid DT\ N\ PP$

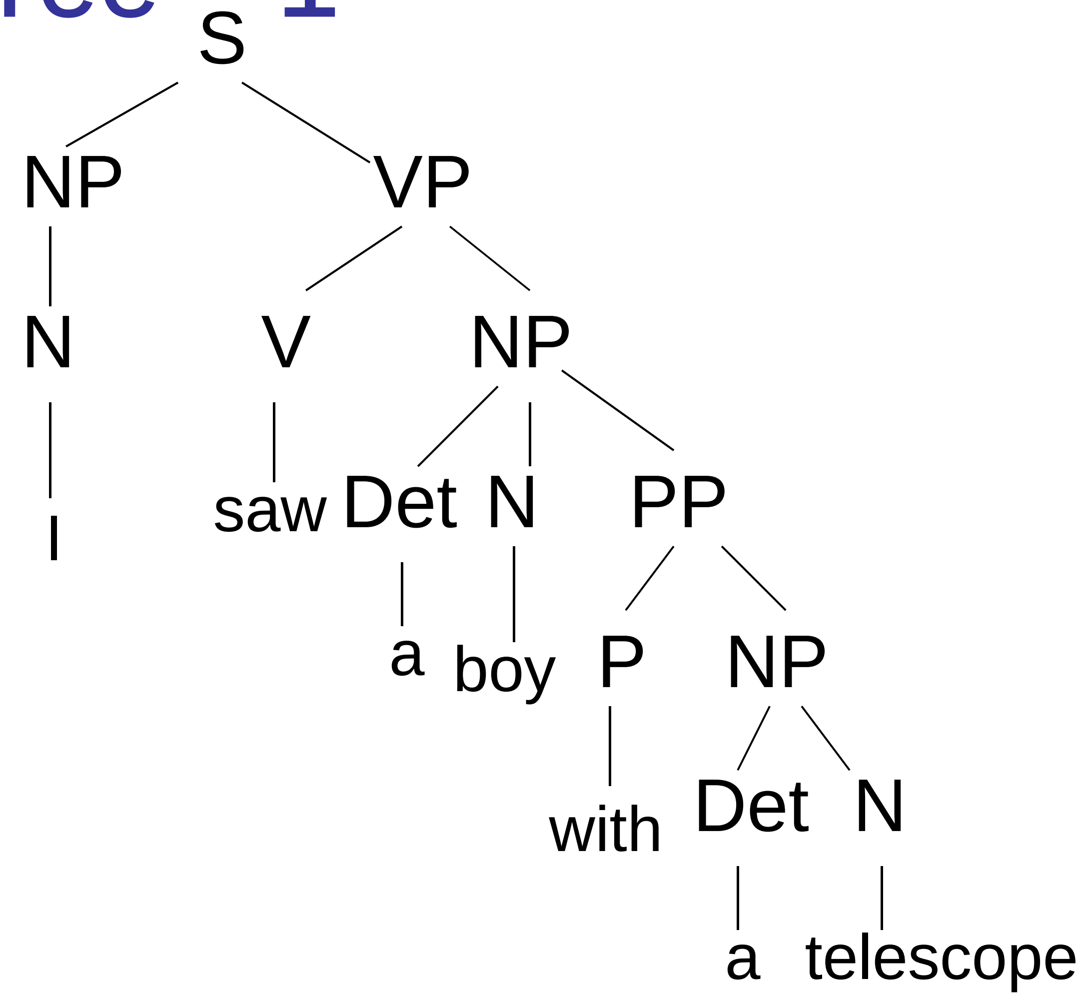
$PP \rightarrow P\ NP$

$VP \rightarrow V\ NP\ PP \mid V\ NP$

For the sentence,

“I saw a boy with a telescope”

Parse Tree - 1



Parse Tree -2