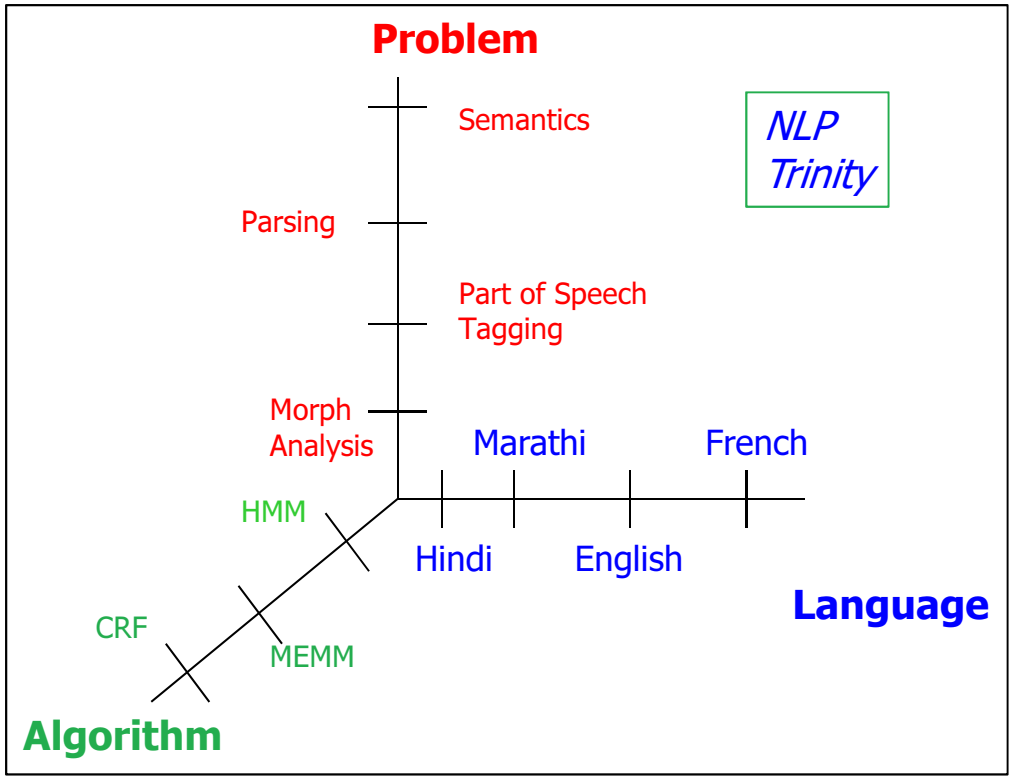
CS626: NLP, Speech and the Web

Pushpak Bhattacharyya
CSE Dept.,
IIT Bombay

Lecture 14: Parsing Algorithms

30th August, 2012

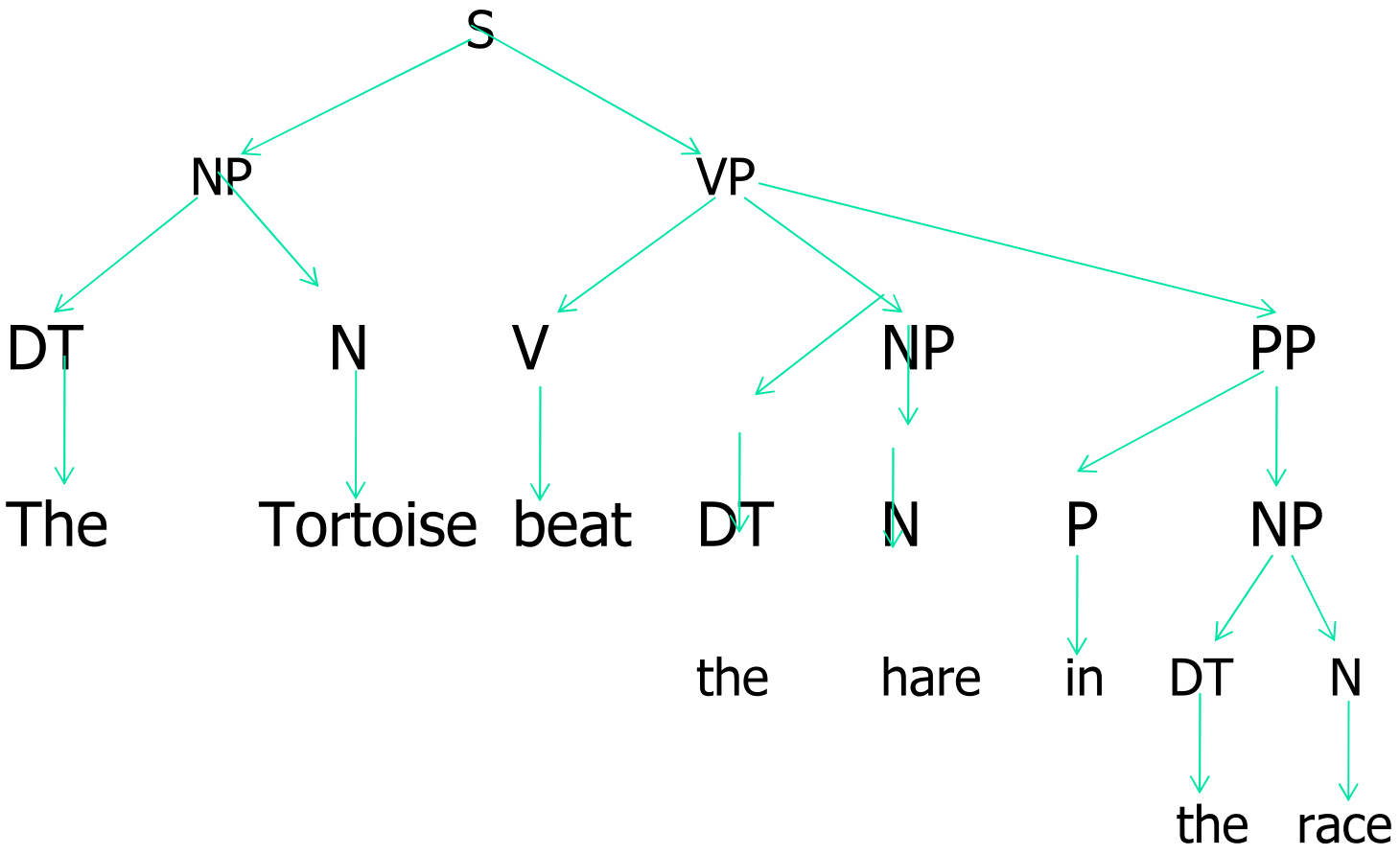

Problem
Semantics
NLP
Trinity
Parsing
Part of Speech
Tagging
Morph
Analysis
Marathi
French
HMM
Hindi
English
Language
CRF
MEMM
Algorithm

A note on Language Modeling

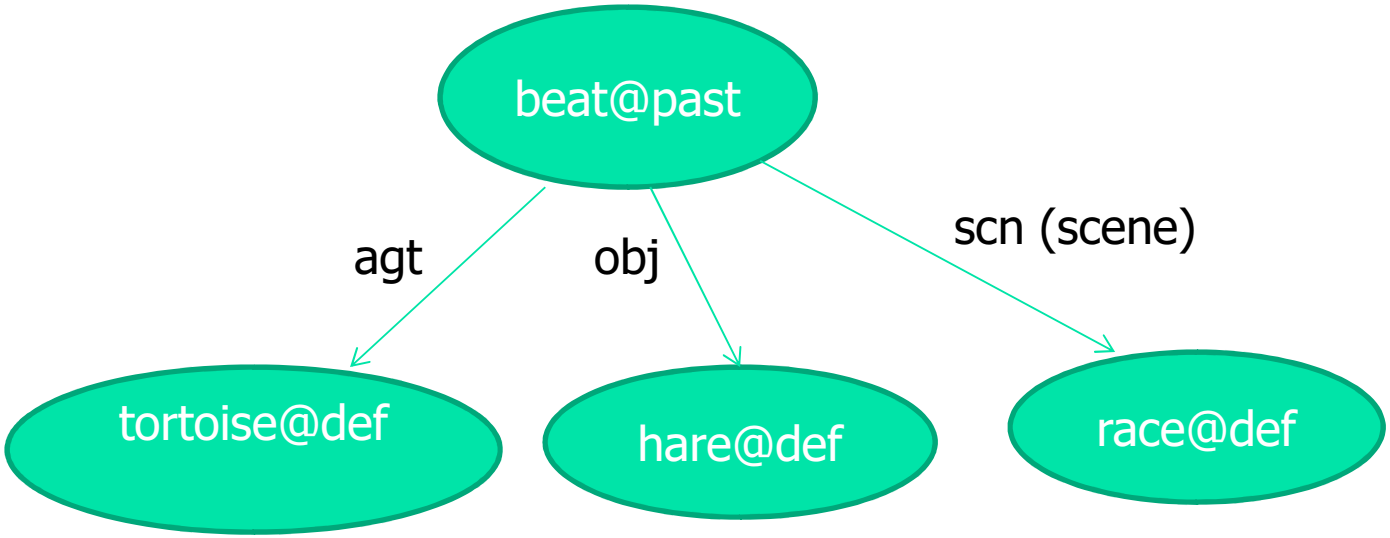
- Example sentence
  - " ^ *The tortoise beat the hare in the race.*"

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

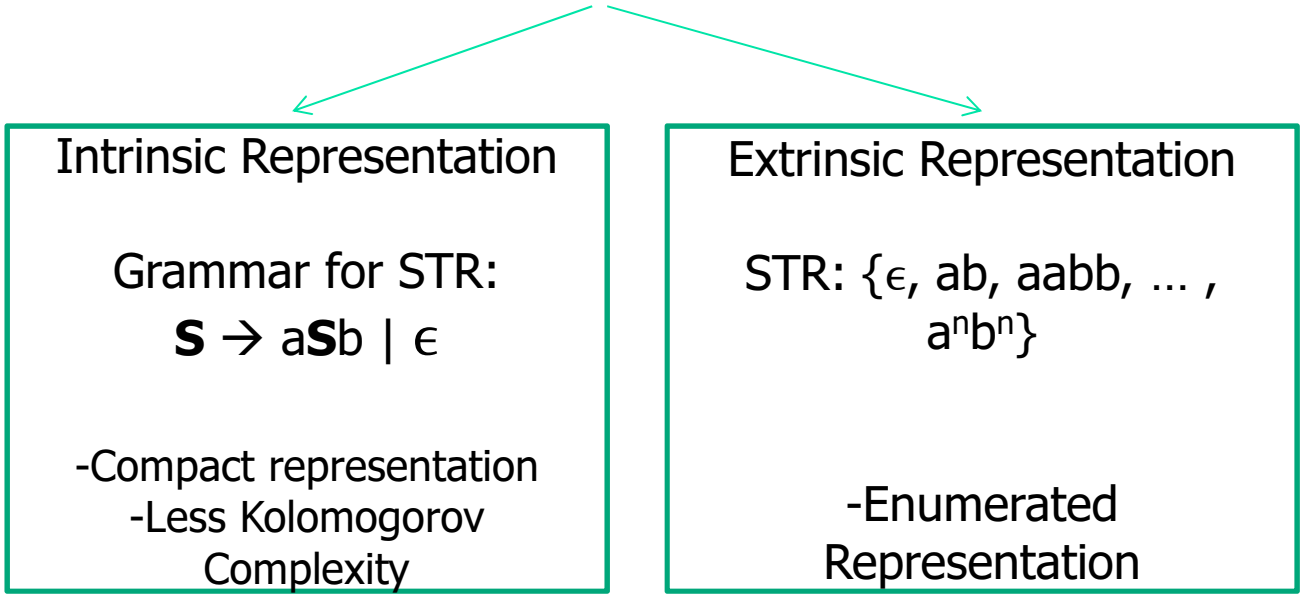
- Prediction of next word (Speech Processing)
- Language Identification (for same script)
- Belongingness check (parsing)
- P(NP->DT N) means what is the probability that the 'YIELD' of the non terminal NP is DT N

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



- 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:

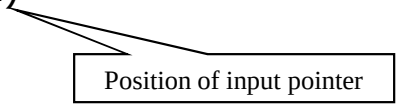
People - N, V

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"



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}

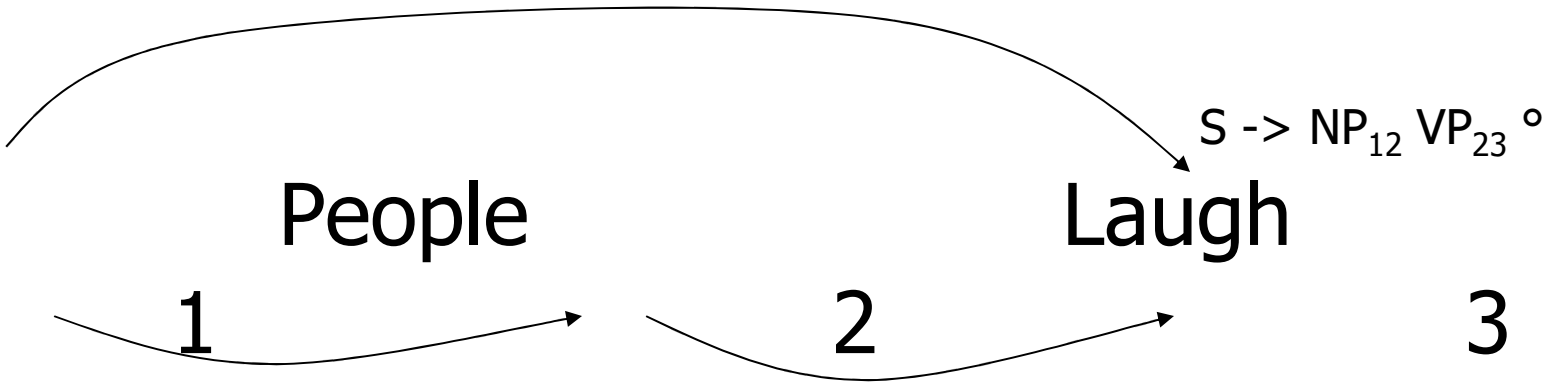
Represents positions

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

End position unknown

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

Bottom-Up Parsing (pictorial representation)



N_{12}
V_{12}
NP_{12} -> N_{12} °
VP_{12} -> V_{12} °
$S_{1?}$ -> NP_{12} ° $VP_{2?}$

N_{23}
V_{23}
NP_{23} -> N_{23} °
VP_{23} -> 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

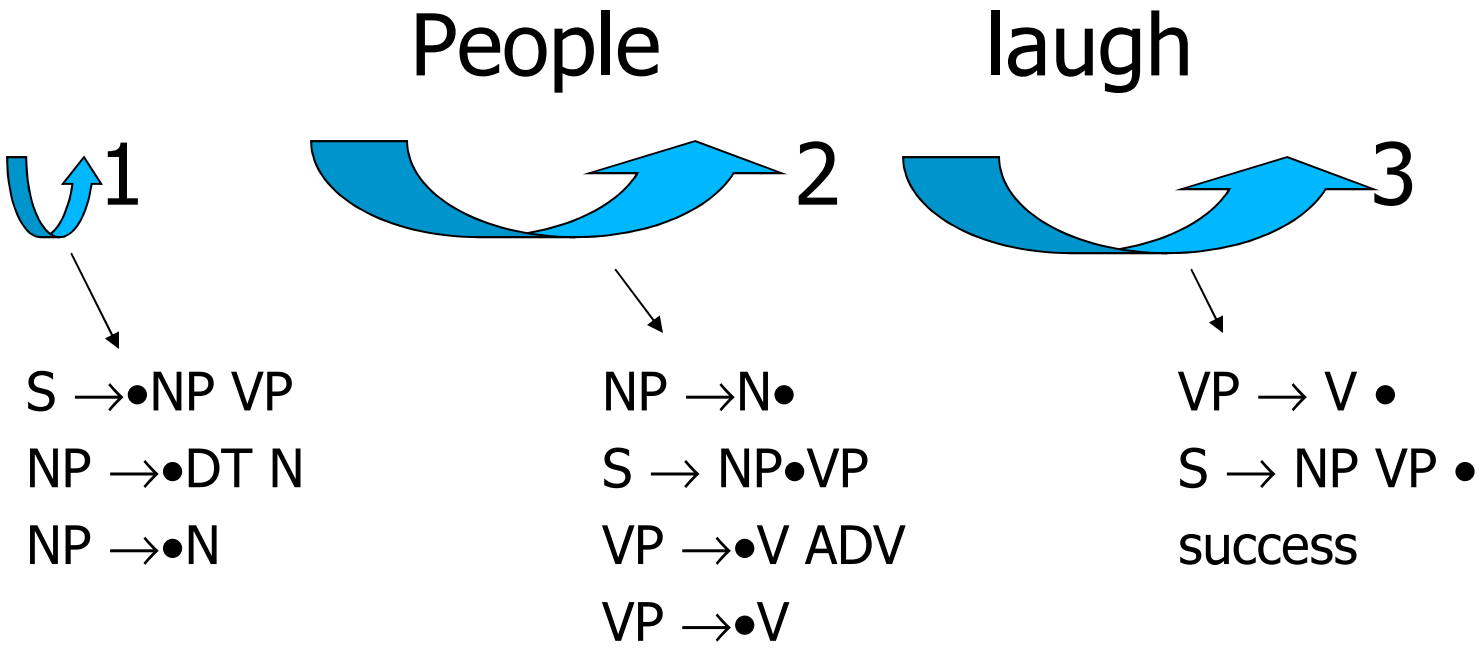
Top-Down Bottom-Up Chart Parsing

- 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 – $S \rightarrow NP\ VP$

    $NP \rightarrow DT\ N \mid N$

    $VP \rightarrow V\ ADV \mid V$

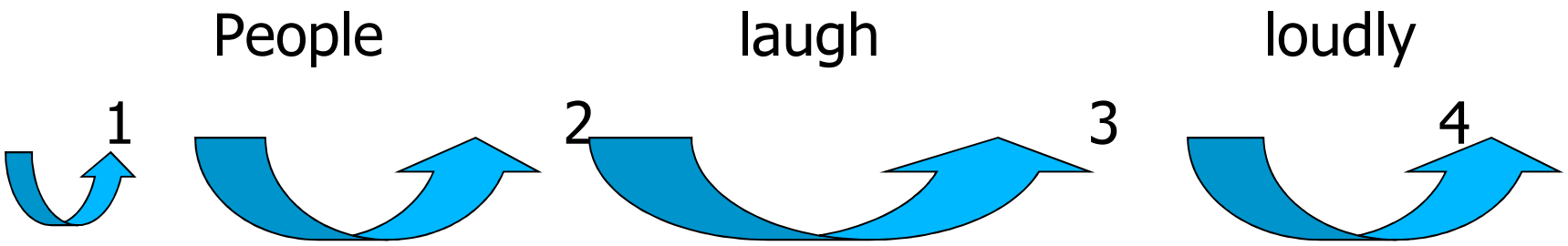
Transitive Closure



Arcs in Parsing

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

Example



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 –

$S \rightarrow NP\ VP$

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

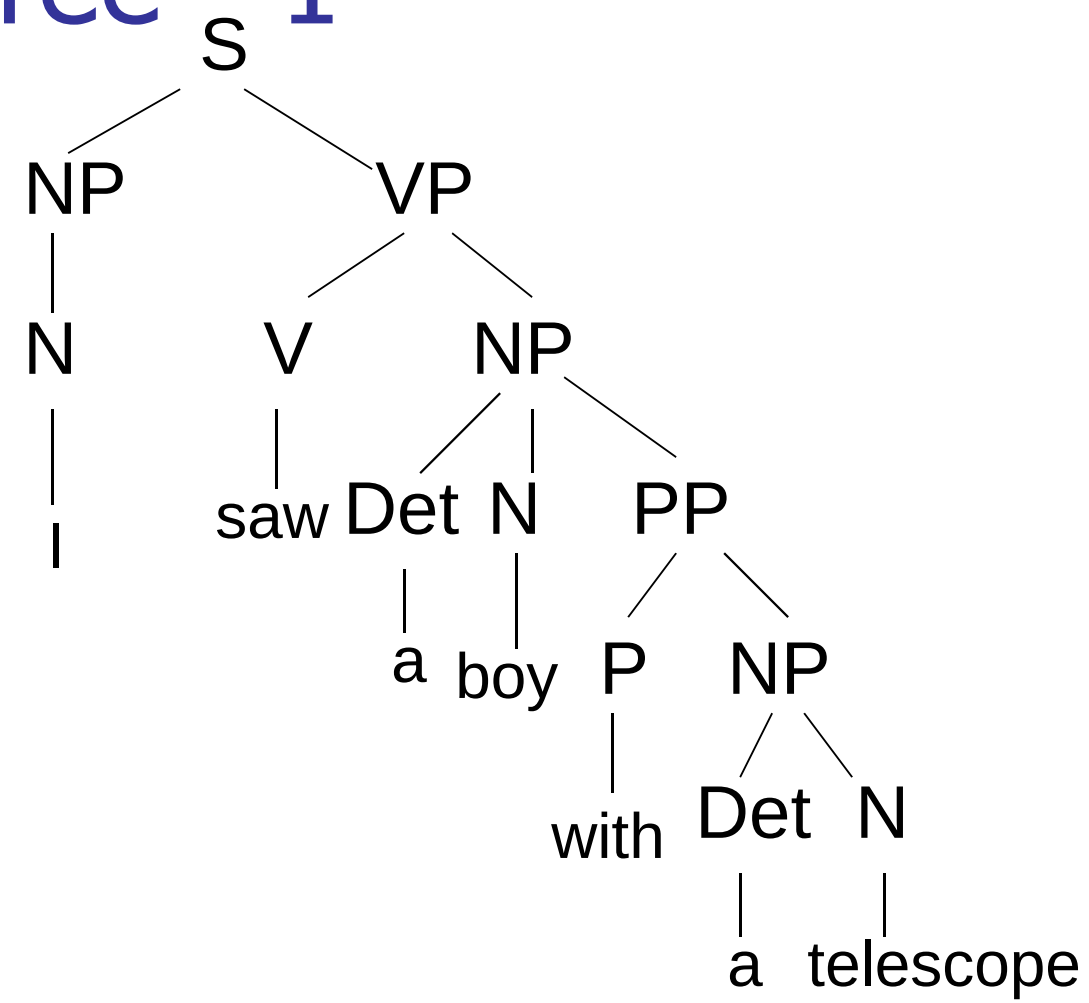
$PP \rightarrow P\ NP$

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

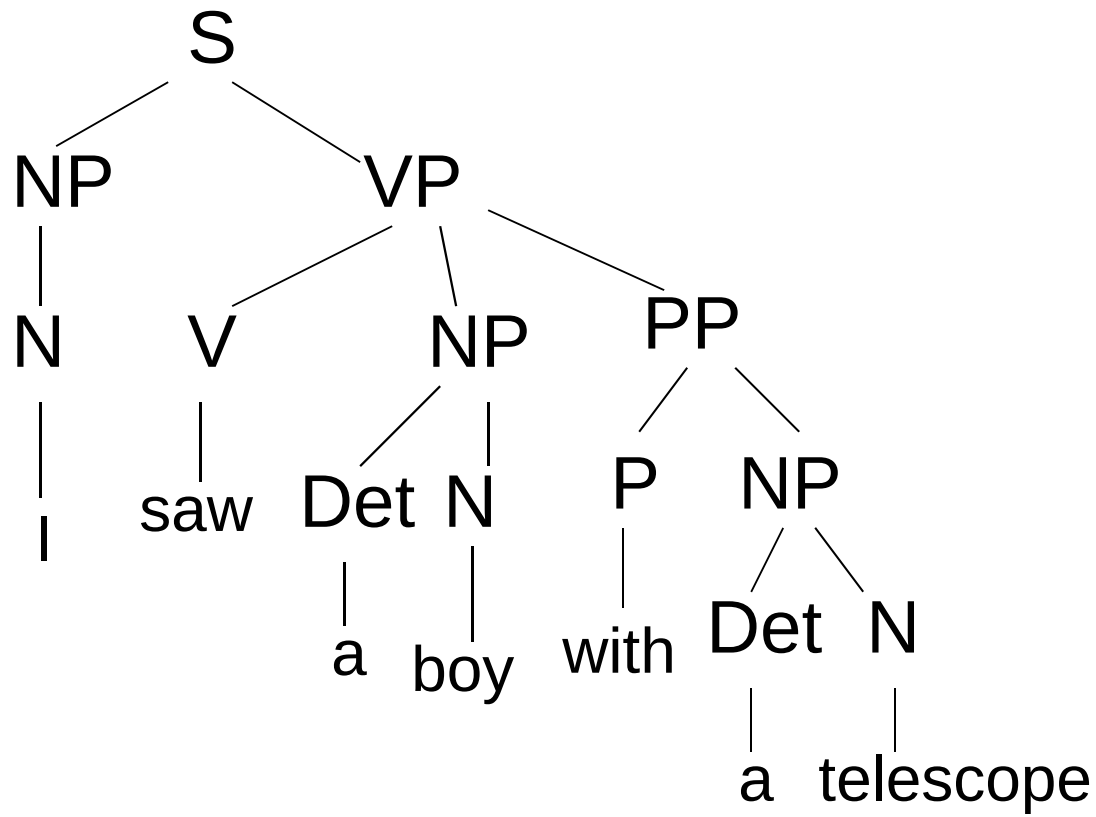
For the sentence,

"I saw a boy with a telescope"

Parse Tree - 1



Parse Tree -2



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

$V - NP_1 - P - NP_2$

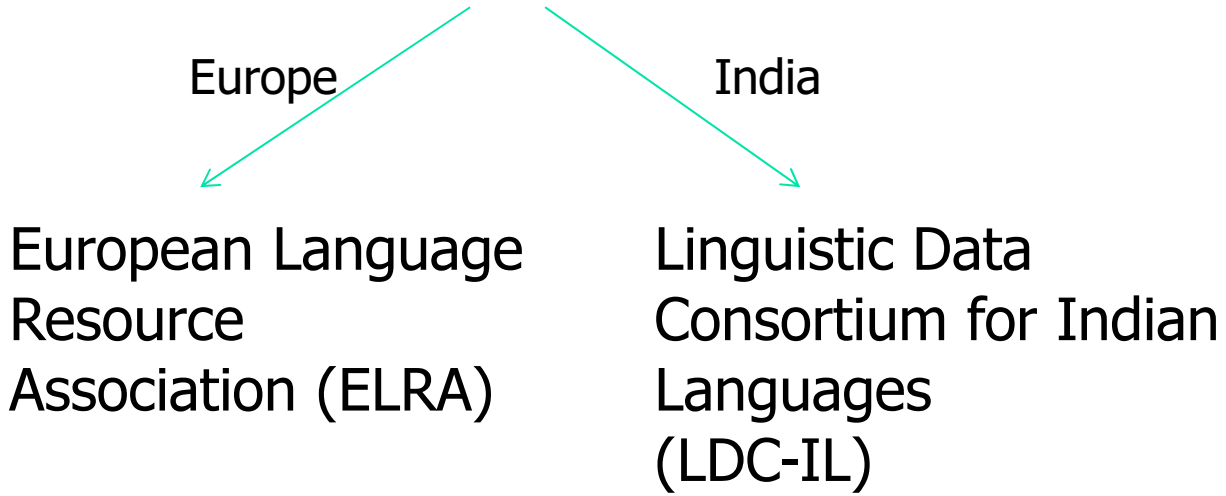
(Here P means preposition)

NP_2 attaches to NP_1 ?

or NP_2 attaches to V ?

- Xerox Parsers
  - XLE Parser (Freely available)
  - XIP Parser (expensive)

Linguistic Data Consortium
(LDC at UPenn)



Parse Trees for a Structurally Ambiguous Sentence

Let the grammar be –

$S \rightarrow NP\ VP$

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

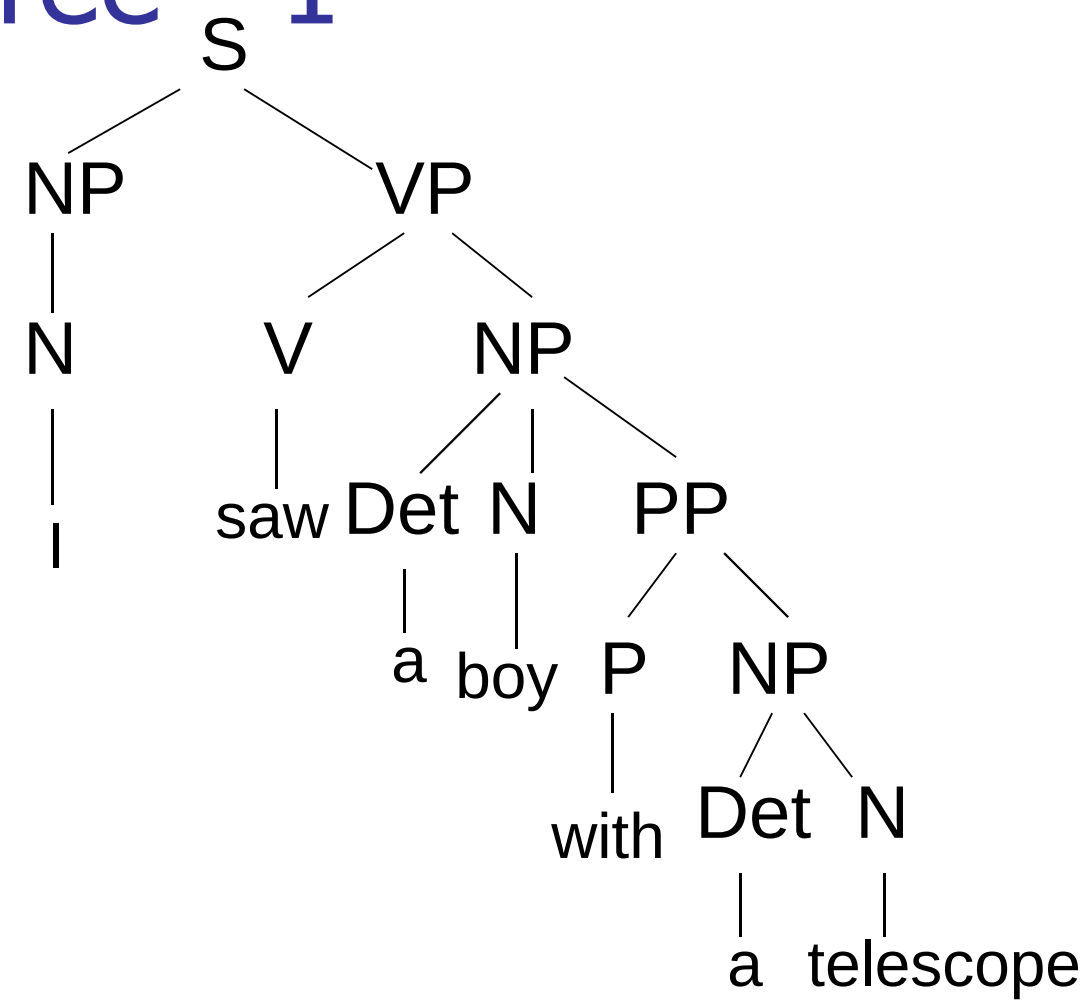
$PP \rightarrow P\ NP$

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

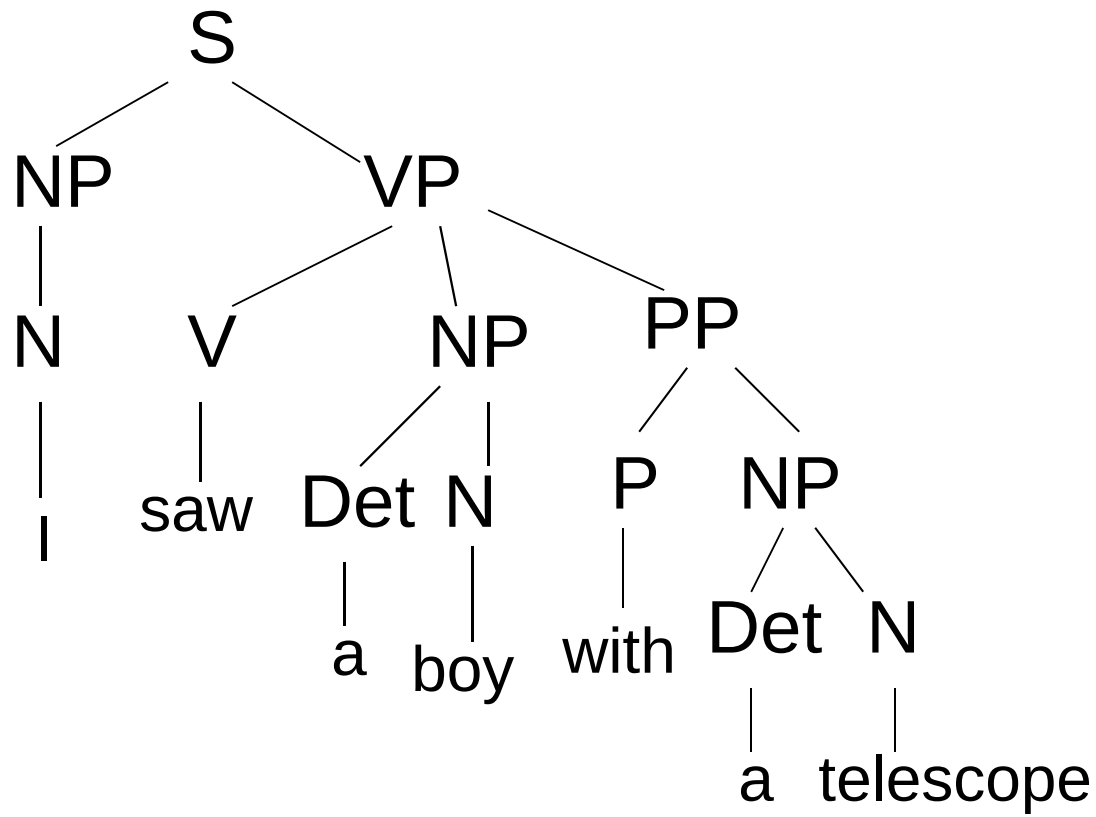
For the sentence,

"I saw a boy with a telescope"

Parse Tree - 1



Parse Tree -2



Parsing Structural Ambiguity

Parsing for Structurally Ambiguous Sentences

- Sentence "I saw a boy with a telescope"
- Grammar:

S	→	NP VP
NP	→	ART N | ART N PP | PRON
VP	→	V NP PP | V NP
ART	→	a | an | the
N	→	boy | telescope
PRON	→	I
V	→	saw

Ambiguous Parses

- Two possible parses:
  - PP attached with Verb (i.e. *I used a telescope to see*)
  - (S (NP (PRON "I")) (VP (V "saw")
    (NP ((ART "a") (N "boy"))
    (PP (P "with") (NP (ART "a") (N "telescope")))))
  - PP attached with Noun (i.e. *boy had a telescope*)
  - (S (NP (PRON "I")) (VP (V "saw")
    (NP ((ART "a") (N "boy")
    (PP (P "with") (NP (ART "a") (N "telescope"))))))

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP
2	((NP VP) 1)	–	–	Use NP → ART N | ART N PP | PRON

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP
2	((NP VP) 1)	–	–	Use NP → ART N | ART N PP | PRON
3	((ART N VP) 1)	(a) ((ART N PP VP) 1) (b) ((PRON VP) 1)	–	ART does not match "I", backup state (b) used

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP
2	((NP VP) 1)	–	–	Use NP → ART N | ART N PP | PRON
3	((ART N VP) 1)	(a) ((ART N PP VP) 1) (b) ((PRON VP) 1)	–	ART does not match "I", backup state (b) used
3 B	((PRON VP) 1)	–	–	

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP
2	((NP VP) 1)	–	–	Use NP → ART N | ART N PP | PRON
3	((ART N VP) 1)	(a) ((ART N PP VP) 1) (b) ((PRON VP) 1)	–	ART does not match "I", backup state (b) used
3 B	((PRON VP) 1)	–	–	
4	((VP) 2)	–	Consumed "I"	

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP
2	((NP VP) 1)	–	–	Use NP → ART N | ART N PP | PRON
3	((ART N VP) 1)	(a) ((ART N PP VP) 1) (b) ((PRON VP) 1)	–	ART does not match "I", backup state (b) used
3 B	((PRON VP) 1)	–	–	
4	((VP) 2)	–	Consumed "I"	
5	((V NP PP) 2)	((V NP) 2)	–	Verb Attachment Rule used

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP
2	((NP VP) 1)	–	–	Use NP → ART N | ART N PP | PRON
3	((ART N VP) 1)	(a) ((ART N PP VP) 1) (b) ((PRON VP) 1)	–	ART does not match "I", backup state (b) used
3B	((PRON VP) 1)	–	–	
4	((VP) 2)	–	Consumed "I"	
5	((V NP PP) 2)	((V NP) 2)	–	Verb Attachment Rule used
6	((NP PP) 3)	–	Consumed "saw"	

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP
2	((NP VP) 1)	–	–	Use NP → ART N | ART N PP | PRON
3	((ART N VP) 1)	(a) ((ART N PP VP) 1) (b) ((PRON VP) 1)	–	ART does not match "I", backup state (b) used
3 B	((PRON VP) 1)	–	–	
4	((VP) 2)	–	Consumed "I"	
5	((V NP PP) 2)	((V NP) 2)	–	Verb Attachment Rule used
6	((NP PP) 3)	–	Consumed "saw"	
7	((ART N PP) 3)	(a) ((ART N PP PP) 3) (b) ((PRON PP) 3)		

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP
2	((NP VP) 1)	–	–	Use NP → ART N | ART N PP | PRON
3	((ART N VP) 1)	(a) ((ART N PP VP) 1) (b) ((PRON VP) 1)	–	ART does not match "I", backup state (b) used
3 B	((PRON VP) 1)	–	–	
4	((VP) 2)	–	Consumed "I"	
5	((V NP PP) 2)	((V NP) 2)	–	Verb Attachment Rule used
6	((NP PP) 3)	–	Consumed "saw"	
7	((ART N PP) 3)	(a) ((ART N PP PP) 3) (b) ((PRON PP) 3)		
8	((N PP) 4)	–	Consumed "a"	

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP
2	((NP VP) 1)	–	–	Use NP → ART N | ART N PP | PRON
3	((ART N VP) 1)	(a) ((ART N PP VP) 1) (b) ((PRON VP) 1)	–	ART does not match "I", backup state (b) used
3 B	((PRON VP) 1)	–	–	
4	((VP) 2)	–	Consumed "I"	
5	((V NP PP) 2)	((V NP) 2)	–	Verb Attachment Rule used
6	((NP PP) 3)	–	Consumed "saw"	
7	((ART N PP) 3)	(a) ((ART N PP PP) 3) (b) ((PRON PP) 3)		
8	((N PP) 4)	–	Consumed "a"	
9	((PP) 5)	–	Consumed	

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP
2	((NP VP) 1)	–	–	Use NP → ART N | ART N PP | PRON
3	((ART N VP) 1)	(a) ((ART N PP VP) 1) (b) ((PRON VP) 1)	–	ART does not match "I", backup state (b) used
3 B	((PRON VP) 1)	–	–	
4	((VP) 2)	–	Consumed "I"	
5	((V NP PP) 2)	((V NP) 2)	–	Verb Attachment Rule used
6	((NP PP) 3)	–	Consumed "saw"	
7	((ART N PP) 3)	(a) ((ART N PP PP) 3) (b) ((PRON PP) 3)		
8	((N PP) 4)	–	Consumed "a"	
9	((PP) 5)	–	Consumed	

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP
…	…	…	…	…
7	((ART N PP) 3)	(a) ((ART N PP PP) 3) (b) ((PRON PP) 3)		
8	((N PP) 4)	–	Consumed "a"	
9	((PP) 5)	–	Consumed "boy"	
10	((P NP) 5)	–	–	
11	((NP) 6)	–	Consumed "with"	

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP
…	…	…	…	…
7	((ART N PP) 3)	(a) ((ART N PP PP) 3) (b) ((PRON PP) 3)		
8	((N PP) 4)	–	Consumed "a"	
9	((PP) 5)	–	Consumed "boy"	
10	((P NP) 5)	–	–	
11	((NP) 6)	–	Consumed "with"	
12	((ART N) 6)	(a) ((ART N PP) 6) (b) ((PRON) 6)	–	

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP
…	…	…	…	…
7	((ART N PP) 3)	(a) ((ART N PP PP) 3) (b) ((PRON PP) 3)		
8	((N PP) 4)	–	Consumed "a"	
9	((PP) 5)	–	Consumed "boy"	
10	((P NP) 5)	–	–	
11	((NP) 6)	–	Consumed "with"	
12	((ART N) 6)	(a) ((ART N PP) 6) (b) ((PRON) 6)	–	
13	((N) 7)	–	Consumed "a"	

Top Down Parse

	State	Backup State	Action	Comments
1	((S) 1)	–	–	Use S → NP VP
…	…	…	…	…
7	((ART N PP) 3)	(a) ((ART N PP PP) 3) (b) ((PRON PP) 3)		
8	((N PP) 4)	–	Consumed "a"	
9	((PP) 5)	–	Consumed "boy"	
10	((P NP) 5)	–	–	
11	((NP) 6)	–	Consumed "with"	
12	((ART N) 6)	(a) ((ART N PP) 6) (b) ((PRON) 6)	–	
13	((N) 7)	–	Consumed "a"	
14	((–) 8)	–	Consume "telescope" Finish Parsing	

Top Down Parsing - Observations

- Top down parsing gave us the Verb Attachment Parse Tree (i.e., *I used a telescope*)
- To obtain the alternate parse tree, the backup state in step 5 will have to be invoked
- Is there an efficient way to obtain all parses ?

Bottom Up Parse

I	saw	a	boy	with	a	telescope
1	2	3	4	5	6	7	8

Colour Scheme :

- Blue for Normal Parse
- Green for Verb Attachment Parse
- Purple for Noun Attachment Parse
- Red for Invalid Parse

Bottom Up Parse

I	saw	a	boy	with	a	telescope
1	2	3	4	5	6	7	8

$NP_{12} \rightarrow PRON_{12}\bullet$

$S_{1?} \rightarrow NP_{12}\bullet VP_{2?}$

Bottom Up Parse

I	saw	a	boy	with	a	telescope
1	2	3	4	5	6	7	8

$NP_{12} \rightarrow PRON_{12}\bullet$

$S_{1?} \rightarrow NP_{12}\bullet VP_{2?}$

$VP_{2?} \rightarrow V_{23}\bullet NP_{3?}$

$VP_{2?} \rightarrow V_{23}\bullet NP_{3?}PP_{??}$

Bottom Up Parse

I	saw	a	boy	with	a	telescope
1	2	3	4	5	6	7 8

$NP_{12} \rightarrow PRON_{12}\bullet$

$S_{1?} \rightarrow NP_{12}\bullet VP_{2?}$

$VP_{2?} \rightarrow V_{23}\bullet NP_{3?}$

$VP_{2?} \rightarrow V_{23}\bullet NP_{3?}PP_{??}$

$NP_{35} \rightarrow ART_{34}\bullet N_{45}$

$NP_{3?} \rightarrow ART_{34}\bullet N_{45}PP_{5?}$

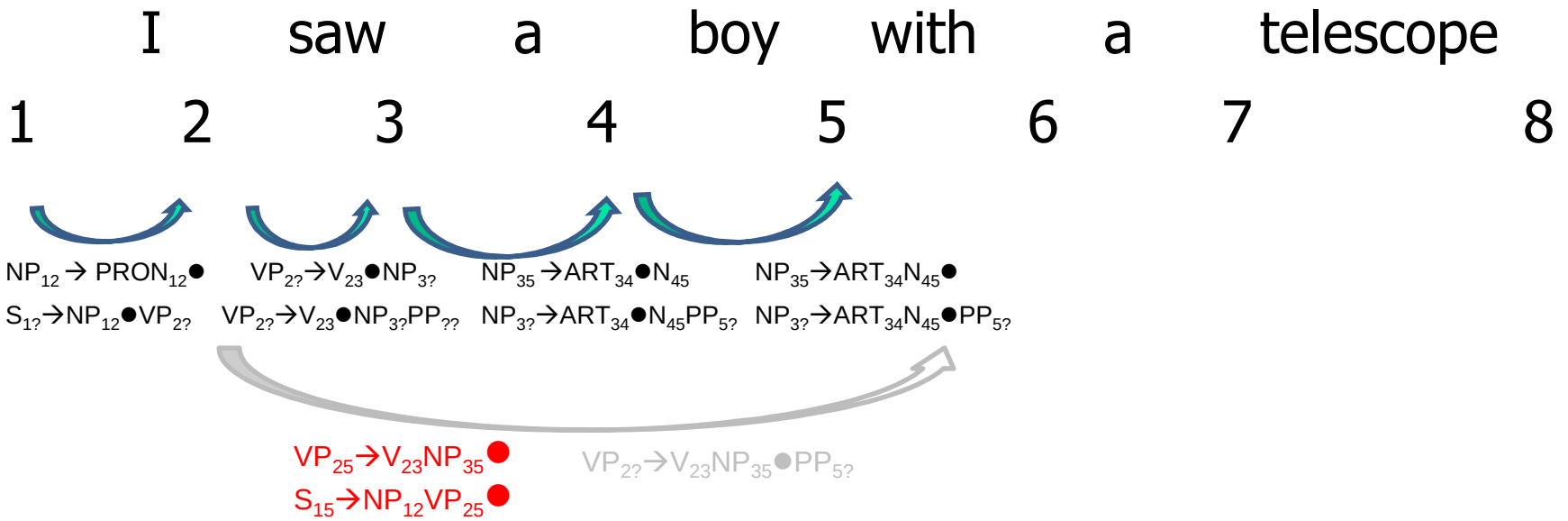
Bottom Up Parse

I	saw	a	boy	with	a	telescope
1	2	3	4	5	6	7 8

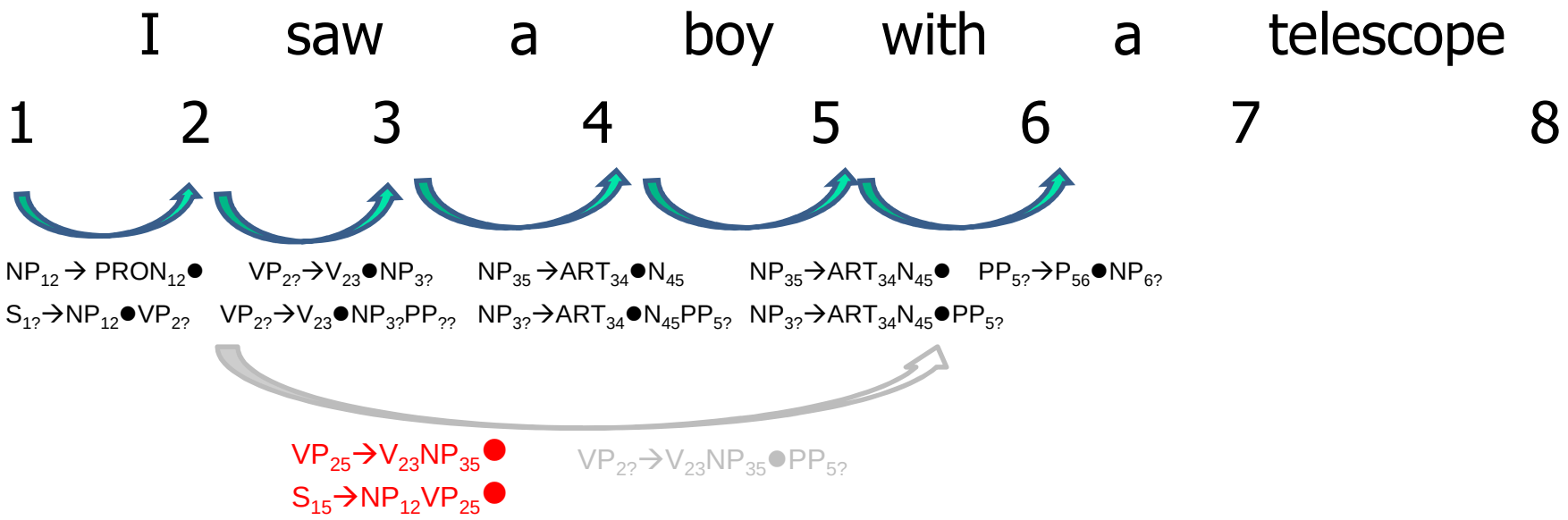
$NP_{12} \rightarrow PRON_{12}\bullet$ $VP_{2?} \rightarrow V_{23}\bullet NP_{3?}$ $NP_{35} \rightarrow ART_{34}\bullet N_{45}$ $NP_{35} \rightarrow ART_{34}N_{45}\bullet$

$S_{1?} \rightarrow NP_{12}\bullet VP_{2?}$ $VP_{2?} \rightarrow V_{23}\bullet NP_{3?}PP_{??}$ $NP_{3?} \rightarrow ART_{34}\bullet N_{45}PP_{5?}$ $NP_{3?} \rightarrow ART_{34}N_{45}\bullet PP_{5?}$

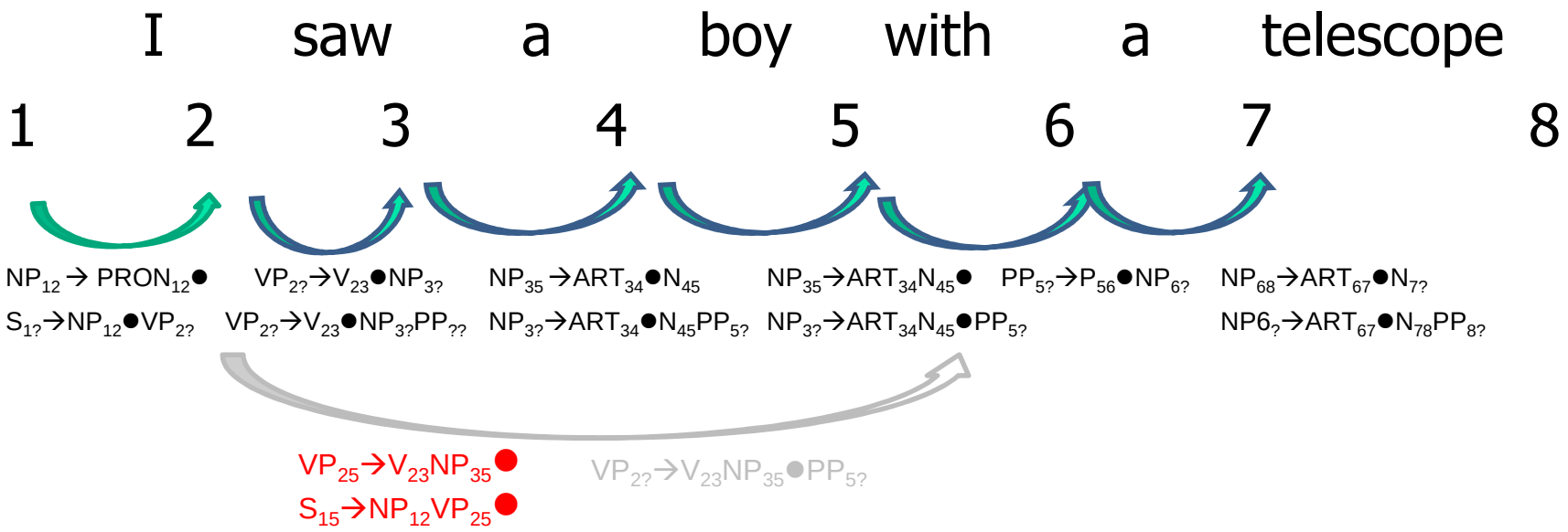
Bottom Up Parse



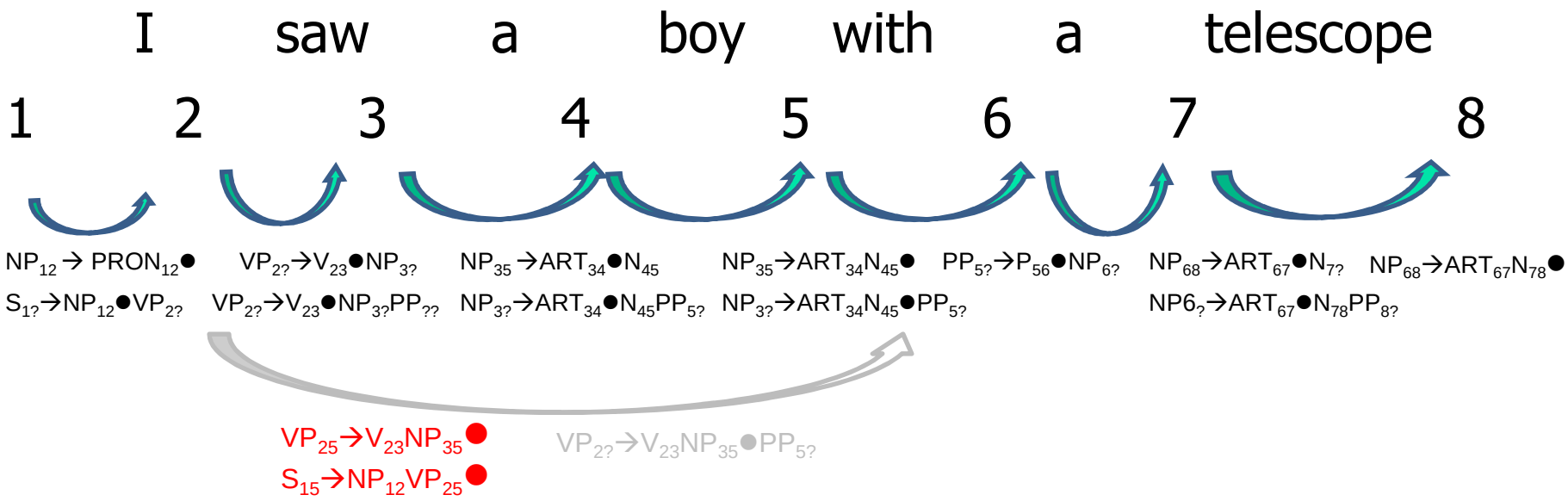
Bottom Up Parse



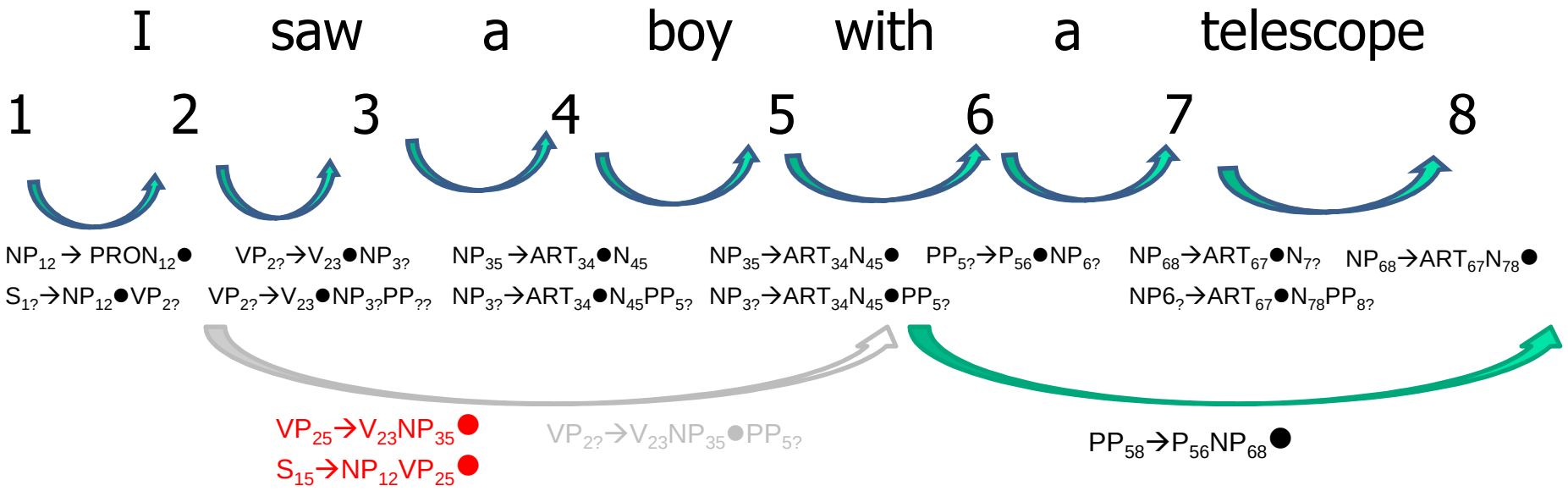
Bottom Up Parse



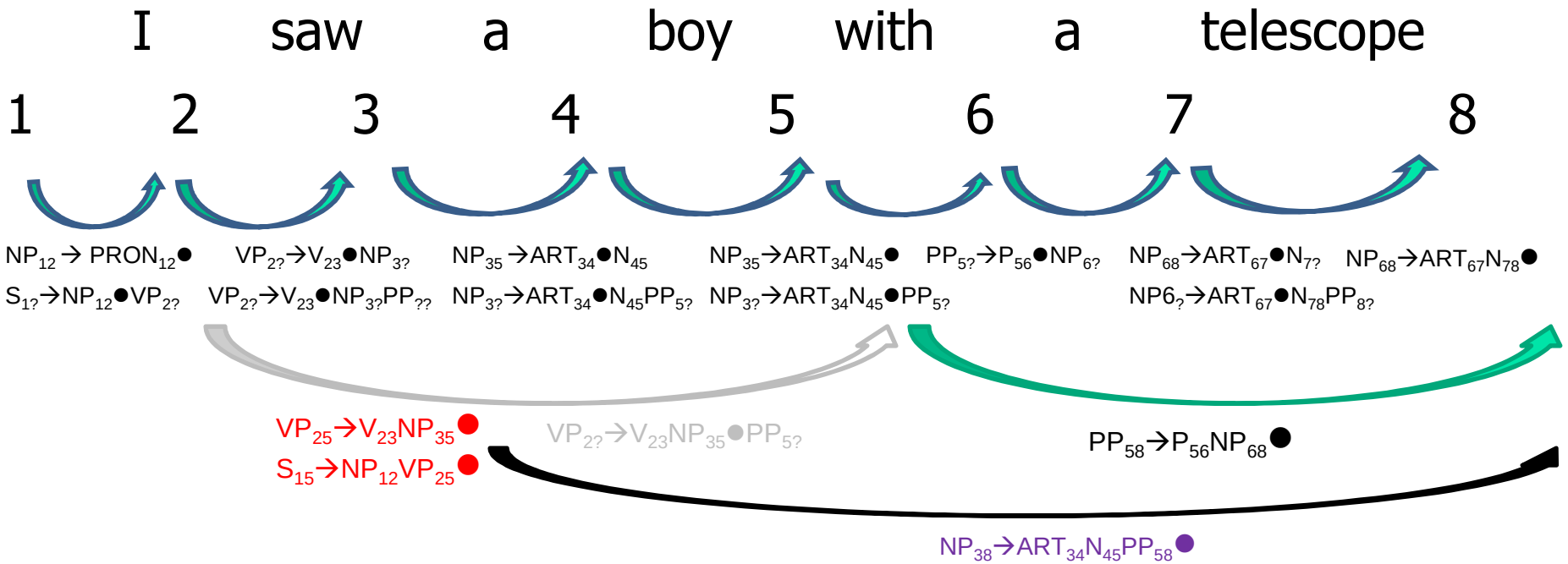
Bottom Up Parse



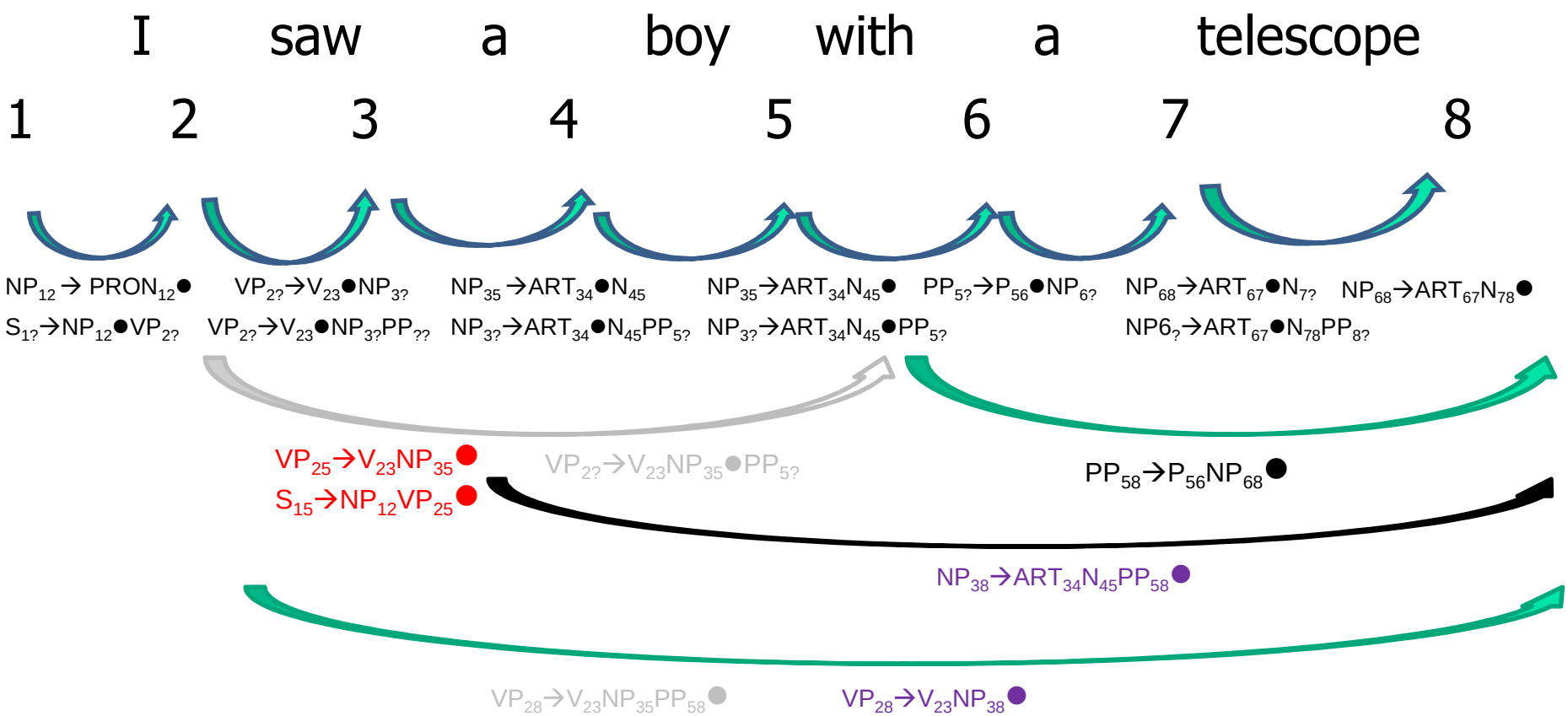
Bottom Up Parse



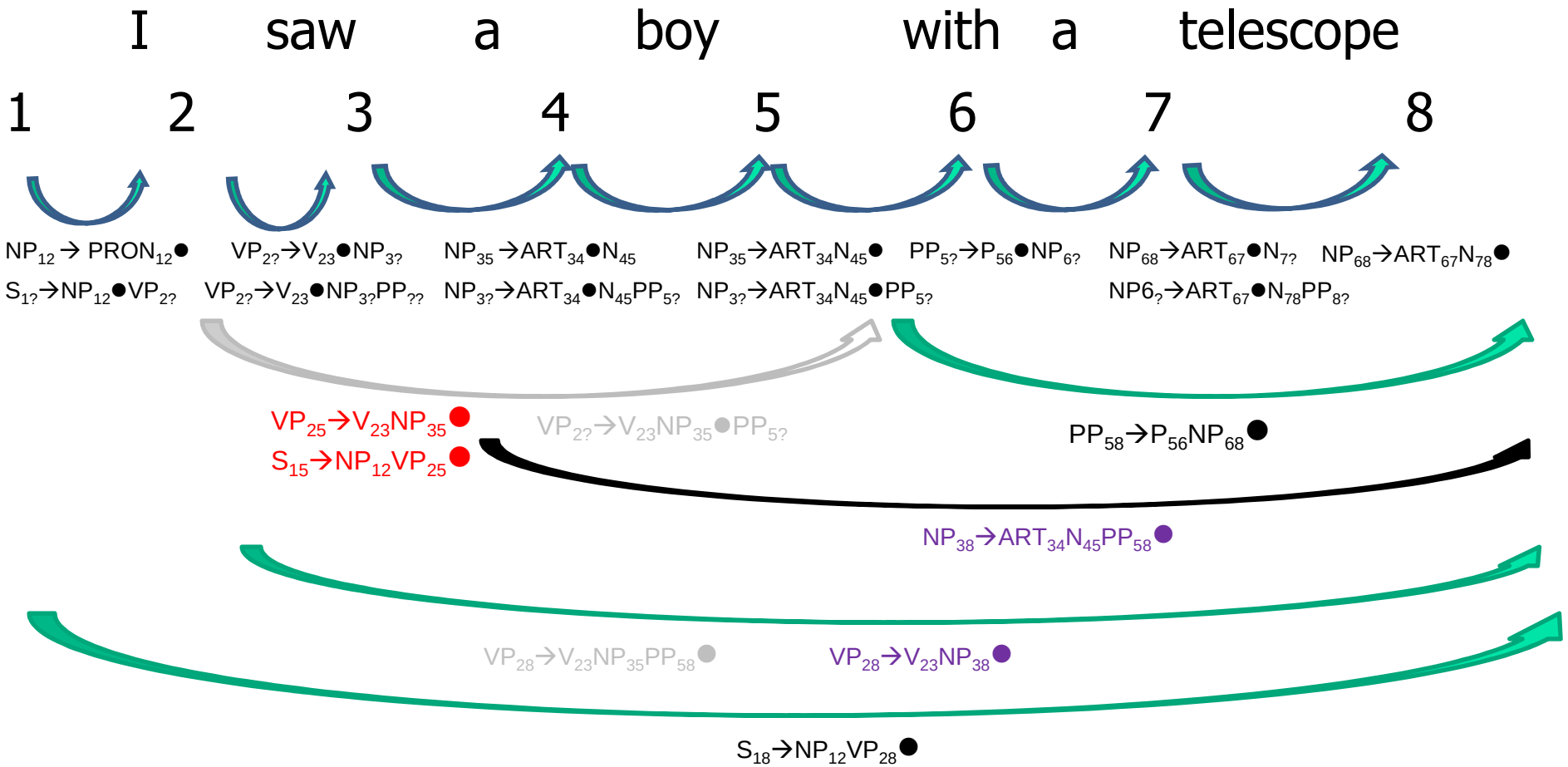
Bottom Up Parse



Bottom Up Parse



Bottom Up Parse



Bottom Up Parsing - Observations

- Both Noun Attachment and Verb Attachment Parses obtained by simply systematically applying the rules
- Numbers in subscript help in verifying the parse and getting chunks from the parse

Exercise

For the sentence,

"The man saw the boy with a telescope"
& the grammar given previously,
compare the performance of top-down,
bottom-up & top-down chart parsing.