Bitext parsing

Wilker Aziz
3/5/17

Context-Free Grammars

A **CFG** grammar G is denoted by

- a finite set of **nonterminal** symbols N
- a finite set of **terminal** symbols Σ with $\Sigma \cap N = \varnothing$
- a finite set R of **rules** of the form $X \rightarrow \alpha$ where
  - $X \in N$ and $\alpha \in (\Sigma \cup N)^*$
- $S \in N$ a distinguished **start** symbol

Let ε denote an **empty** string

Example CFG

S → NP VP	Vi → sleeps
VP → Vi	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Generative Device

Left-most derivation

- sequence of strings $\mathbf{s}_1 \ldots \mathbf{s}_n$
  - $\mathbf{s}_1 = S$
  - $\mathbf{s}_n \in \Sigma^*$
  - $\mathbf{s}_{i\geq 2}$ derived from $\mathbf{s}_{i-1}$ by picking the left-most nonterminal X
    - replacing it by some α such that $X \rightarrow \alpha \in R$

Example of Derivation

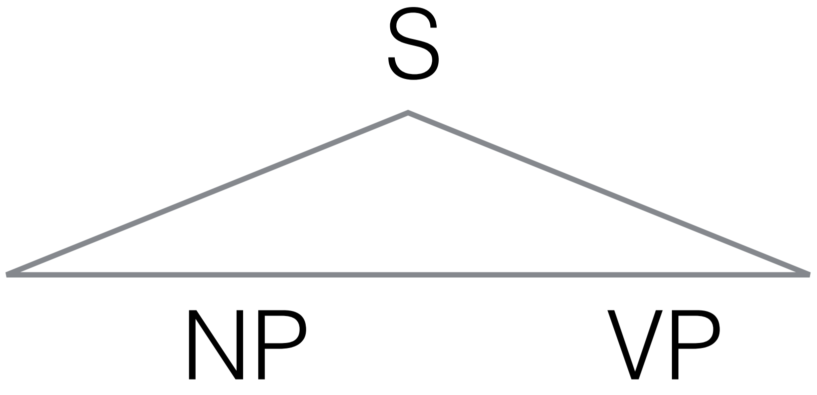
		Substitution
$\mathbf{s}_1$ =	S	S → NP VP
$\mathbf{s}_2$ =	NP VP	NP → DT NN
$\mathbf{s}_3$ =	DT NN VP	DT → the
$\mathbf{s}_4$ =	the NN VP	NN → man
$\mathbf{s}_5$ =	the man VP	VP → Vi
$\mathbf{s}_6$ =	the man Vi	Vi → sleeps
$\mathbf{s}_7$ =	the man sleeps	
$\mathbf{s}_7$ =	S $\Rightarrow^*$ the man sleeps	

Example of Generation

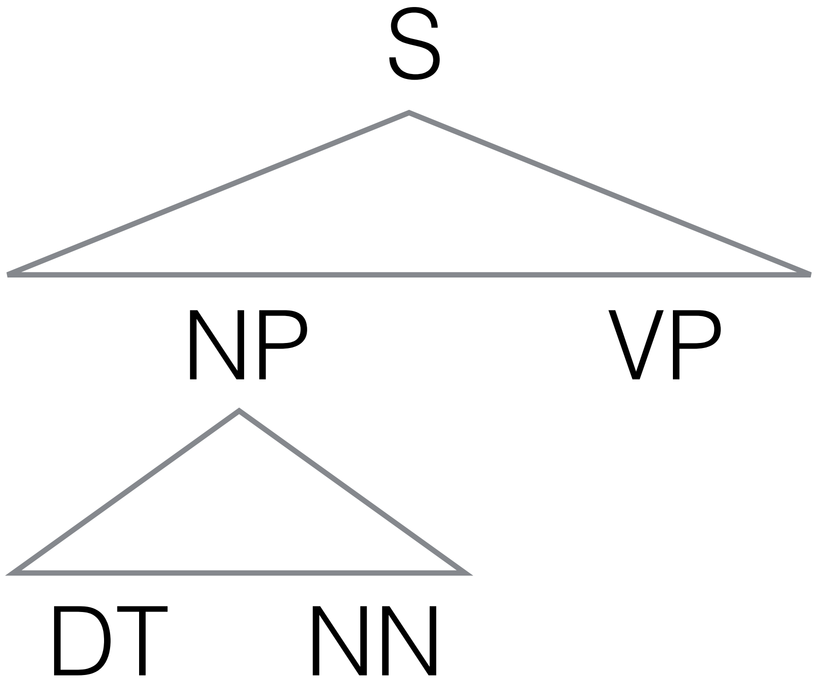
Example of Generation

S

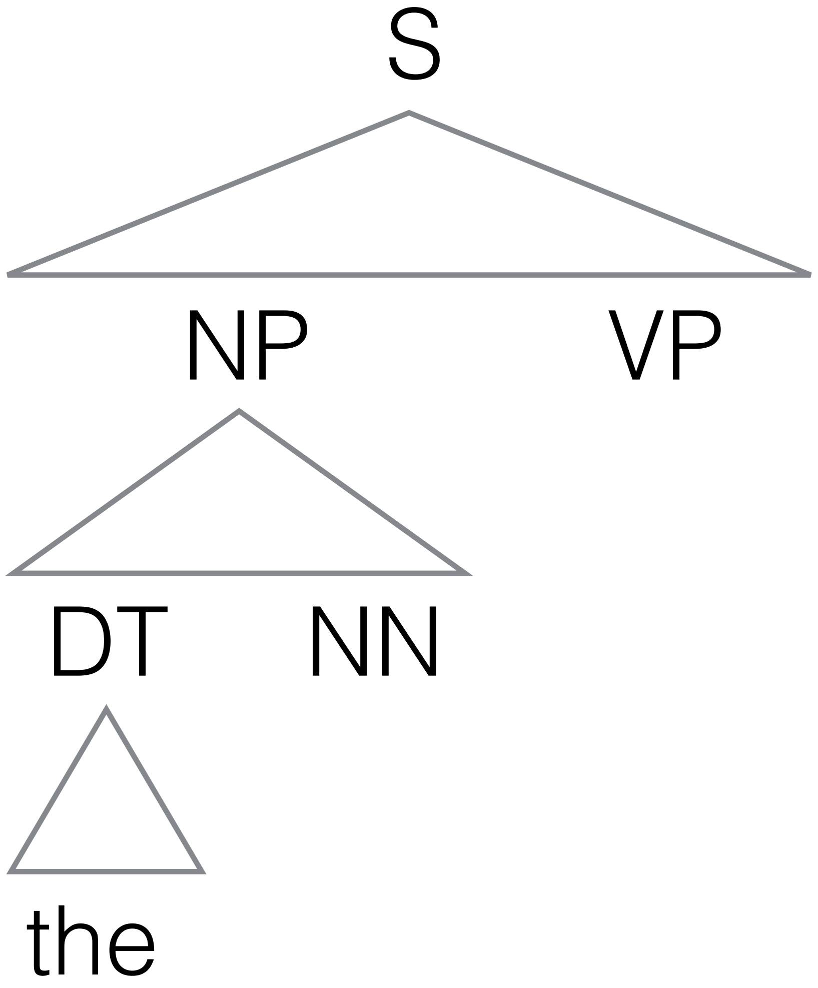
Example of Generation



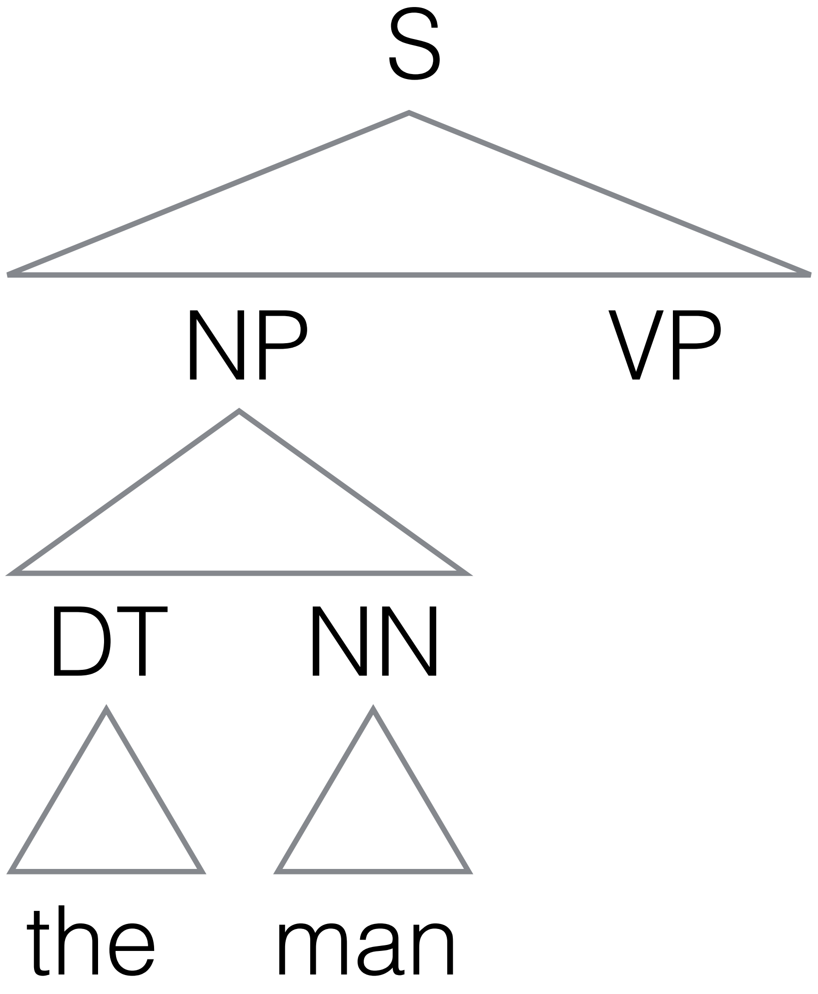
Example of Generation



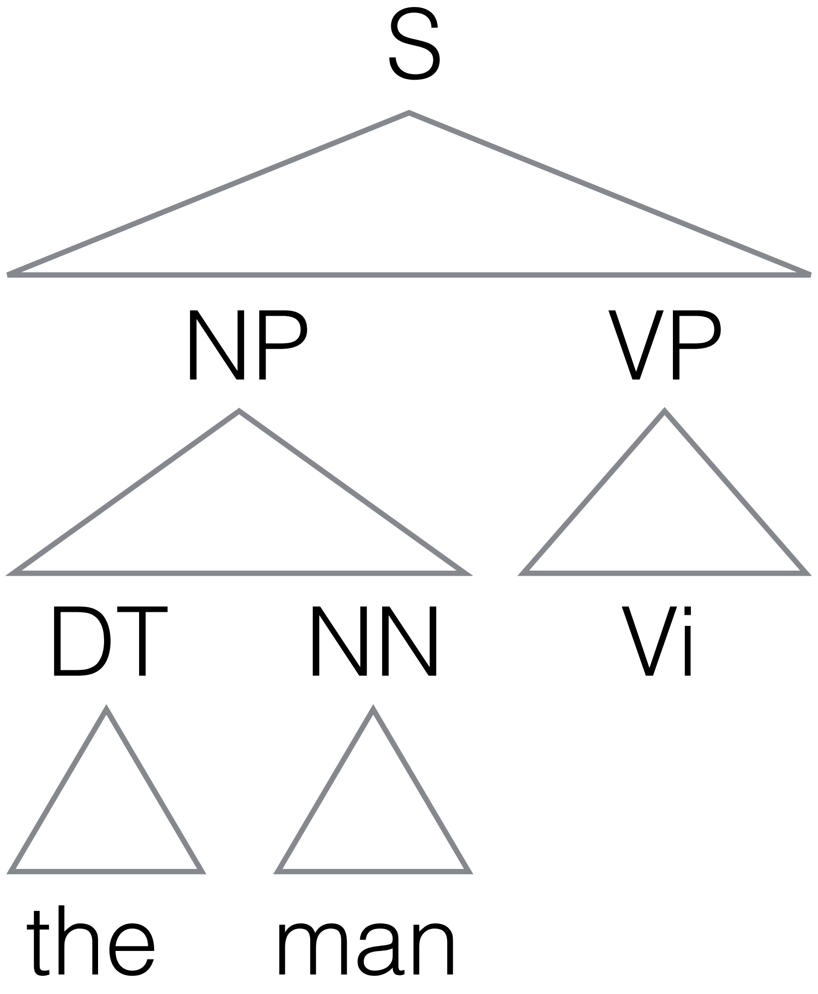
Example of Generation



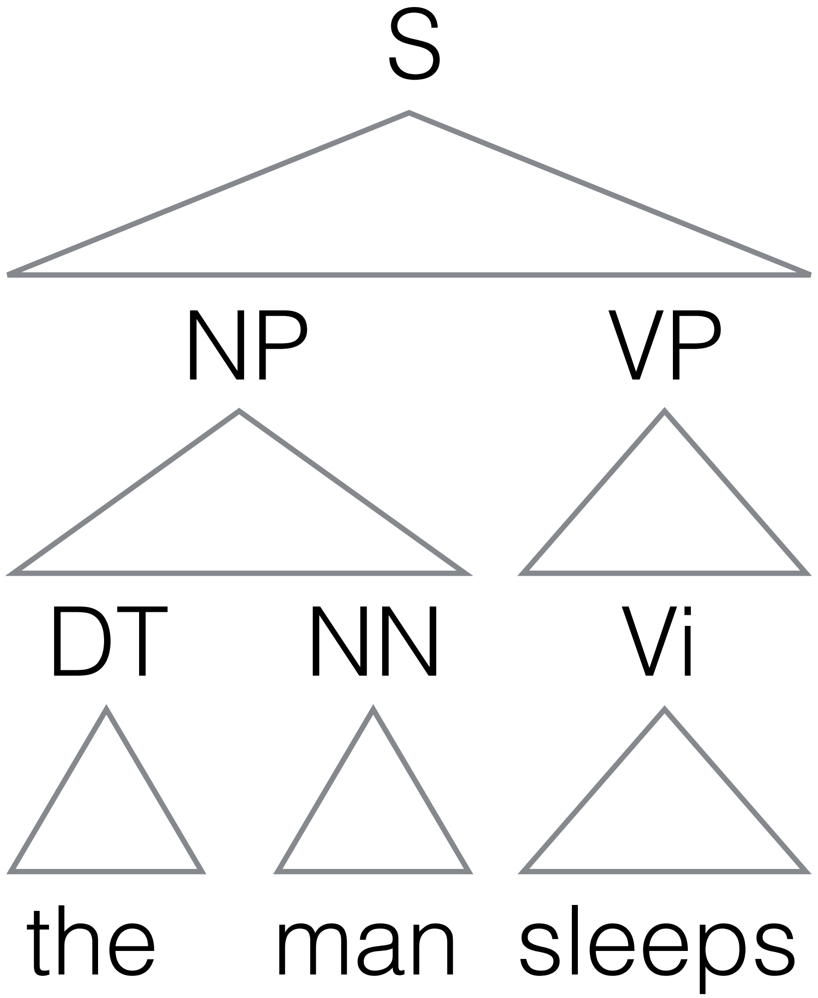
Example of Generation



Example of Generation



Example of Generation

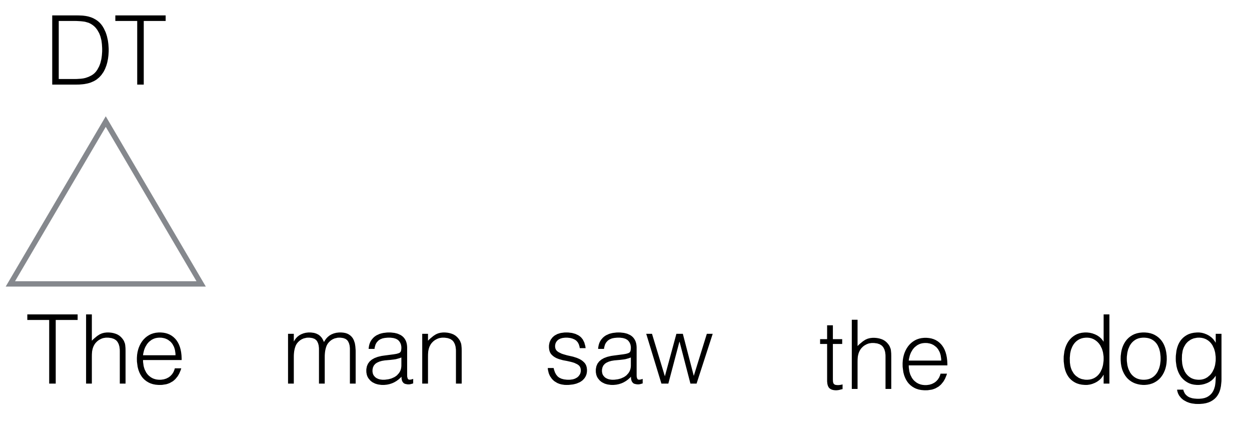


Example of Recognition

Example of Recognition

The man saw the dog

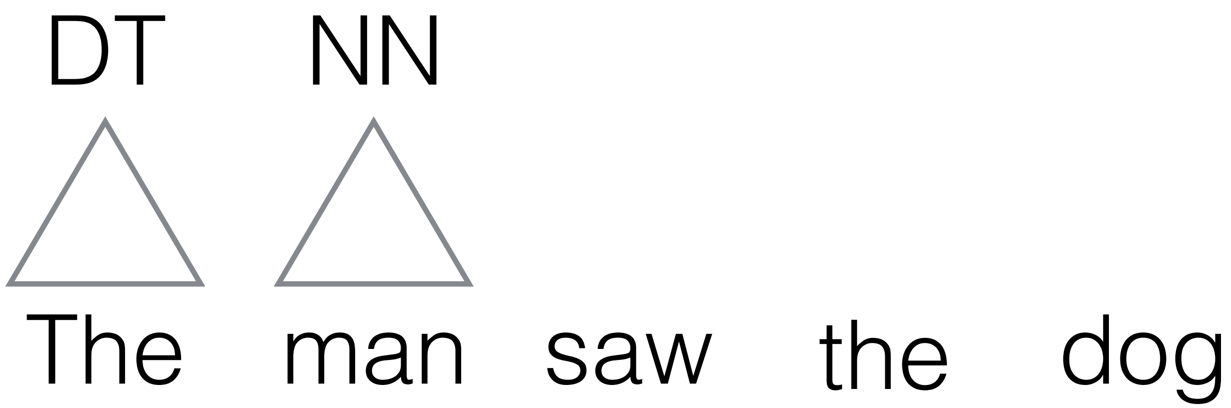
Example of Recognition



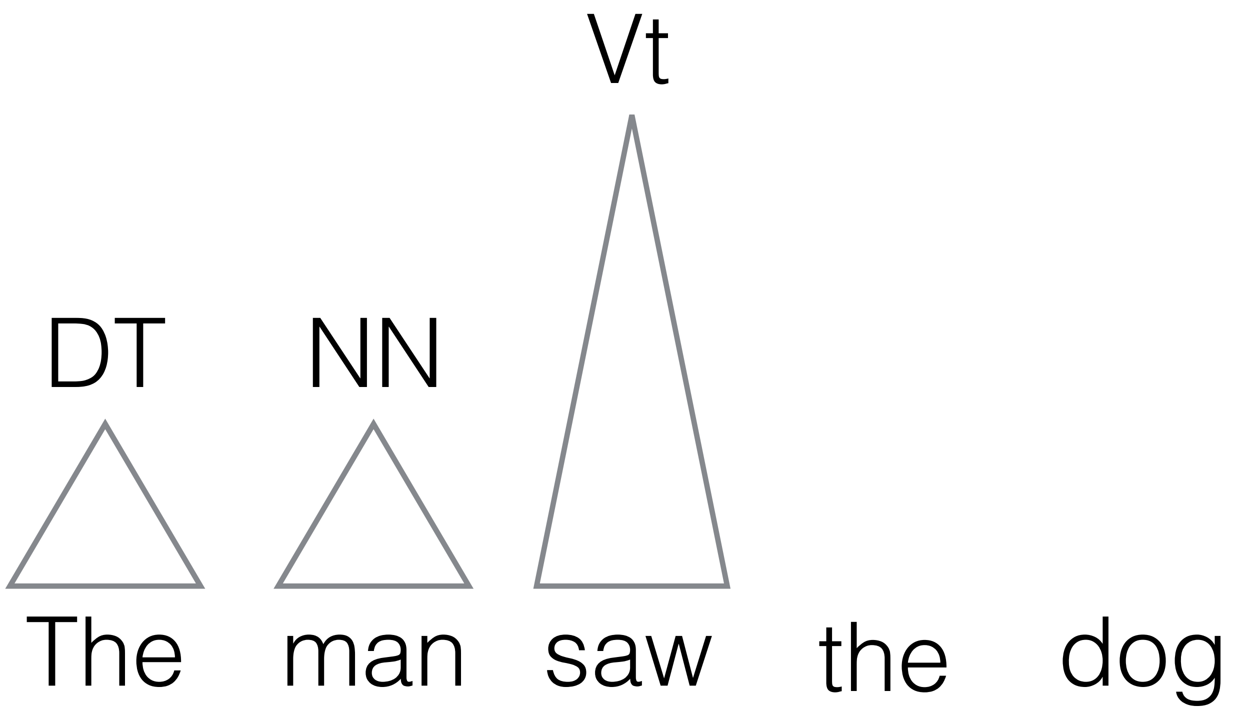
DT

The man saw the dog

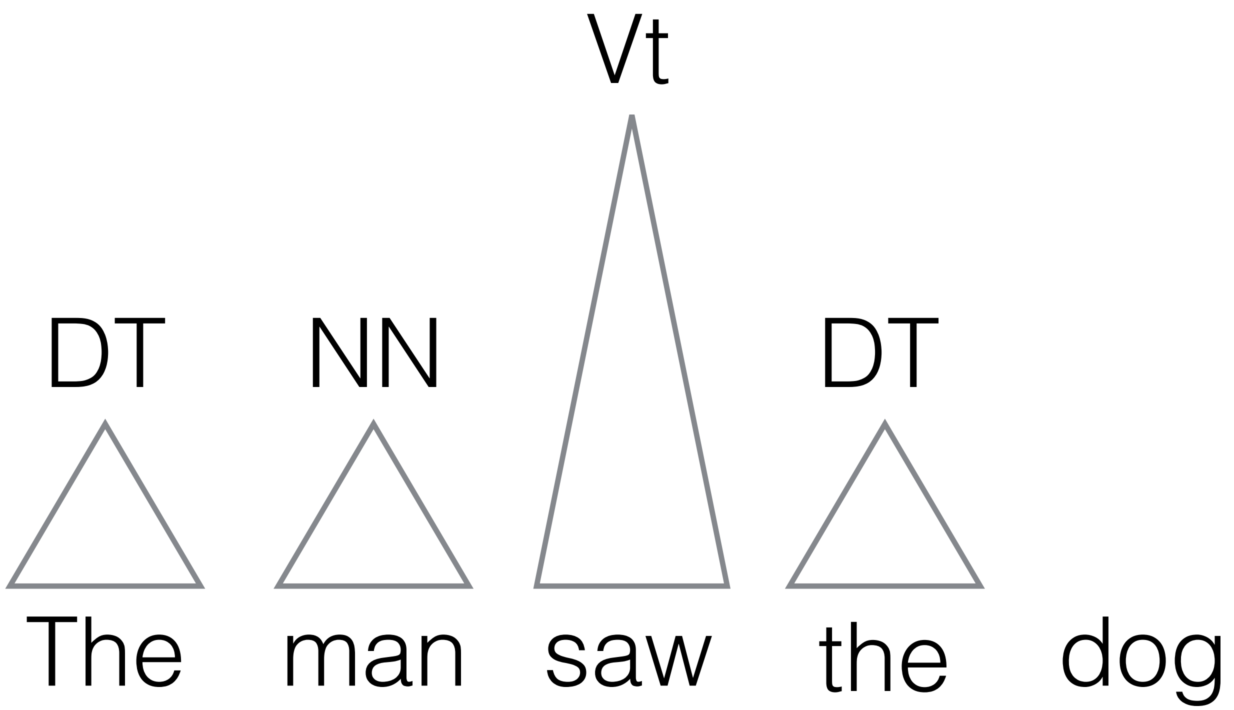
Example of Recognition



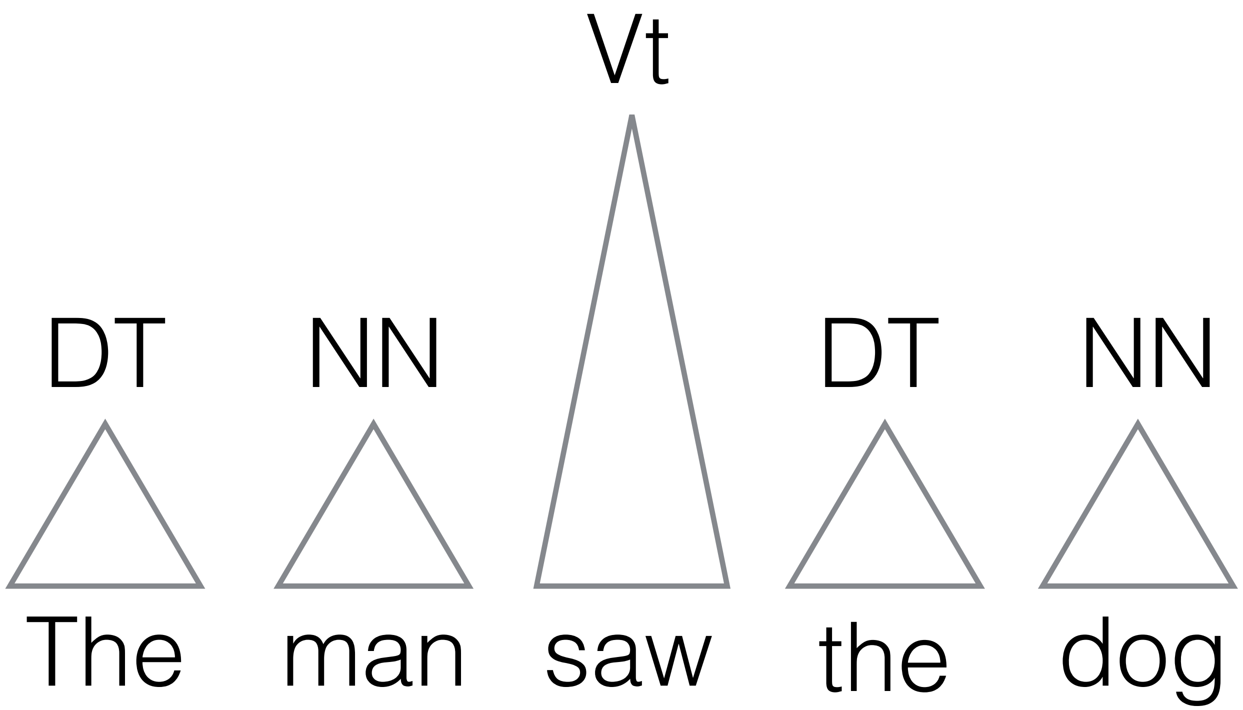
Example of Recognition



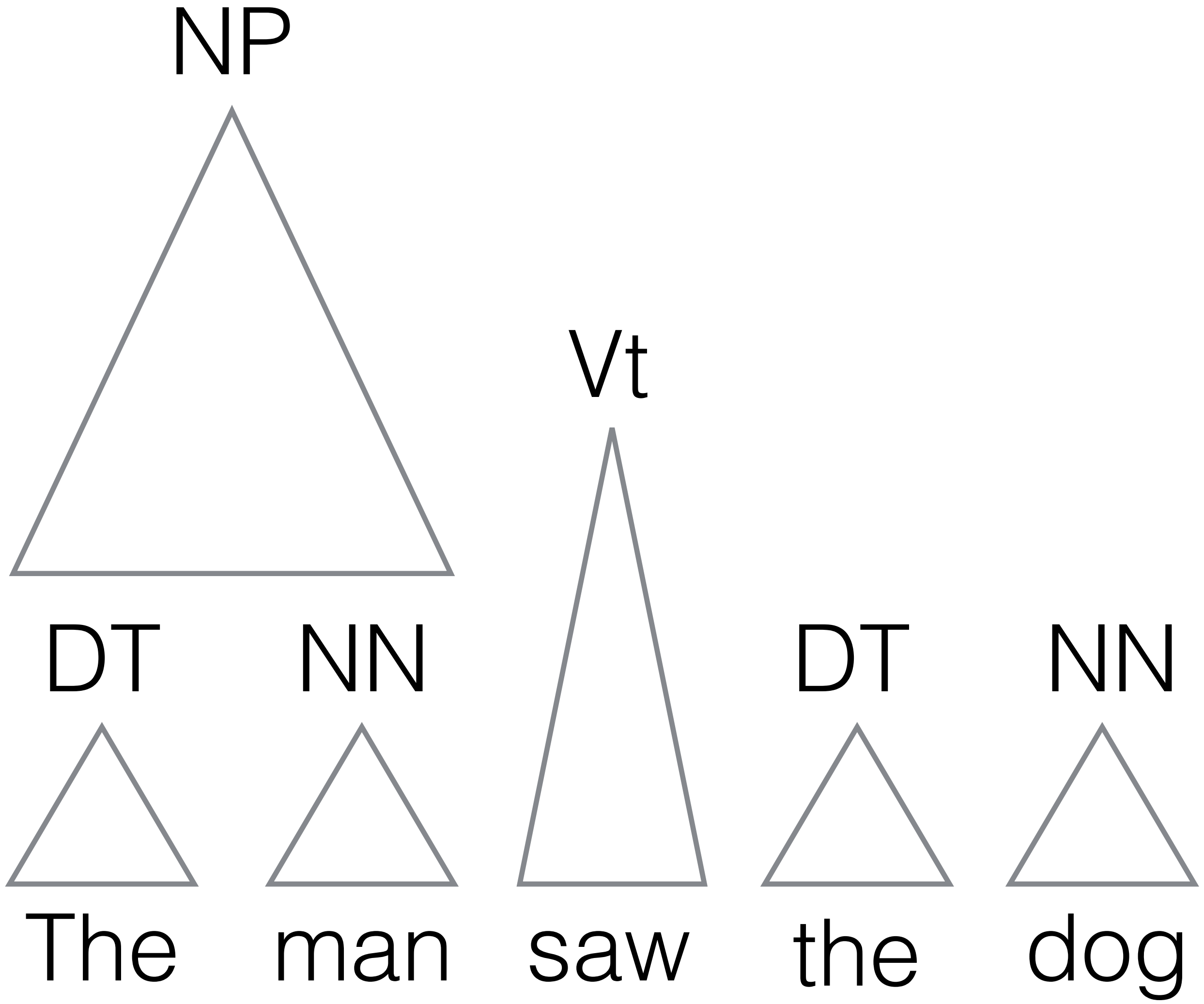
Example of Recognition



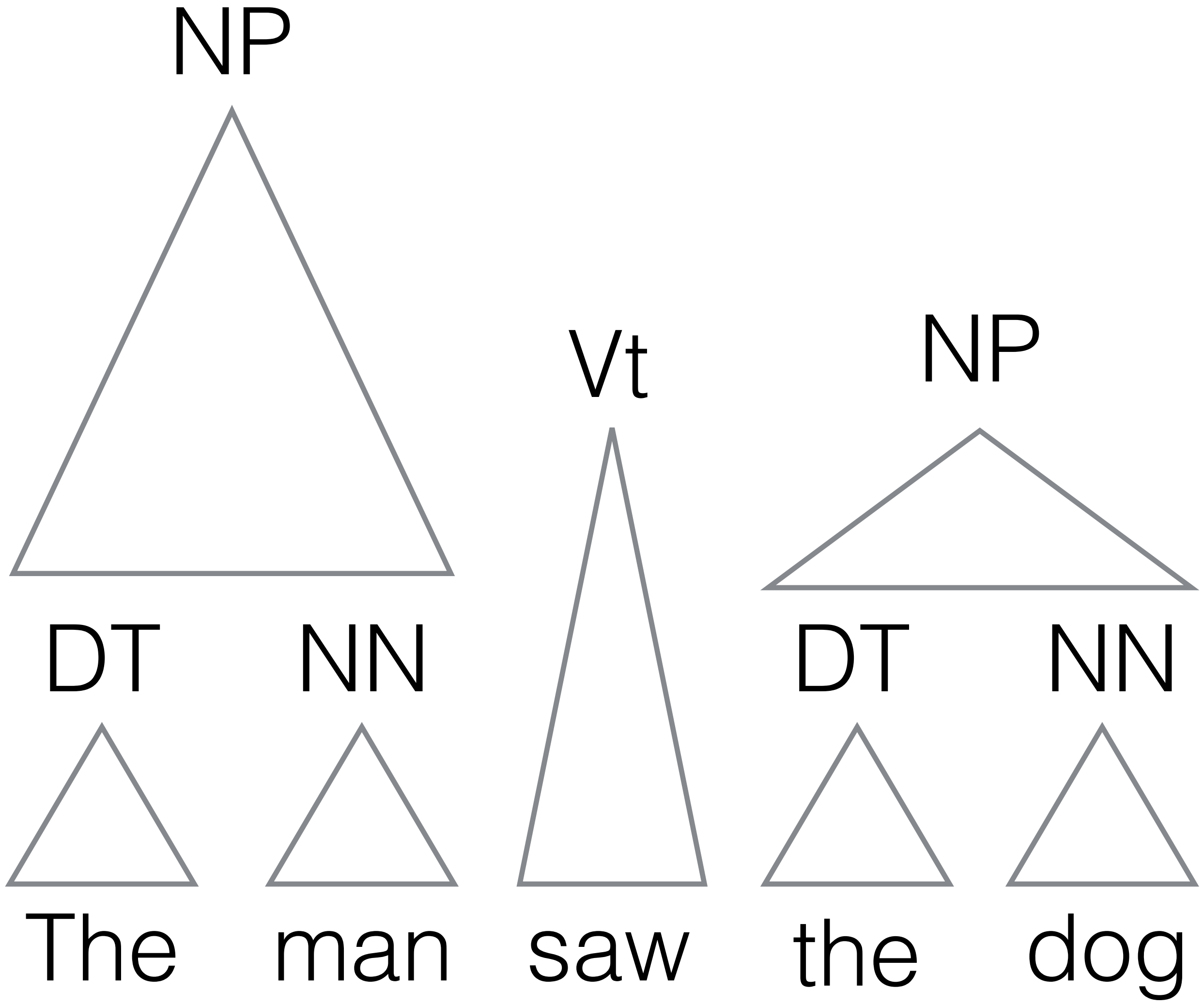
Example of Recognition



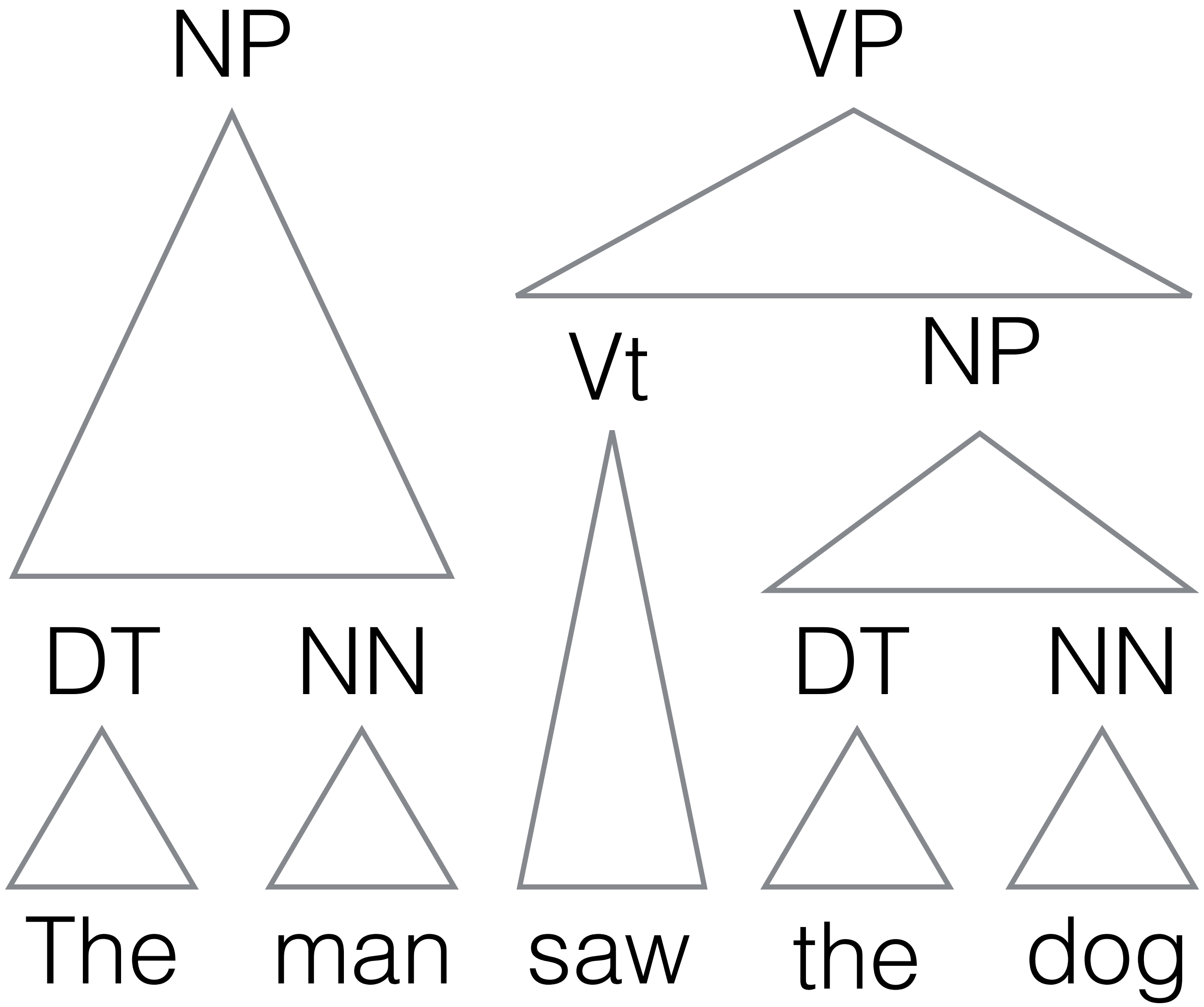
Example of Recognition



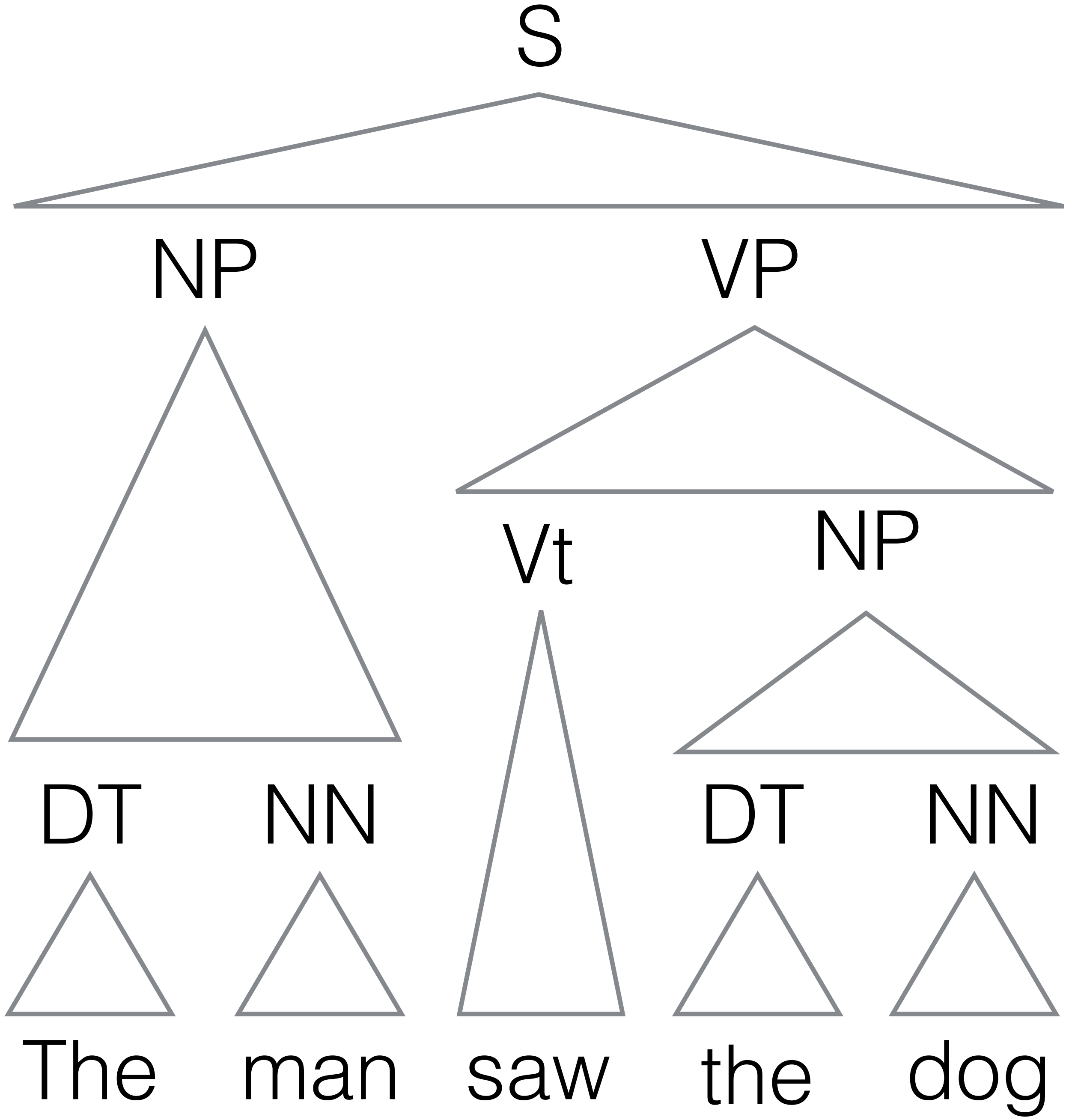
Example of Recognition



Example of Recognition



Example of Recognition



Language

A string $\mathbf{s} = s_1 \ldots s_n$ is generated/accepted by G if

$$S \Rightarrow^* \mathbf{s}$$

$\Rightarrow^*$ denotes a sequence of rule applications

Language of G

$$L(G) = \{\mathbf{s}: S \Rightarrow^* \mathbf{s}\} \subseteq \Sigma^*$$

Chomsky Normal Form

Every CFG is weakly equivalent to another such that

- $X \rightarrow Y\ Z$ where $X, Y, Z \in N$
- $X \rightarrow w$ where $w \in \Sigma$
- and possibly $S \rightarrow \varepsilon$

[Hopcroft and Ullman, 1979]

Parsing as Deduction

Parsing as Deduction

Deductive process to prove claims about grammaticality

[Shieber et al., 1995]

Parsing as Deduction

Deductive process to prove claims about grammaticality

[Shieber et al., 1995]

- focus on strategy rather than implementation

Parsing as Deduction

Deductive process to prove claims about grammaticality
[Shieber et al., 1995]

- focus on strategy rather than implementation
- soundness/completeness easier to prove

Parsing as Deduction

Deductive process to prove claims about grammaticality
[Shieber et al., 1995]

- focus on strategy rather than implementation
- soundness/completeness easier to prove
- complexity determined by inspection

Parsing as Deduction

Deductive process to prove claims about grammaticality
[Shieber et al., 1995]

- focus on strategy rather than implementation
- soundness/completeness easier to prove
- complexity determined by inspection
- dynamic program follows directly

Parsing as Deduction

Deductive process to prove claims about grammaticality
[Shieber et al., 1995]

- focus on strategy rather than implementation
- soundness/completeness easier to prove
- complexity determined by inspection
- dynamic program follows directly
- generality

Deductive systems

Item: a statement / intermediate sound result

- formula or schemata expressed with variables

Inference rule: statement derived from existing items

- $\frac{A_1 \ldots A_m}{B}$ (condition) where A_i and B are items
    - Ai are called antecedents
    - B is called consequent

Deductive program

Axioms: trivial items

- do not depend on previous statements

Goal: states that a proof exists

Proof:

- start from axioms
- exhaustively deduce items
    - never twice under the same premises
- accept if goal is proven

Shift-Reduce Example

Input: *the man sleeps*

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition	Statement	Queue

S → NP VP

VP → Vi

VP → Vt NP

VP → VP PP

NP → DT NN

NP → NP PP

PP → IN NP

Vi → sleeps

Vt → saw

NN → man

NN → dog

NN → telescope

DT → the

IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition	Statement	Queue
Axiom	1	[•,0]	1

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition	Statement	Queue
Axiom	1	[•,0]	1

S → NP VP

VP → Vi

VP → Vt NP

VP → VP PP

NP → DT NN

NP → NP PP

PP → IN NP

Vi → sleeps

Vt → saw

NN → man

NN → dog

NN → telescope

DT → the

IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2

S → NP VP

VP → Vi

VP → Vt NP

VP → VP PP

NP → DT NN

NP → NP PP

PP → IN NP

Vi → sleeps

Vt → saw

NN → man

NN → dog

NN → telescope

DT → the

IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition	Statement	Queue
Axiom	1	[•,0]	1
Shift: [1]	2	[the•,1]	2

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition	Statement		Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3
Shift: [3]		4	[DT man •, 2]	4

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3
Shift: [3]		4	[DT man •, 2]	4

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3
Shift: [3]		4	[DT man •, 2]	4
Reduce: [4]	NN → man	5	[DT NN •, 2]	5

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3
Shift: [3]		4	[DT man •, 2]	4
Reduce: [4]	NN → man	5	[DT NN •, 2]	5

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3
Shift: [3]		4	[DT man •, 2]	4
Reduce: [4]	NN → man	5	[DT NN •, 2]	5
Reduce: [5]	NP → DT NN	6	[NP •, 2]	6

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3
Shift: [3]		4	[DT man •, 2]	4
Reduce: [4]	NN → man	5	[DT NN •, 2]	5
Reduce: [5]	NP → DT NN	6	[NP •, 2]	6

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3
Shift: [3]		4	[DT man •, 2]	4
Reduce: [4]	NN → man	5	[DT NN •, 2]	5
Reduce: [5]	NP → DT NN	6	[NP •, 2]	6
Shift: [6]		7	[NP sleeps •, 3]	7

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3
Shift: [3]		4	[DT man •, 2]	4
Reduce: [4]	NN → man	5	[DT NN •, 2]	5
Reduce: [5]	NP → DT NN	6	[NP •, 2]	6
Shift: [6]		7	[NP sleeps •, 3]	7

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3
Shift: [3]		4	[DT man •, 2]	4
Reduce: [4]	NN → man	5	[DT NN •, 2]	5
Reduce: [5]	NP → DT NN	6	[NP •, 2]	6
Shift: [6]		7	[NP sleeps •, 3]	7
Reduce: [7]	Vi → sleeps	8	[NP Vi •, 3]	8

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3
Shift: [3]		4	[DT man •, 2]	4
Reduce: [4]	NN → man	5	[DT NN •, 2]	5
Reduce: [5]	NP → DT NN	6	[NP •, 2]	6
Shift: [6]		7	[NP sleeps •, 3]	7
Reduce: [7]	Vi → sleeps	8	[NP Vi •, 3]	8

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3
Shift: [3]		4	[DT man •, 2]	4
Reduce: [4]	NN → man	5	[DT NN •, 2]	5
Reduce: [5]	NP → DT NN	6	[NP •, 2]	6
Shift: [6]		7	[NP sleeps •, 3]	7
Reduce: [7]	Vi → sleeps	8	[NP Vi •, 3]	8
Reduce: [8]	VP → Vi	9	[NP VP •, 3]	9

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3
Shift: [3]		4	[DT man •, 2]	4
Reduce: [4]	NN → man	5	[DT NN •, 2]	5
Reduce: [5]	NP → DT NN	6	[NP •, 2]	6
Shift: [6]		7	[NP sleeps •, 3]	7
Reduce: [7]	Vi → sleeps	8	[NP Vi •, 3]	8
Reduce: [8]	VP → Vi	9	[NP VP •, 3]	9

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3
Shift: [3]		4	[DT man •, 2]	4
Reduce: [4]	NN → man	5	[DT NN •, 2]	5
Reduce: [5]	NP → DT NN	6	[NP •, 2]	6
Shift: [6]		7	[NP sleeps •, 3]	7
Reduce: [7]	Vi → sleeps	8	[NP Vi •, 3]	8
Reduce: [8]	VP → Vi	9	[NP VP •, 3]	9
Reduce: [9]	S → NP VP	10	[S •, 3]	10

S → NP VP
VP → Vi
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Shift-Reduce Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[•,0]	1
Shift: [1]		2	[the•,1]	2
Reduce: [2]	DT → the	3	[DT•,1]	3
Shift: [3]		4	[DT man •, 2]	4
Reduce: [4]	NN → man	5	[DT NN •, 2]	5
Reduce: [5]	NP → DT NN	6	[NP •, 2]	6
Shift: [6]		7	[NP sleeps •, 3]	7
Reduce: [7]	Vi → sleeps	8	[NP Vi •, 3]	8
Reduce: [8]	VP → Vi	9	[NP VP •, 3]	9
Reduce: [9]	S → NP VP	10	[S •, 3]	10
GOAL: [10]				∅

S → NP VP ◀
VP → Vi ◀
VP → Vt NP
VP → VP PP
NP → DT NN ◀
NP → NP PP
PP → IN NP
Vi → sleeps ◀
Vt → saw
NN → man ◀
NN → dog
NN → telescope
DT → the ◀
IN → with

Shift-Reduce

Input: G and $w_1 \ldots w_n$

Item form: [α•, j]
asserts that $\alpha \Rightarrow^* w_1 \ldots w_j$ or

that $\alpha\, w_{j+1} \ldots w_n \Rightarrow^* w_1 \ldots w_j$

Axiom: [•,0]

Goal: [S•,n]

Scan (shift)
asserts that $\alpha\, w_{j+1} \Rightarrow^* w_1 \ldots w_j\, w_{j+1}$

$$\textsc{Shift} \quad \frac{[\alpha \bullet, j]}{[\alpha\, w_{j+1}, j+1]}$$

Complete (reduce)
asserts that $\alpha B \Rightarrow^* w_1 \ldots w_j$

$$\textsc{Reduce} \quad \frac{[\alpha\, \gamma \bullet, j]}{[\alpha\, B \bullet, j]} \quad B \rightarrow \gamma \in R$$

Top-Down recognition

Input: G and $w_1 \ldots w_n$

Item form: [•β, j]
asserts that $S \Rightarrow^* w_1 \ldots w_j \; \beta$

Axiom: [•S,0]

Goal: [•,n]

Scan
asserts that $S \Rightarrow^* w_1 \ldots w_j \, w_{j+1} \, \beta$

$$\textsc{Scan} \quad \frac{[\bullet w_{j+1}\beta, j]}{[\bullet\beta, j+1]}$$

Predict
asserts that $S \Rightarrow^* w_1 \ldots w_j \, B \, \beta$

$$\textsc{Predict} \quad \frac{\bullet B \ \beta, j}{[\bullet\gamma \ \beta, j]} \quad B \rightarrow \gamma \in R$$

Top-Down Example

Input: *the man sleeps*

S → NP VP

VP → Vi

VP → Vt NP

~~VP → VP PP~~

NP → DT NN

NP → NP PP

PP → IN NP

Vi → sleeps

Vt → saw

NN → man

NN → dog

NN → telescope

DT → the

IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition	Statement	Queue

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition	Statement	Queue
Axiom	1	[• S, 0]	1

S → NP VP

VP → Vi

VP → Vt NP

~~VP → VP PP~~

NP → DT NN

NP → NP PP

PP → IN NP

Vi → sleeps

Vt → saw

NN → man

NN → dog

NN → telescope

DT → the

IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition	Statement	Queue
Axiom	1	[• S, 0]	1

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2

S → NP VP

VP → Vi

VP → Vt NP

~~VP → VP PP~~

NP → DT NN

NP → NP PP

PP → IN NP

Vi → sleeps

Vt → saw

NN → man

NN → dog

NN → telescope

DT → the

IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4

S → NP VP

VP → Vi

VP → Vt NP

~~VP → VP PP~~

NP → DT NN

NP → NP PP

PP → IN NP

Vi → sleeps

Vt → saw

NN → man

NN → dog

NN → telescope

DT → the

IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5

S → NP VP

VP → Vi

VP → Vt NP

~~VP → VP PP~~

NP → DT NN

NP → NP PP

PP → IN NP

Vi → sleeps

Vt → saw

NN → man

NN → dog

NN → telescope

DT → the

IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5
Predict: [5]	NN → man	6	[• man VP, 1]	6

S → NP VP

VP → Vi

VP → Vt NP

~~VP → VP PP~~

NP → DT NN

NP → NP PP

PP → IN NP

Vi → sleeps

Vt → saw

NN → man

NN → dog

NN → telescope

DT → the

IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5
Predict: [5]	NN → man	6	[• man VP, 1]	6

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5
Predict: [5]	NN → man	6	[• man VP, 1]	6
Scan: [6]		7	[• VP, 2]	7

S → NP VP

VP → Vi

VP → Vt NP

~~VP → VP PP~~

NP → DT NN

NP → NP PP

PP → IN NP

Vi → sleeps

Vt → saw

NN → man

NN → dog

NN → telescope

DT → the

IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5
Predict: [5]	NN → man	6	[• man VP, 1]	6
Scan: [6]		7	[• VP, 2]	7

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5
Predict: [5]	NN → man	6	[• man VP, 1]	6
Scan: [6]		7	[• VP, 2]	7

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5
Predict: [5]	NN → man	6	[• man VP, 1]	6
Scan: [6]		7	[• VP, 2]	7
Predict: [7]	VP → Vi	8	[• Vi, 2]	8, 9

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5
Predict: [5]	NN → man	6	[• man VP, 1]	6
Scan: [6]		7	[• VP, 2]	7
Predict: [7]	VP → Vi	8	[• Vi, 2]	8, 9
	VP → Vt NP	9	[• Vt NP, 2]	

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5
Predict: [5]	NN → man	6	[• man VP, 1]	6
Scan: [6]		7	[• VP, 2]	7
Predict: [7]	VP → Vi	8	[• Vi, 2]	8, 9
	VP → Vt NP	9	[• Vt NP, 2]	

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5
Predict: [5]	NN → man	6	[• man VP, 1]	6
Scan: [6]		7	[• VP, 2]	7
Predict: [7]	VP → Vi	8	[• Vi, 2]	8, 9
	VP → Vt NP	9	[• Vt NP, 2]	
Predict: [8]	Vi → sleeps	10	[• sleeps, 2]	9, 10

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5
Predict: [5]	NN → man	6	[• man VP, 1]	6
Scan: [6]		7	[• VP, 2]	7
Predict: [7]	VP → Vi	8	[• Vi, 2]	8, 9
	VP → Vt NP	9	[• Vt NP, 2]	
Predict: [8]	Vi → sleeps	10	[• sleeps, 2]	9, 10

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5
Predict: [5]	NN → man	6	[• man VP, 1]	6
Scan: [6]		7	[• VP, 2]	7
Predict: [7]	VP → Vi	8	[• Vi, 2]	8, 9
	VP → Vt NP	9	[• Vt NP, 2]	
Predict: [8]	Vi → sleeps	10	[• sleeps, 2]	9, 10
[9]				10

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5
Predict: [5]	NN → man	6	[• man VP, 1]	6
Scan: [6]		7	[• VP, 2]	7
Predict: [7]	VP → Vi	8	[• Vi, 2]	8, 9
	VP → Vt NP	9	[• Vt NP, 2]	
Predict: [8]	Vi → sleeps	10	[• sleeps, 2]	9, 10
[9]				10

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5
Predict: [5]	NN → man	6	[• man VP, 1]	6
Scan: [6]		7	[• VP, 2]	7
Predict: [7]	VP → Vi	8	[• Vi, 2]	8, 9
	VP → Vt NP	9	[• Vt NP, 2]	
Predict: [8]	Vi → sleeps	10	[• sleeps, 2]	9, 10
[9]				10
Scan: [10]		11	[•, 3]	11

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Top-Down Example

Input: *the man sleeps*

Rule	Condition		Statement	Queue
Axiom		1	[• S, 0]	1
Predict: [1]	S → NP VP	2	[• NP VP, 0]	2
Predict: [2]	NP → DT NN	3	[• DT NN VP, 0]	3
Predict: [3]	DT → the	4	[• the NN VP,0]	4
Scan: [4]		5	[• NN VP,1]	5
Predict: [5]	NN → man	6	[• man VP, 1]	6
Scan: [6]		7	[• VP, 2]	7
Predict: [7]	VP → Vi	8	[• Vi, 2]	8, 9
	VP → Vt NP	9	[• Vt NP, 2]	
Predict: [8]	Vi → sleeps	10	[• sleeps, 2]	9, 10
[9]				10
Scan: [10]		11	[•, 3]	11
GOAL: [11]				∅

S → NP VP
VP → Vi
VP → Vt NP
~~VP → VP PP~~
NP → DT NN
NP → NP PP
PP → IN NP
Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

CKY - CNF only

Input: G and *s* = $w_1 \dots w_n$

Item form: $[i, X, j]$
asserts that $X \Rightarrow^* w_{i+1} \dots w_j$

Axioms: $[i, X, i+1]$ $X \rightarrow w_i \in R$

Goal: $[0, S, n]$

Merge:
asserts that
$w_{i+1} \dots w_k w_{k+1} \dots w_j \Rightarrow^* w_{i+1} \dots w_j$

$$\frac{[i, A, k]\ [k, B, j]}{[i, C, j]} \quad C \rightarrow A\, B \in R$$

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Statement	Queue	Passive

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Statement	Queue	Passive
Axiom	DT → the 1	[0, DT, 1] 1		

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Statement	Queue	Passive
Axiom	DT → the	1 [0, DT, 1]	1	
	NN → man	2 [1, NN, 2]	1, 2	

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Statement		Queue	Passive
Axiom	DT → the	1	[0, DT, 1]	1	
	NN → man	2	[1, NN, 2]	1, 2	
	Vt → saw	3	[2, Vt, 3]	1, 2, 3	

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Statement	Queue	Passive
Axiom	DT → the	1 [0, DT, 1]	1	
	NN → man	2 [1, NN, 2]	1, 2	
	Vt → saw	3 [2, Vt, 3]	1, 2, 3	
	DT → the	4 [3, DT, 4]	1, 2, 3, 4	

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Statement	Queue	Passive
Axiom	DT → the	1 [0, DT, 1]	1	
	NN → man	2 [1, NN, 2]	1, 2	
	Vt → saw	3 [2, Vt, 3]	1, 2, 3	
	DT → the	4 [3, DT, 4]	1, 2, 3, 4	
	NN → dog	5 [4, NN, 5]	1, 2, 3, 4, 5	

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Statement		Queue	Passive
Axiom	DT → the	1	[0, DT, 1]	1	
	NN → man	2	[1, NN, 2]	1, 2	
	Vt → saw	3	[2, Vt, 3]	1, 2, 3	
	DT → the	4	[3, DT, 4]	1, 2, 3, 4	
	NN → dog	5	[4, NN, 5]	1, 2, 3, 4, 5	
				2, 3, 4, 5	1

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Statement		Queue	Passive
Axiom	DT → the	1	[0, DT, 1]	1	
	NN → man	2	[1, NN, 2]	1, 2	
	Vt → saw	3	[2, Vt, 3]	1, 2, 3	
	DT → the	4	[3, DT, 4]	1, 2, 3, 4	
	NN → dog	5	[4, NN, 5]	1, 2, 3, 4, 5	
				2, 3, 4, 5	1
Merge: [1][2]	NP → DT NN	6	[0, NP, 2]	3, 4, 5, 6	2

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Statement	Queue	Passive
Axiom	DT → the	1 [0, DT, 1]	1	
	NN → man	2 [1, NN, 2]	1, 2	
	Vt → saw	3 [2, Vt, 3]	1, 2, 3	
	DT → the	4 [3, DT, 4]	1, 2, 3, 4	
	NN → dog	5 [4, NN, 5]	1, 2, 3, 4, 5	
			2, 3, 4, 5	1
Merge: [1][2]	NP → DT NN	6 [0, NP, 2]	3, 4, 5, 6	2
			4, 5, 6	3

CKY Example

Input: *the man saw the dog*

S → NP VP
~~VP → Vi~~
VP → Vt NP
VP → VP PP
NP → DT NN
NP → NP PP
PP → IN NP

Vi → sleeps
Vt → saw
NN → man
NN → dog
NN → telescope
DT → the
IN → with

Rule	Condition	Statement		Queue	Passive
Axiom	DT → the	1	[0, DT, 1]	1	
	NN → man	2	[1, NN, 2]	1, 2	
	Vt → saw	3	[2, Vt, 3]	1, 2, 3	
	DT → the	4	[3, DT, 4]	1, 2, 3, 4	
	NN → dog	5	[4, NN, 5]	1, 2, 3, 4, 5	
				2, 3, 4, 5	1
Merge: [1][2]	NP → DT NN	6	[0, NP, 2]	3, 4, 5, 6	2
				4, 5, 6	3
				5, 6	4

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Statement		Queue	Passive
Axiom	DT → the	1	[0, DT, 1]	1	
	NN → man	2	[1, NN, 2]	1, 2	
	Vt → saw	3	[2, Vt, 3]	1, 2, 3	
	DT → the	4	[3, DT, 4]	1, 2, 3, 4	
	NN → dog	5	[4, NN, 5]	1, 2, 3, 4, 5	
				2, 3, 4, 5	1
Merge: [1][2]	NP → DT NN	6	[0, NP, 2]	3, 4, 5, 6	2
				4, 5, 6	3
				5, 6	4
Merge: [4][5]	NP → DT NN	7	[3, NP, 5]	6, 7	5

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Statement		Queue	Passive
Axiom	DT → the	1	[0, DT, 1]	1	
	NN → man	2	[1, NN, 2]	1, 2	
	Vt → saw	3	[2, Vt, 3]	1, 2, 3	
	DT → the	4	[3, DT, 4]	1, 2, 3, 4	
	NN → dog	5	[4, NN, 5]	1, 2, 3, 4, 5	
				2, 3, 4, 5	1
Merge: [1][2]	NP → DT NN	6	[0, NP, 2]	3, 4, 5, 6	2
				4, 5, 6	3
				5, 6	4
Merge: [4][5]	NP → DT NN	7	[3, NP, 5]	6, 7	5
				7	6

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Statement		Queue	Passive
Axiom	DT → the	1	[0, DT, 1]	1	
	NN → man	2	[1, NN, 2]	1, 2	
	Vt → saw	3	[2, Vt, 3]	1, 2, 3	
	DT → the	4	[3, DT, 4]	1, 2, 3, 4	
	NN → dog	5	[4, NN, 5]	1, 2, 3, 4, 5	
				2, 3, 4, 5	1
Merge: [1][2]	NP → DT NN	6	[0, NP, 2]	3, 4, 5, 6	2
				4, 5, 6	3
				5, 6	4
Merge: [4][5]	NP → DT NN	7	[3, NP, 5]	6, 7	5
				7	6
Merge: [3] [7]	VP → Vt NP	8	[2, VP, 5]	8	7

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Statement		Queue	Passive
Axiom	DT → the	1	[0, DT, 1]	1	
	NN → man	2	[1, NN, 2]	1, 2	
	Vt → saw	3	[2, Vt, 3]	1, 2, 3	
	DT → the	4	[3, DT, 4]	1, 2, 3, 4	
	NN → dog	5	[4, NN, 5]	1, 2, 3, 4, 5	
				2, 3, 4, 5	1
Merge: [1][2]	NP → DT NN	6	[0, NP, 2]	3, 4, 5, 6	2
				4, 5, 6	3
				5, 6	4
Merge: [4][5]	NP → DT NN	7	[3, NP, 5]	6, 7	5
				7	6
Merge: [3] [7]	VP → Vt NP	8	[2, VP, 5]	8	7
Merge: [6] [8]	S → NP VP	9	[0, S, 5]	9	8

CKY Example

Input: *the man saw the dog*

S → NP VP	Vi → sleeps
~~VP → Vi~~	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Statement		Queue	Passive
Axiom	DT → the	1	[0, DT, 1]	1	
	NN → man	2	[1, NN, 2]	1, 2	
	Vt → saw	3	[2, Vt, 3]	1, 2, 3	
	DT → the	4	[3, DT, 4]	1, 2, 3, 4	
	NN → dog	5	[4, NN, 5]	1, 2, 3, 4, 5	
				2, 3, 4, 5	1
Merge: [1][2]	NP → DT NN	6	[0, NP, 2]	3, 4, 5, 6	2
				4, 5, 6	3
				5, 6	4
Merge: [4][5]	NP → DT NN	7	[3, NP, 5]	6, 7	5
				7	6
Merge: [3] [7]	VP → Vt NP	8	[2, VP, 5]	8	7
Merge: [6] [8]	S → NP VP	9	[0, S, 5]	9	8
GOAL: [9]				∅	9

Rule Segmentation: "Split Points"

${}_0\mathrm{S}_3 \rightarrow {}_0\mathrm{NP}_2\ {}_2\mathrm{VP}_3$

${}_0\mathrm{NP}_2 \rightarrow {}_0\mathrm{DT}_1\ {}_1\mathrm{NN}_2$

${}_2\mathrm{VP}_3 \rightarrow {}_2\mathrm{Vi}_3$

${}_0\mathrm{DT}_1 \rightarrow \text{the}$

${}_1\mathrm{NN}_2 \rightarrow \text{man}$

${}_2\mathrm{Vi}_3 \rightarrow \text{sleeps}$

0 1 2 3

Rule Segmentation: "Split Points"

$_0S_3 \rightarrow {_0NP_2}\ {_2VP_3}$

$_0NP_2 \rightarrow {_0DT_1}\ {_1NN_2}$

$_2VP_3 \rightarrow {_2Vi_3}$

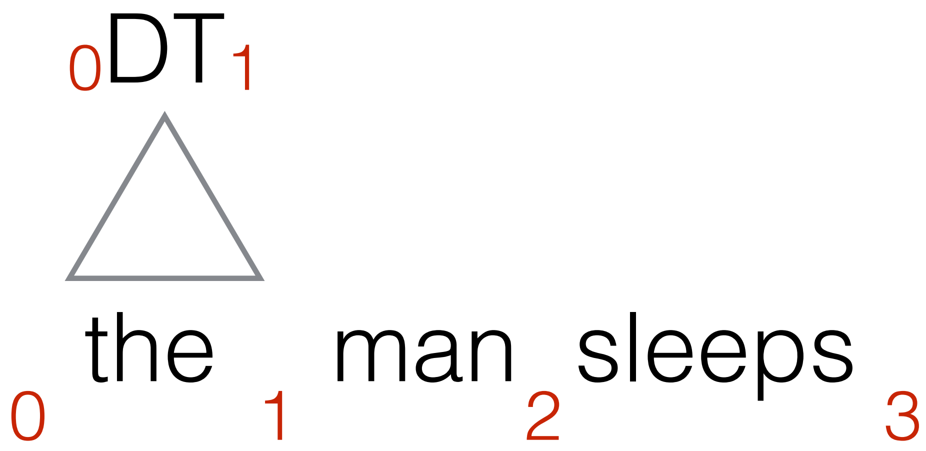
$_0DT_1 \rightarrow$ the

$_1NN_2 \rightarrow$ man

$_2Vi_3 \rightarrow$ sleeps

$_0$ the $_1$ man $_2$ sleeps $_3$

Rule Segmentation: "Split Points"



${}_0S_3 \rightarrow {}_0NP_2\ {}_2VP_3$

${}_0NP_2 \rightarrow {}_0DT_1\ {}_1NN_2$

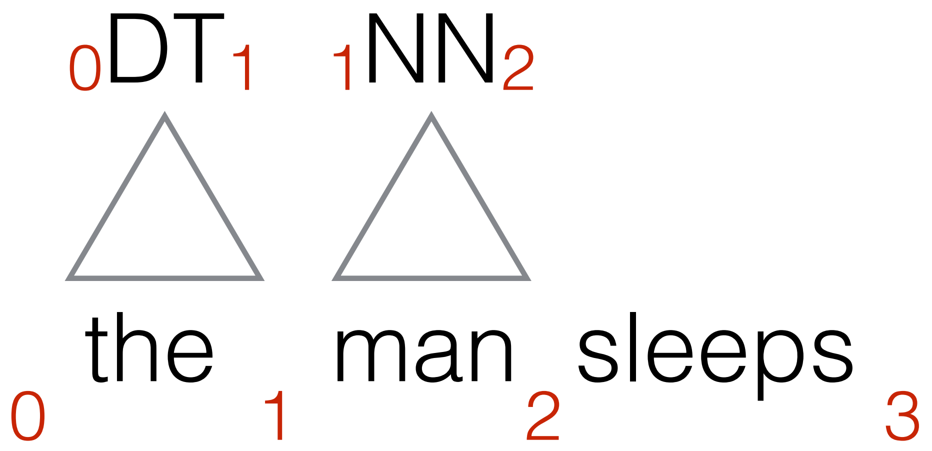
${}_2VP_3 \rightarrow {}_2Vi_3$

${}_0DT_1 \rightarrow \text{the}$

${}_1NN_2 \rightarrow \text{man}$

${}_2Vi_3 \rightarrow \text{sleeps}$

Rule Segmentation: "Split Points"



${}_0S_3 \rightarrow {}_0NP_2\ {}_2VP_3$

${}_0NP_2 \rightarrow {}_0DT_1\ {}_1NN_2$

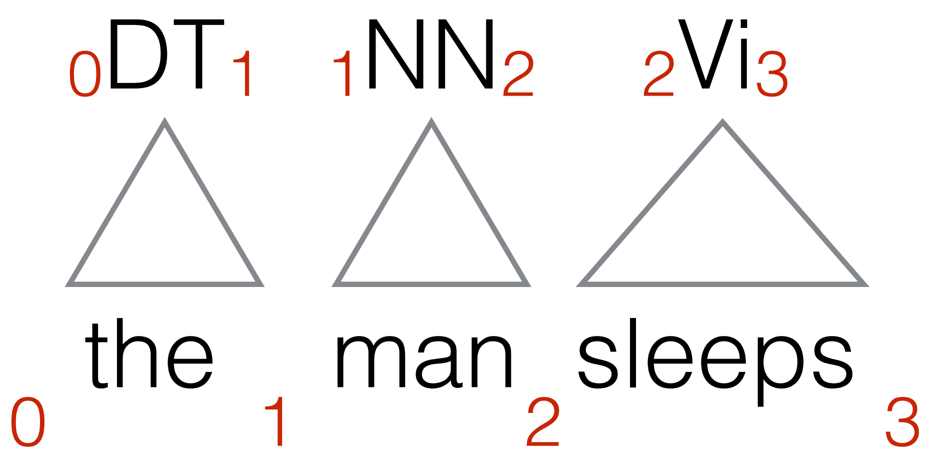
${}_2VP_3 \rightarrow {}_2Vi_3$

${}_0DT_1 \rightarrow$ the

${}_1NN_2 \rightarrow$ man

${}_2Vi_3 \rightarrow$ sleeps

Rule Segmentation: "Split Points"



${}_0\mathrm{DT}_1 \quad {}_1\mathrm{NN}_2 \quad {}_2\mathrm{Vi}_3$

${}_0$ the ${}_1$ man ${}_2$ sleeps ${}_3$

${}_0\mathrm{S}_3 \rightarrow {}_0\mathrm{NP}_2\ {}_2\mathrm{VP}_3$

${}_0\mathrm{NP}_2 \rightarrow {}_0\mathrm{DT}_1\ {}_1\mathrm{NN}_2$

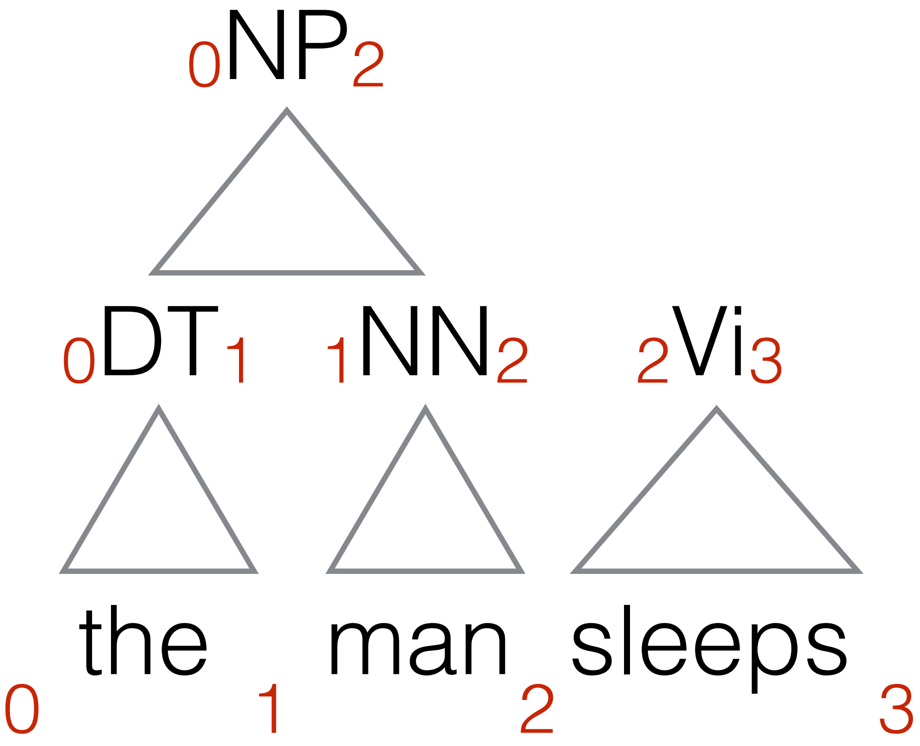
${}_2\mathrm{VP}_3 \rightarrow {}_2\mathrm{Vi}_3$

${}_0\mathrm{DT}_1 \rightarrow$ the

${}_1\mathrm{NN}_2 \rightarrow$ man

${}_2\mathrm{Vi}_3 \rightarrow$ sleeps

Rule Segmentation: "Split Points"



${}_0\mathrm{S}_3 \rightarrow {}_0\mathrm{NP}_2\ {}_2\mathrm{VP}_3$

${}_0\mathrm{NP}_2 \rightarrow {}_0\mathrm{DT}_1\ {}_1\mathrm{NN}_2$

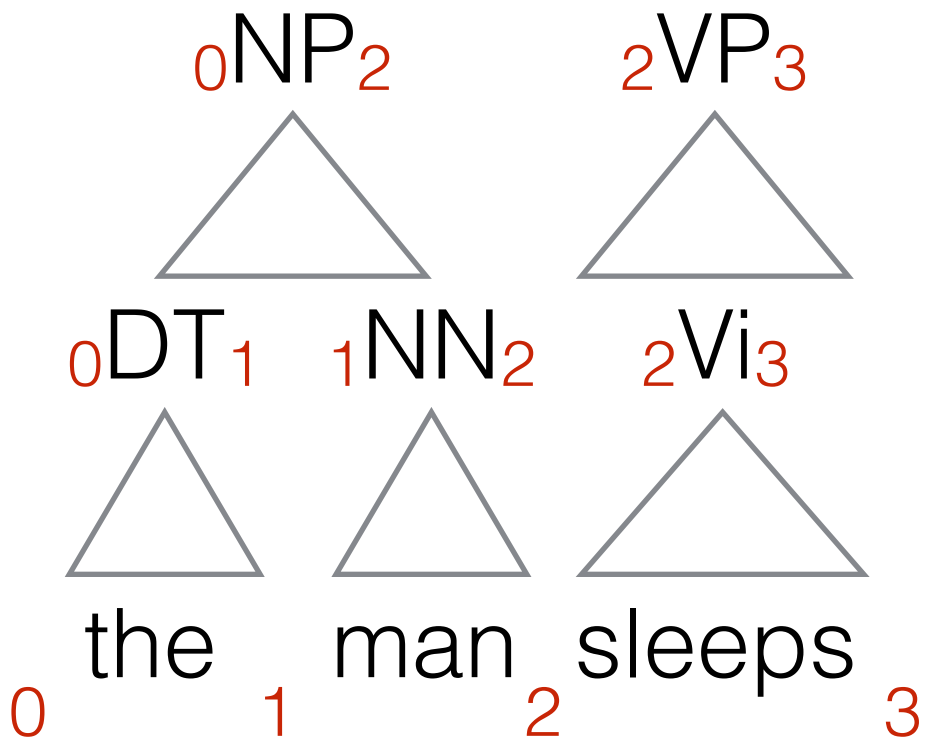
${}_2\mathrm{VP}_3 \rightarrow {}_2\mathrm{Vi}_3$

${}_0\mathrm{DT}_1 \rightarrow$ the

${}_1\mathrm{NN}_2 \rightarrow$ man

${}_2\mathrm{Vi}_3 \rightarrow$ sleeps

Rule Segmentation: "Split Points"



${}_0NP_2$ ${}_2VP_3$

${}_0DT_1$ ${}_1NN_2$ ${}_2Vi_3$

the man sleeps

0 1 2 3

${}_0S_3 \rightarrow {}_0NP_2\ {}_2VP_3$

${}_0NP_2 \rightarrow {}_0DT_1\ {}_1NN_2$

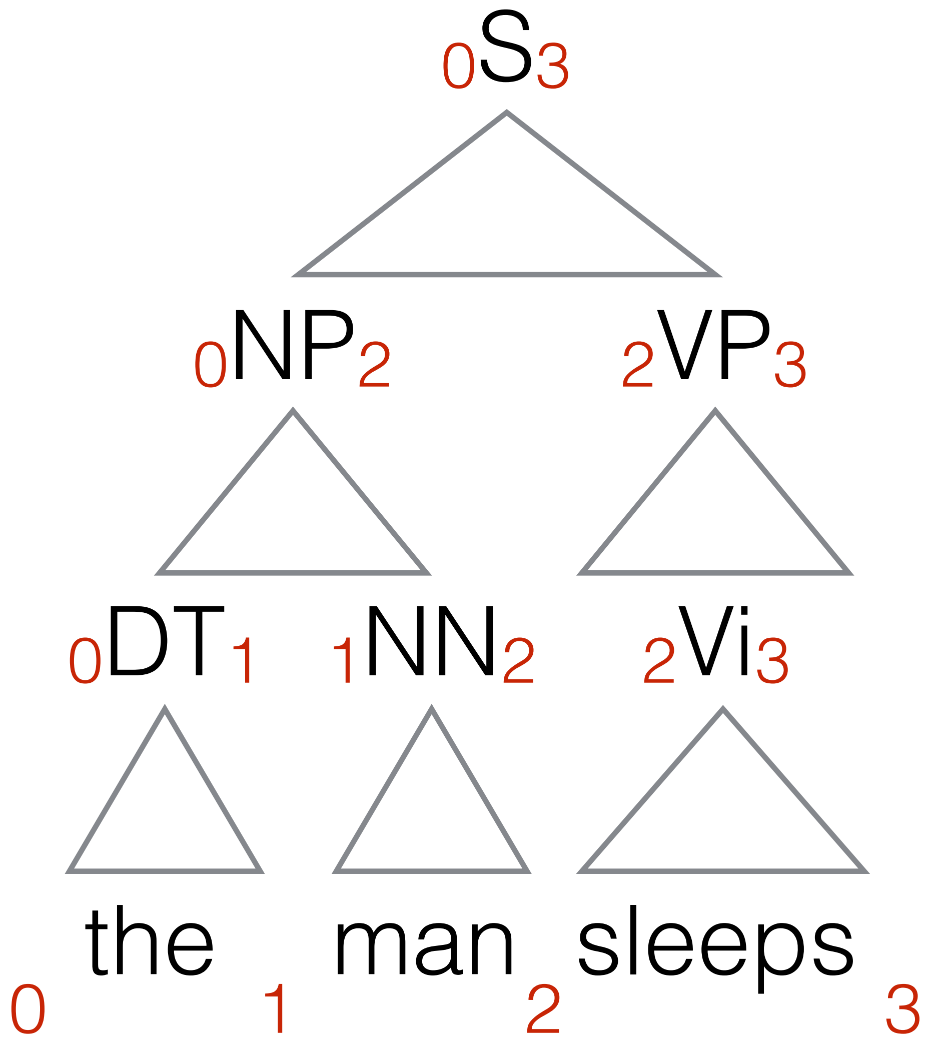
${}_2VP_3 \rightarrow {}_2Vi_3$

${}_0DT_1 \rightarrow$ the

${}_1NN_2 \rightarrow$ man

${}_2Vi_3 \rightarrow$ sleeps

Rule Segmentation: "Split Points"



$_0S_3 \rightarrow {}_0NP_2\ {}_2VP_3$

$_0NP_2 \rightarrow {}_0DT_1\ {}_1NN_2$

$_2VP_3 \rightarrow {}_2Vi_3$

$_0DT_1 \rightarrow$ the

$_1NN_2 \rightarrow$ man

$_2Vi_3 \rightarrow$ sleeps

"Dotted items"

"Dotted items"

Parsing a CNF grammar is easy because we know the shape of rules

"Dotted items"

Parsing a CNF grammar is easy because we know the shape of rules

When that's not the case, we have to **scan rules symbol by symbol** using a general mechanism:

"Dotted items"

Parsing a CNF grammar is easy because we know the shape of rules

When that's not the case, we have to **scan rules symbol by symbol** using a general mechanism:

Item form: $[i, X \to \alpha_{\blacksquare} \bullet \beta_{\square}, j]$ where $X \to \alpha\ \beta \in R$ is a rule

"Dotted items"

Parsing a CNF grammar is easy because we know the shape of rules

When that's not the case, we have to **scan rules symbol by symbol** using a general mechanism:

Item form: $[i, X \to \alpha_{\blacksquare} \bullet \beta_{\square}, j]$ where $X \to \alpha\ \beta \in R$ is a rule

- In general, we segment rules with respect to the input w_1 ... w_n

"Dotted items"

Parsing a CNF grammar is easy because we know the shape of rules

When that's not the case, we have to **scan rules symbol by symbol** using a general mechanism:

Item form: $[i, X \rightarrow \alpha_{\blacksquare} \bullet \beta_{\square}, j]$ where $X \rightarrow \alpha\ \beta \in R$ is a rule

- In general, we segment rules with respect to the input w_1 ... w_n
- The dot represents progress through the rule's right-hand side (RHS)

"Dotted items"

Parsing a CNF grammar is easy because we know the shape of rules

When that's not the case, we have to **scan rules symbol by symbol** using a general mechanism:

Item form: $[i, X \to \alpha_{\blacksquare} \bullet \beta_{\square}, j]$ where $X \to \alpha\ \beta \in R$ is a rule

- In general, we segment rules with respect to the input $w_1 \ldots w_n$
- The dot represents progress through the rule's right-hand side (RHS)
- The prefix α has already been parsed and we are waiting for β

"Dotted items"

Parsing a CNF grammar is easy because we know the shape of rules

When that's not the case, we have to **scan rules symbol by symbol** using a general mechanism:

Item form: $[i, X \rightarrow \alpha_{\blacksquare} \bullet \beta_{\square}, j]$ where $X \rightarrow \alpha\ \beta \in R$ is a rule

- In general, we segment rules with respect to the input $w_1 \ldots w_n$
- The dot represents progress through the rule's right-hand side (RHS)
- The prefix α has already been parsed and we are waiting for β
- The filled box represents a segmentation of $[0 .. j]$ into $|\alpha|$ adjacent parts

"Dotted items"

Parsing a CNF grammar is easy because we know the shape of rules

When that's not the case, we have to **scan rules symbol by symbol** using a general mechanism:

Item form: [i, $X \rightarrow \alpha_{\blacksquare} \bullet \beta_{\square}$, j] where $X \rightarrow \alpha\ \beta \in R$ is a rule

- In general, we segment rules with respect to the input w_1 ... w_n
- The dot represents progress through the rule's right-hand side (RHS)
- The prefix α has already been parsed and we are waiting for β
- The filled box represents a segmentation of [0 .. j] into |α| adjacent parts
- The empty box has no actual role, it's just a reminder that the segmentation beyond j is unknown

CKY+

Input: G and *s* = $w_1 \ldots w_n$

Item form: $[i, X \to \alpha_\blacksquare \bullet \beta_\square, j]$
asserts that $X \Rightarrow^* w_{i+1} \ldots w_j\, \beta$

Axioms: $[i, X \to w_i \bullet \alpha_\square, i+1]$ $X \to w_i\, \alpha \in R$

$[i, X \to \varepsilon \bullet, i]$ $X \to \varepsilon \in R$

Goal: $[0, S \to \alpha_\blacksquare \bullet, n]$

Scan

$$\frac{[i, X \to \alpha_\blacksquare \bullet w_{j+1}\,\beta_\square, j]}{[i, X \to \alpha_\blacksquare w_{j+1} \bullet \beta_\square, j+1]}$$

Prefix

$$\frac{[i, Y \to \alpha_\blacksquare \bullet, j]}{[i, X \to Y_{i,j} \bullet \beta_\square, j]} \quad X \to Y\beta \in R$$

Complete

$$\frac{[i, X \to \alpha_\blacksquare \bullet Y\beta_\square, k]\,[k, Y \to \gamma_\blacksquare \bullet, j]}{[i, X \to \alpha_\blacksquare\, Y_{k,j} \bullet \beta_\square, j]}$$

CKY+ Example

Input: *the man sleeps*

S → NP VP	Vi → sleeps
VP → Vi	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

CKY+ Example

S → NP VP	Vi → sleeps
VP → Vi	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Input: *the man sleeps*

Rule	Condition	Item	Active	Passive

CKY+ Example

Input: *the man sleeps*

S → NP VP	Vi → sleeps
VP → Vi	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition	Item	Active	Passive
Axiom	DT → the 1	[0, DT → the •,1]	1	

CKY+ Example

S → NP VP	Vi → sleeps
VP → Vi	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Input: *the man sleeps*

Rule	Condition		Item	Active	Passive
Axiom	DT → the	1	[0, DT → the •,1]	1	
	NN → man	2	[1, NN → man •, 2]	1, 2	

CKY+ Example

S → NP VP	Vi → sleeps
VP → Vi	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Input: *the man sleeps*

Rule	Condition		Item	Active	Passive
Axiom	DT → the	1	[0, DT → the •,1]	1	
	NN → man	2	[1, NN → man •, 2]	1, 2	
	Vi → sleeps	3	[2, Vi → sleeps •, 3]	1, 2, 3	

CKY+ Example

Input: *the man sleeps*

S → NP VP	Vi → sleeps
VP → Vi	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition		Item	Active	Passive
Axiom	DT → the	1	[0, DT → the •,1]	1	
	NN → man	2	[1, NN → man •, 2]	1, 2	
	Vi → sleeps	3	[2, Vi → sleeps •, 3]	1, 2, 3	
Prefix: [1]	NP → DT NN	4	[0, NP → $\text{DT}_{0,1}$ • NN, 2]	2, 3, 4	1

CKY+ Example

S → NP VP	Vi → sleeps
VP → Vi	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Input: *the man sleeps*

Rule	Condition		Item	Active	Passive
Axiom	DT → the	1	[0, DT → the •,1]	1	
	NN → man	2	[1, NN → man •, 2]	1, 2	
	Vi → sleeps	3	[2, Vi → sleeps •, 3]	1, 2, 3	
Prefix: [1]	NP → DT NN	4	[0, NP → $\text{DT}_{0,1}$ • NN, 2]	2, 3, 4	1
				3, 4	2

CKY+ Example

S → NP VP	Vi → sleeps
VP → Vi	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Input: *the man sleeps*

Rule	Condition		Item	Active	Passive
Axiom	DT → the	1	[0, DT → the •,1]	1	
	NN → man	2	[1, NN → man •, 2]	1, 2	
	Vi → sleeps	3	[2, Vi → sleeps •, 3]	1, 2, 3	
Prefix: [1]	NP → DT NN	4	[0, NP → $\text{DT}_{0,1}$ • NN, 2]	2, 3, 4	1
				3, 4	2
Prefix: [3]	VP → Vi	5	[2, VP → $\text{Vi}_{2,3}$ •, 3]	4, 5	3

CKY+ Example

Input: *the man sleeps*

S → NP VP	Vi → sleeps
VP → Vi	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition		Item	Active	Passive
Axiom	DT → the	1	[0, DT → the •,1]	1	
	NN → man	2	[1, NN → man •, 2]	1, 2	
	Vi → sleeps	3	[2, Vi → sleeps •, 3]	1, 2, 3	
Prefix: [1]	NP → DT NN	4	[0, NP → $\mathrm{DT}_{0,1}$ • NN, 2]	2, 3, 4	1
				3, 4	2
Prefix: [3]	VP → Vi	5	[2, VP → $\mathrm{Vi}_{2,3}$ •, 3]	4, 5	3
Complete: [4] [2]		6	[0, NP → $\mathrm{DT}_{0,1}$ $\mathrm{NN}_{1,2}$ •, 2]	5, 6	4

CKY+ Example

Input: *the man sleeps*

S → NP VP	Vi → sleeps
VP → Vi	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Rule	Condition		Item	Active	Passive
Axiom	DT → the	1	[0, DT → the •,1]	1	
	NN → man	2	[1, NN → man •, 2]	1, 2	
	Vi → sleeps	3	[2, Vi → sleeps •, 3]	1, 2, 3	
Prefix: [1]	NP → DT NN	4	[0, NP → DT$_{0,1}$ • NN, 2]	2, 3, 4	1
				3, 4	2
Prefix: [3]	VP → Vi	5	[2, VP → Vi$_{2,3}$ •, 3]	4, 5	3
Complete: [4] [2]		6	[0, NP → DT$_{0,1}$ NN$_{1,2}$ •, 2]	5, 6	4
				6	5

CKY+ Example

S → NP VP	Vi → sleeps
VP → Vi	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Input: *the man sleeps*

Rule	Condition		Item	Active	Passive
Axiom	DT → the	1	[0, DT → the •,1]	1	
	NN → man	2	[1, NN → man •, 2]	1, 2	
	Vi → sleeps	3	[2, Vi → sleeps •, 3]	1, 2, 3	
Prefix: [1]	NP → DT NN	4	[0, NP → $\mathrm{DT}_{0,1}$ • NN, 2]	2, 3, 4	1
				3, 4	2
Prefix: [3]	VP → Vi	5	[2, VP → $\mathrm{Vi}_{2,3}$ •, 3]	4, 5	3
Complete: [4] [2]		6	[0, NP → $\mathrm{DT}_{0,1}$ $\mathrm{NN}_{1,2}$ •, 2]	5, 6	4
				6	5
Prefix: [6]	S → NP VP	7	[0, S → $\mathrm{NP}_{0,2}$ • VP, 2]	7	6

CKY+ Example

S → NP VP	Vi → sleeps
VP → Vi	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Input: *the man sleeps*

Rule	Condition		Item	Active	Passive
Axiom	DT → the	1	[0, DT → the •,1]	1	
	NN → man	2	[1, NN → man •, 2]	1, 2	
	Vi → sleeps	3	[2, Vi → sleeps •, 3]	1, 2, 3	
Prefix: [1]	NP → DT NN	4	[0, NP → DT$_{0,1}$ • NN, 2]	2, 3, 4	1
				3, 4	2
Prefix: [3]	VP → Vi	5	[2, VP → Vi$_{2,3}$ •, 3]	4, 5	3
Complete: [4] [2]		6	[0, NP → DT$_{0,1}$ NN$_{1,2}$ •, 2]	5, 6	4
				6	5
Prefix: [6]	S → NP VP	7	[0, S → NP$_{0,2}$ • VP, 2]	7	6
Complete: [7] [5]		8	[0, S → NP$_{0,2}$ VP$_{2,3}$ •, 3]	8	7

CKY+ Example

S → NP VP	Vi → sleeps
VP → Vi	Vt → saw
VP → Vt NP	NN → man
VP → VP PP	NN → dog
NP → DT NN	NN → telescope
NP → NP PP	DT → the
PP → IN NP	IN → with

Input: *the man sleeps*

Rule	Condition		Item	Active	Passive
Axiom	DT → the	1	[0, DT → the •,1]	1	
	NN → man	2	[1, NN → man •, 2]	1, 2	
	Vi → sleeps	3	[2, Vi → sleeps •, 3]	1, 2, 3	
Prefix: [1]	NP → DT NN	4	[0, NP → DT$_{0,1}$ • NN, 2]	2, 3, 4	1
				3, 4	2
Prefix: [3]	VP → Vi	5	[2, VP → Vi$_{2,3}$ •, 3]	4, 5	3
Complete: [4] [2]		6	[0, NP → DT$_{0,1}$ NN$_{1,2}$ •, 2]	5, 6	4
				6	5
Prefix: [6]	S → NP VP	7	[0, S → NP$_{0,2}$ • VP, 2]	7	6
Complete: [7] [5]		8	[0, S → NP$_{0,2}$ VP$_{2,3}$ •, 3]	8	7
GOAL: [8]				∅	

Correctness of Parsing Strategy

Soundness: if a goal item is proven for **s**

- then **s** ∈ L(G)

Completeness: if **s** ∈ L(G)

- then a goal item can be proven for **s**

Parse Forest

Efficient representation of the whole space $T_G(\boldsymbol{s})$

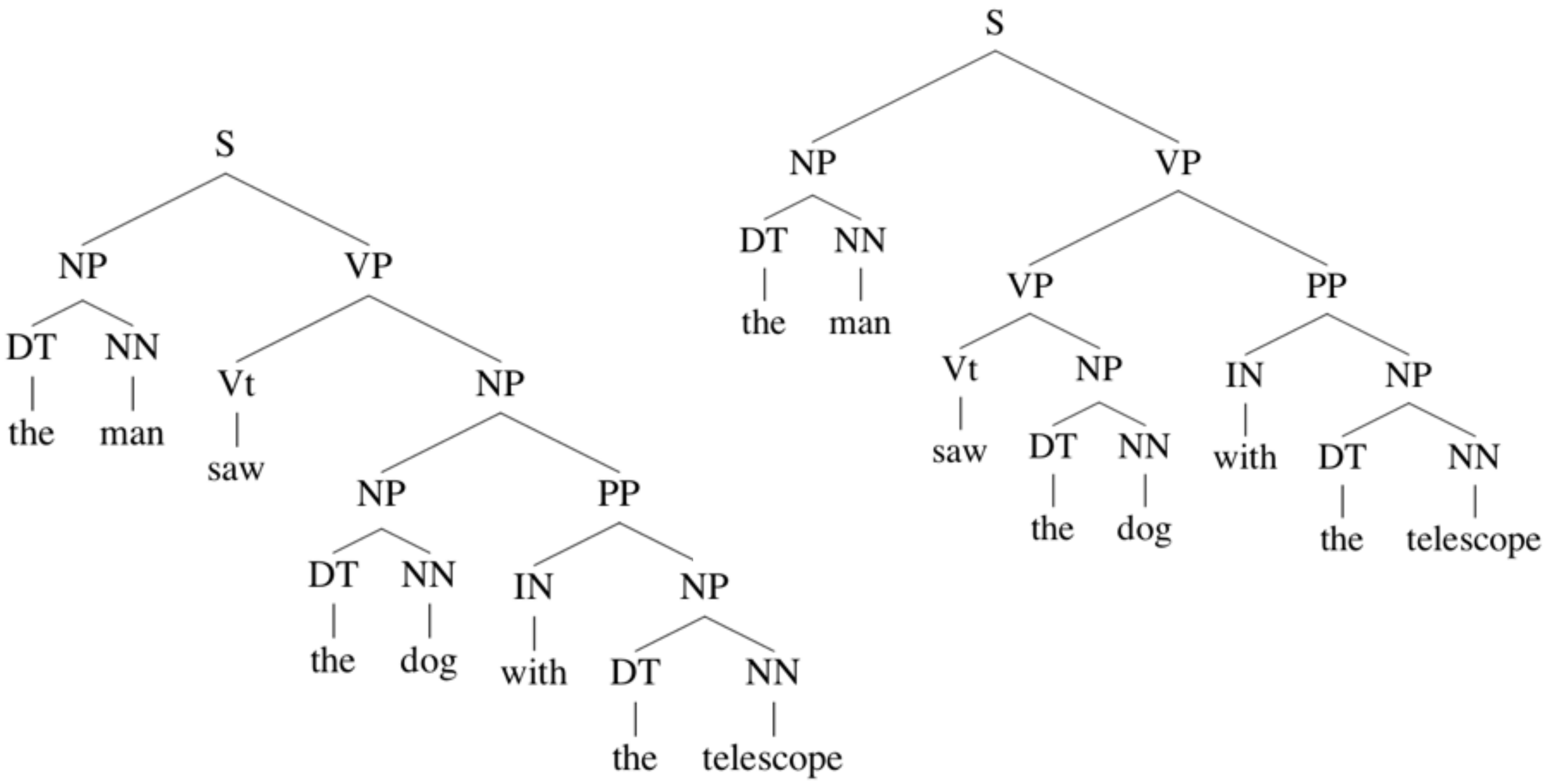
- each and every possible tree yielding **s**

We must be able to represent partial derivations

- including alternative ones

Ambiguity

Some strings may have more than one derivation in G



Dealing with Ambiguity

Statistical model: weight steps in a derivation

- induces a partial ordering over derivations
- can be used to make a decision
  - e.g. best tree under the model

Probabilistic CFG

CFG extended with parameters 0 ≤ θ_r ≤ 1

- where r ∊ R and

$$\sum_{\alpha:X\rightarrow\alpha\in R}\theta_{X\rightarrow\alpha}=1$$

Probabilistic CFG

Distribution over trees

$$P(T = t, S = \text{yield}(t)) = P(T = \langle r_1 \ldots r_n \rangle, S = s)$$

$$= \prod_{i=1}^{n} \theta_{r_i} = \prod_{i=1}^{n} \theta_{X_i \to \alpha_i} = \prod_{r \in t}^{n} \theta_r{}^{n(r,t)}$$

and strings

$$P(S = s) = \sum_{t \in T_G(s)} P(T = t, S = s)$$

Estimation

Let us assume the parametric form of θ is a multinomial

- one categorical distribution per X ∈ N

Suppose we can observe a *treebank*, then by MLE

$$\begin{aligned}\theta_{X \to \alpha} &= \frac{n(X \to \alpha)}{n(X)} \\ &= \frac{n(X \to \alpha)}{\sum_{\alpha'} n(X \to \alpha')}\end{aligned}$$

Weighted CKY+

Input: G and *s* = $w_1 \ldots w_n$

Item form: $[i, X \rightarrow \alpha_{\blacksquare} \bullet \beta_{\square}, j]$
asserts that $X \Rightarrow^* w_{i+1} \ldots w_j \beta$

Axioms: $[i, X \rightarrow w_i \bullet \alpha_{\square}, i+1]:\theta_r$ $r = X \rightarrow w_i\, \alpha \in R$
$[i, X \rightarrow \varepsilon \bullet, i]:\theta_r$ $r = X \rightarrow \varepsilon \in R$

Goal: $[0, S \rightarrow \alpha_{\blacksquare} \bullet, n]$

Scan

$$\frac{[i, X \rightarrow \alpha_{\blacksquare} \bullet w_{j+1}\, \beta_{\square}, j] : \theta_1}{[i, X \rightarrow \alpha_{\blacksquare} w_{j+1} \bullet \beta_{\square}, j+1] : \theta_1}$$

Prefix

$$\frac{[i, Y \rightarrow \alpha_{\blacksquare} \bullet, j] : \theta_1}{[i, X \rightarrow Y_{i,j} \bullet \beta_{\square}, j] : \theta_r} \quad r = X \rightarrow Y\beta \in R$$

Complete

$$\frac{[i, X \rightarrow \alpha_{\blacksquare} \bullet Y\beta_{\square}, k] : \theta_1\ [k, Y \rightarrow \gamma_{\blacksquare} \bullet, j] : \theta_2}{[i, X \rightarrow \alpha_{\blacksquare}\, Y_{k,j} \bullet \beta_{\square}, j] : \theta_1}$$

Joint Distribution

$_0S_3 \rightarrow {}_0NP_2\ {}_2VP_3$

$_0NP_2 \rightarrow {}_0DT_1\ {}_1NN_2$

$_2VP_3 \rightarrow {}_2Vi_3$

$_0DT_1 \rightarrow \text{the}$

$_1NN_2 \rightarrow \text{man}$

$_2Vi_3 \rightarrow \text{sleeps}$

Joint Distribution

${}_{0}\mathrm{S}_{3} \rightarrow {}_{0}\mathrm{NP}_{2}\ {}_{2}\mathrm{VP}_{3}$

${}_{0}\mathrm{NP}_{2} \rightarrow {}_{0}\mathrm{DT}_{1}\ {}_{1}\mathrm{NN}_{2}$

${}_{2}\mathrm{VP}_{3} \rightarrow {}_{2}\mathrm{Vi}_{3}$

${}_{0}\mathrm{DT}_{1} \rightarrow$ the

${}_{1}\mathrm{NN}_{2} \rightarrow$ man

${}_{2}\mathrm{Vi}_{3} \rightarrow$ sleeps

the man sleeps

Joint Distribution

${}_{0}S_{3} \rightarrow {}_{0}NP_{2}\ {}_{2}VP_{3}$

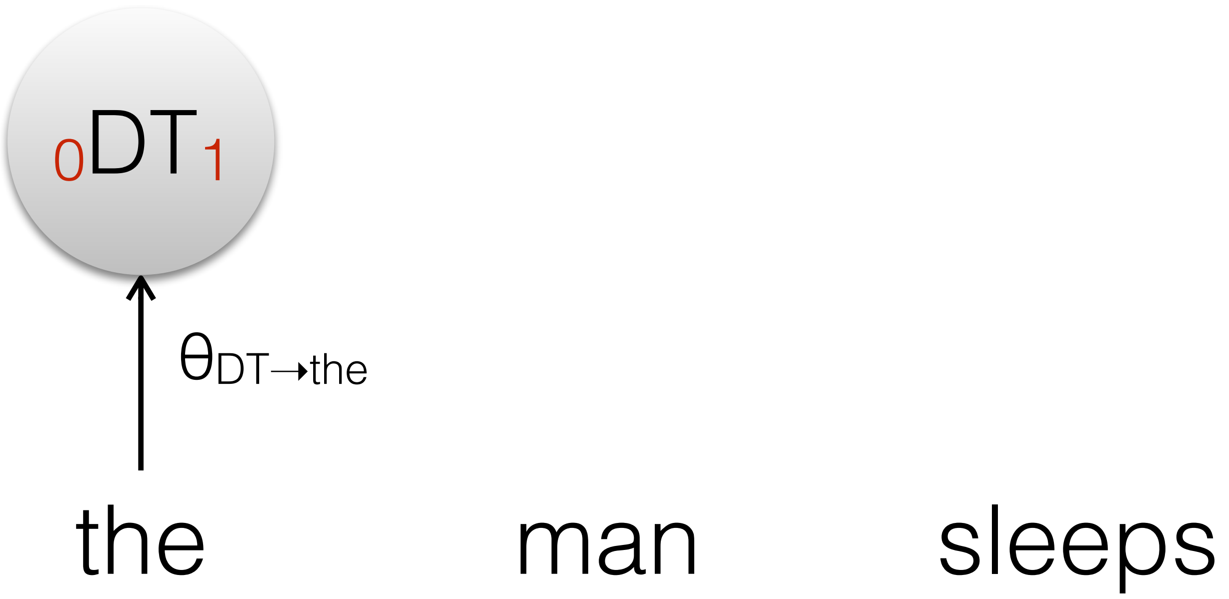
${}_{0}NP_{2} \rightarrow {}_{0}DT_{1}\ {}_{1}NN_{2}$

${}_{2}VP_{3} \rightarrow {}_{2}Vi_{3}$

${}_{0}DT_{1} \rightarrow$ the

${}_{1}NN_{2} \rightarrow$ man

${}_{2}Vi_{3} \rightarrow$ sleeps



Joint Distribution

${}_0S_3 \rightarrow {}_0NP_2\ {}_2VP_3$

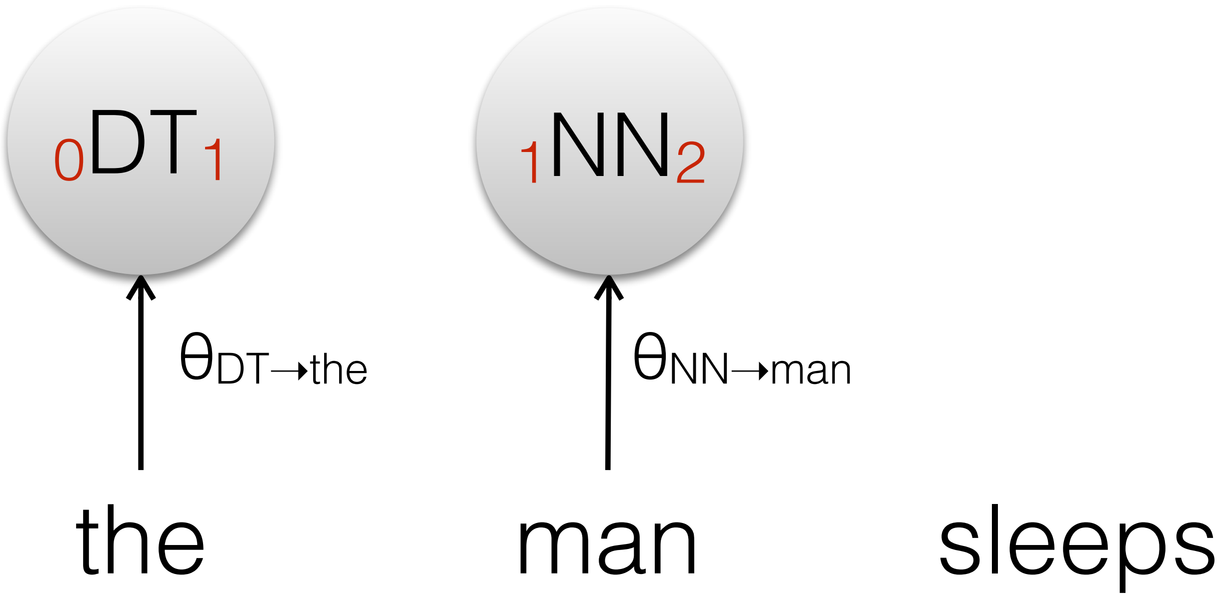
${}_0NP_2 \rightarrow {}_0DT_1\ {}_1NN_2$

${}_2VP_3 \rightarrow {}_2Vi_3$

${}_0DT_1 \rightarrow$ the

${}_1NN_2 \rightarrow$ man

${}_2Vi_3 \rightarrow$ sleeps



Joint Distribution

$_{0}S_{3} \rightarrow {}_{0}NP_{2}\ {}_{2}VP_{3}$

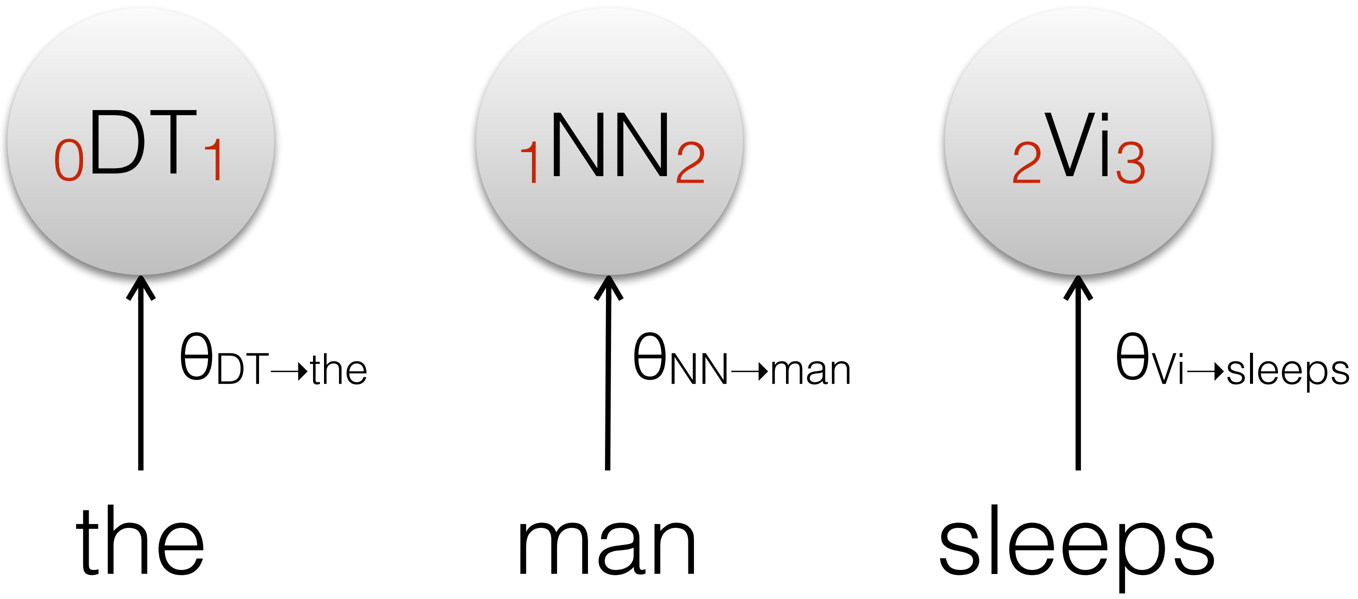
$_{0}NP_{2} \rightarrow {}_{0}DT_{1}\ {}_{1}NN_{2}$

$_{2}VP_{3} \rightarrow {}_{2}Vi_{3}$

$_{0}DT_{1} \rightarrow$ the

$_{1}NN_{2} \rightarrow$ man

$_{2}Vi_{3} \rightarrow$ sleeps


$_{0}DT_{1}$
$_{1}NN_{2}$
$_{2}Vi_{3}$
$\theta_{DT\rightarrow the}$
$\theta_{NN\rightarrow man}$
$\theta_{Vi\rightarrow sleeps}$
the
man
sleeps

Joint Distribution

${}_0S_3 \rightarrow {}_0NP_2\ {}_2VP_3$

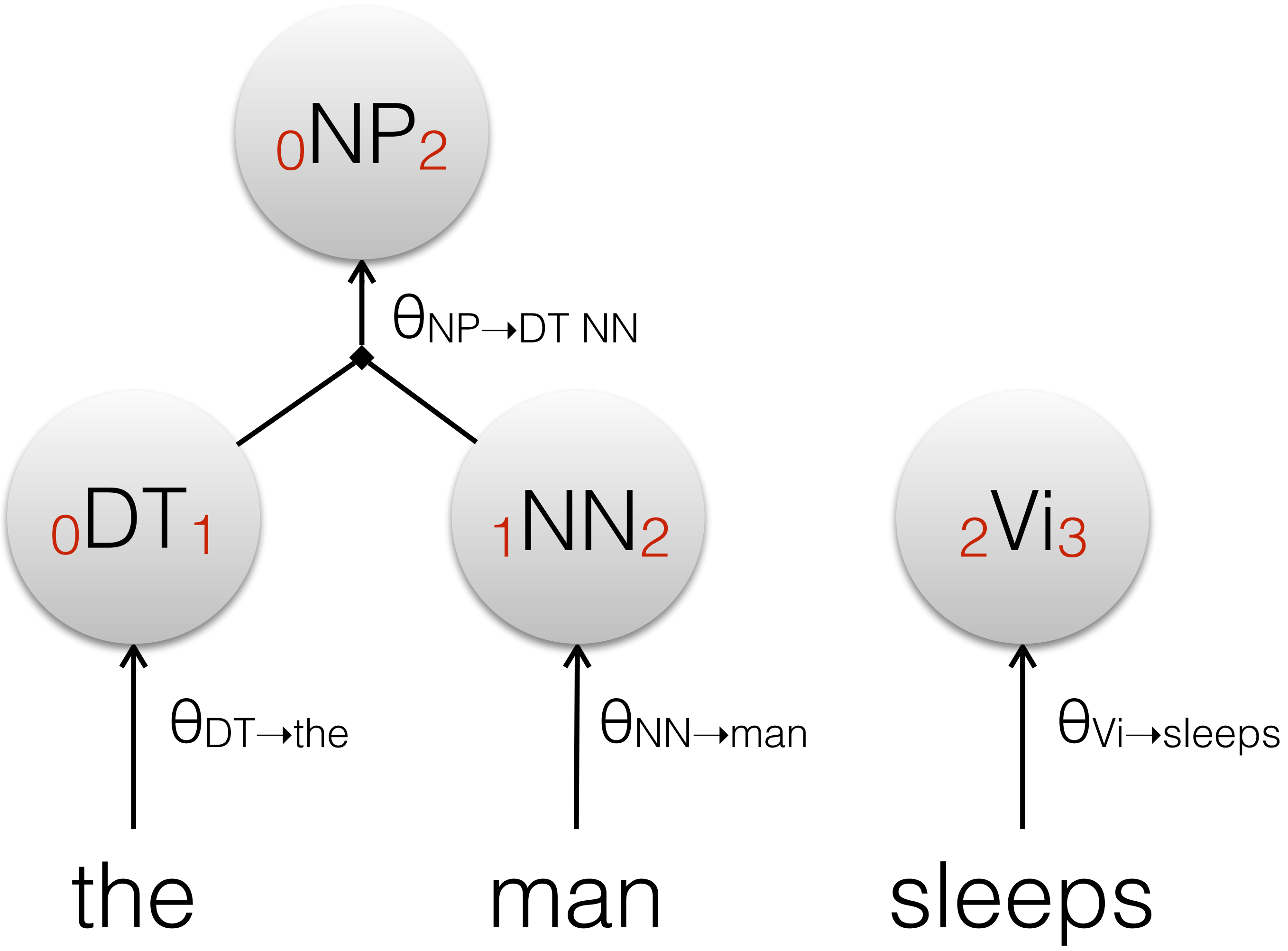
${}_0NP_2 \rightarrow {}_0DT_1\ {}_1NN_2$

${}_2VP_3 \rightarrow {}_2Vi_3$

${}_0DT_1 \rightarrow$ the

${}_1NN_2 \rightarrow$ man

${}_2Vi_3 \rightarrow$ sleeps



Joint Distribution

${}_0\mathrm{S}_3 \rightarrow {}_0\mathrm{NP}_2\ {}_2\mathrm{VP}_3$

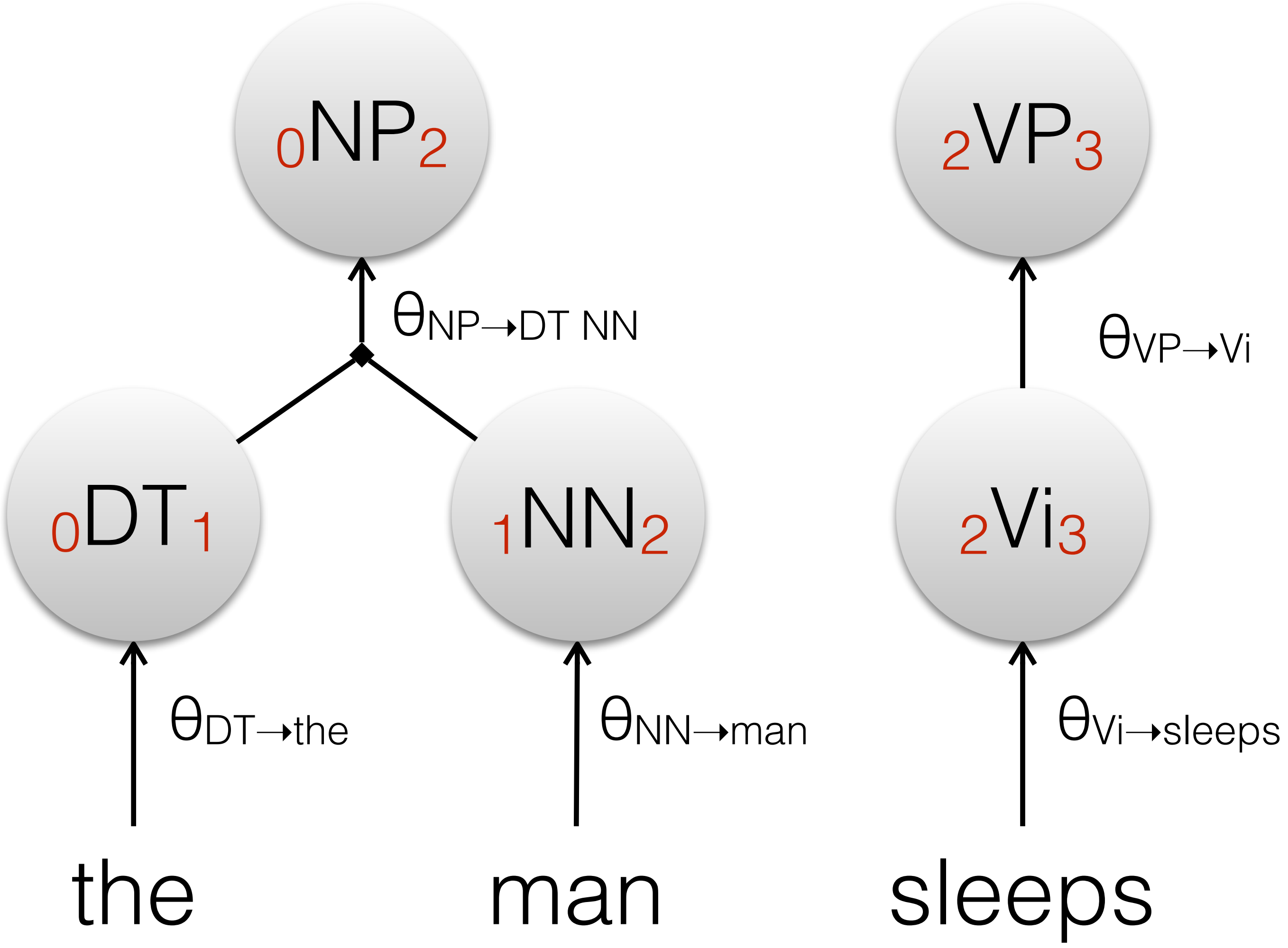
${}_0\mathrm{NP}_2 \rightarrow {}_0\mathrm{DT}_1\ {}_1\mathrm{NN}_2$

${}_2\mathrm{VP}_3 \rightarrow {}_2\mathrm{Vi}_3$

${}_0\mathrm{DT}_1 \rightarrow \text{the}$

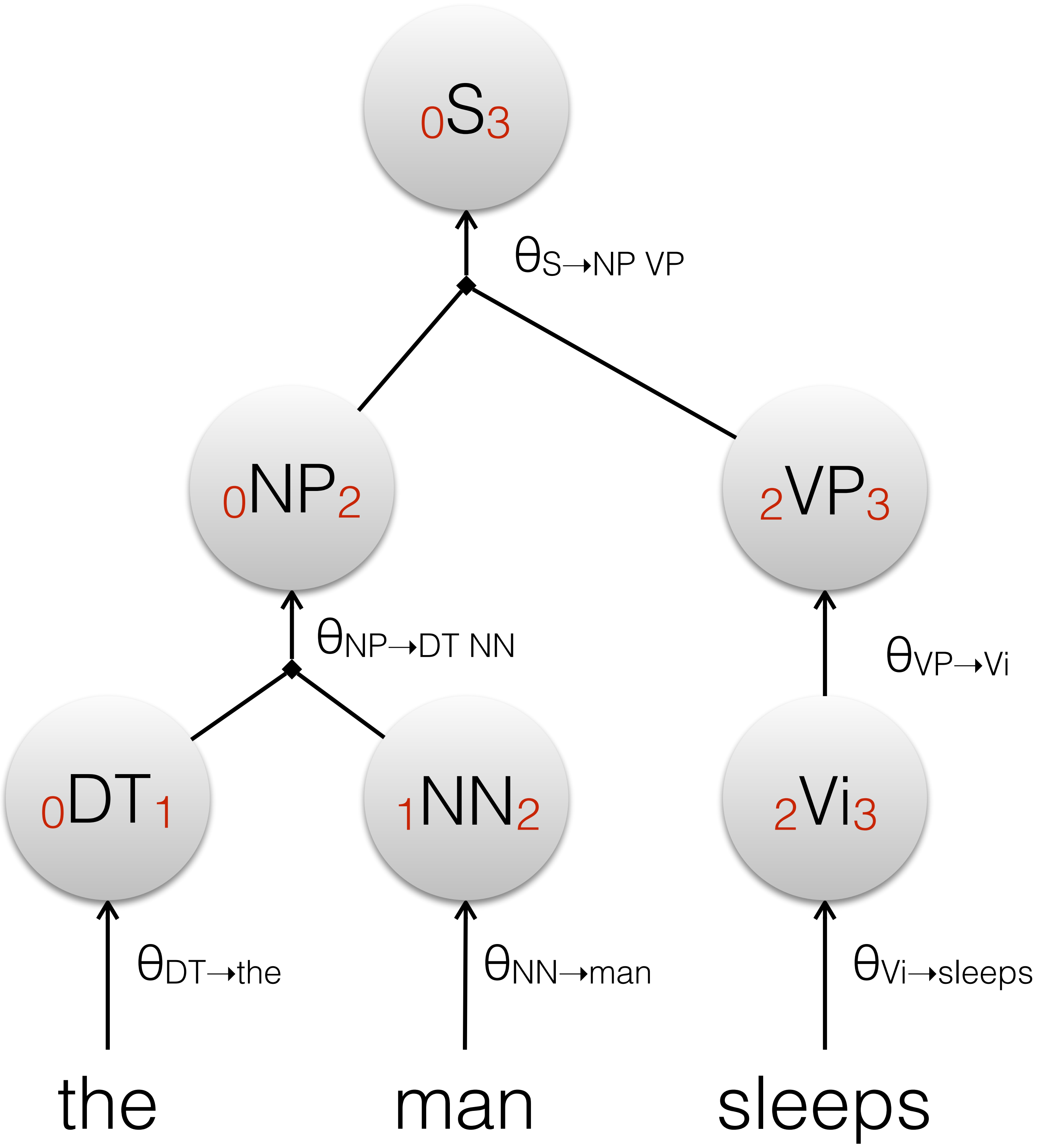
${}_1\mathrm{NN}_2 \rightarrow \text{man}$

${}_2\mathrm{Vi}_3 \rightarrow \text{sleeps}$



Joint Distribution

${}_0S_3 \rightarrow {}_0NP_2\ {}_2VP_3$
${}_0NP_2 \rightarrow {}_0DT_1\ {}_1NN_2$
${}_2VP_3 \rightarrow {}_2Vi_3$
${}_0DT_1 \rightarrow$ the
${}_1NN_2 \rightarrow$ man
${}_2Vi_3 \rightarrow$ sleeps


${}_0S_3$
$\theta_{S \rightarrow NP\ VP}$
${}_0NP_2$
${}_2VP_3$
$\theta_{NP \rightarrow DT\ NN}$
$\theta_{VP \rightarrow Vi}$
${}_0DT_1$
${}_1NN_2$
${}_2Vi_3$
$\theta_{DT \rightarrow the}$
$\theta_{NN \rightarrow man}$
$\theta_{Vi \rightarrow sleeps}$
the
man
sleeps

Ambiguity

Ambiguity

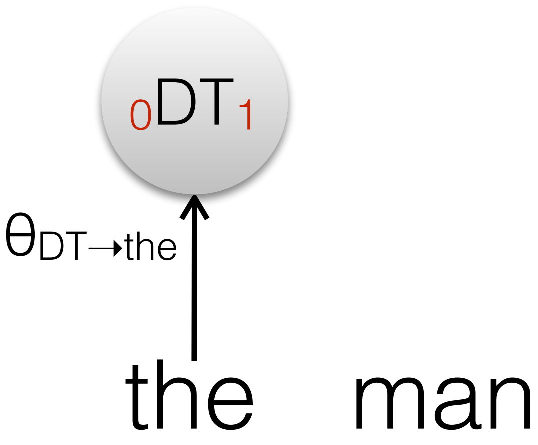
the man

saw

the dog

with the telescope

Ambiguity



${}_0DT_1$

$\theta_{DT \rightarrow the}$

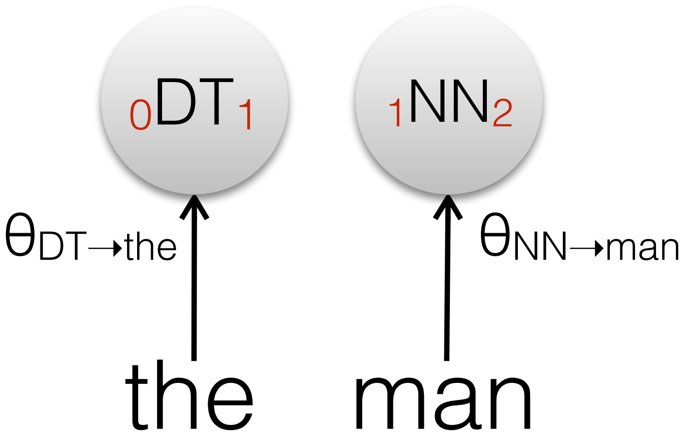
the man

saw

the dog

with the telescope

Ambiguity

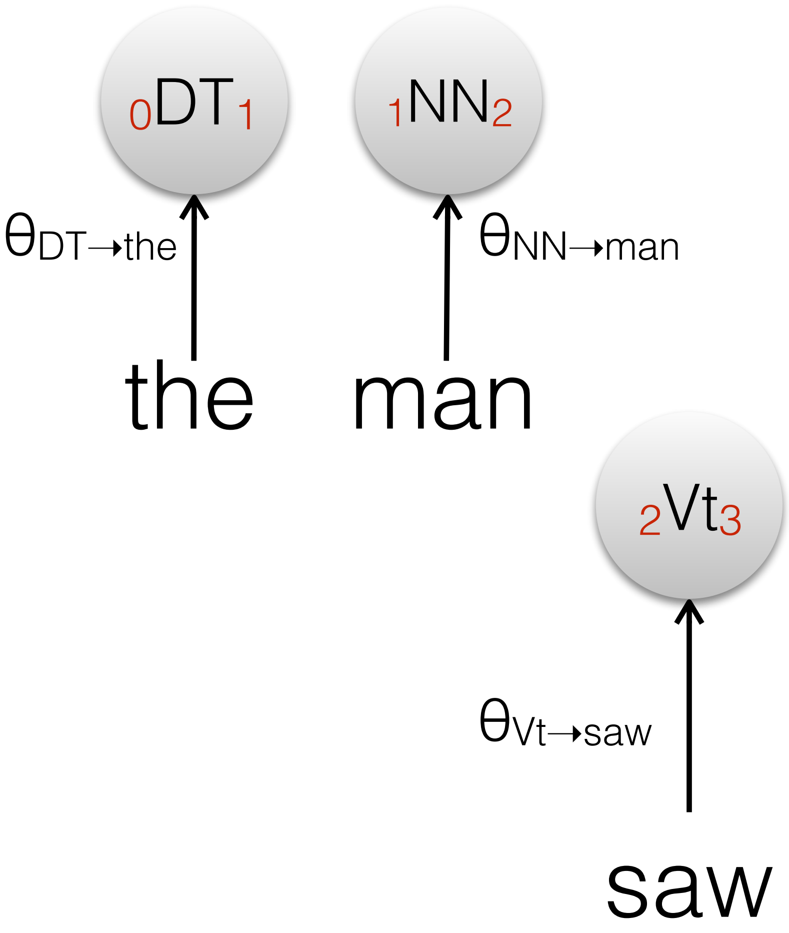

$_0\mathrm{DT}_1$
$_1\mathrm{NN}_2$
$\theta_{\mathrm{DT}\rightarrow\mathrm{the}}$
$\theta_{\mathrm{NN}\rightarrow\mathrm{man}}$
the
man

saw

the dog

with the telescope

Ambiguity



$_0\mathrm{DT}_1$

$_1\mathrm{NN}_2$

$\theta_{\mathrm{DT}\rightarrow\mathrm{the}}$

$\theta_{\mathrm{NN}\rightarrow\mathrm{man}}$

the man

$_2\mathrm{Vt}_3$

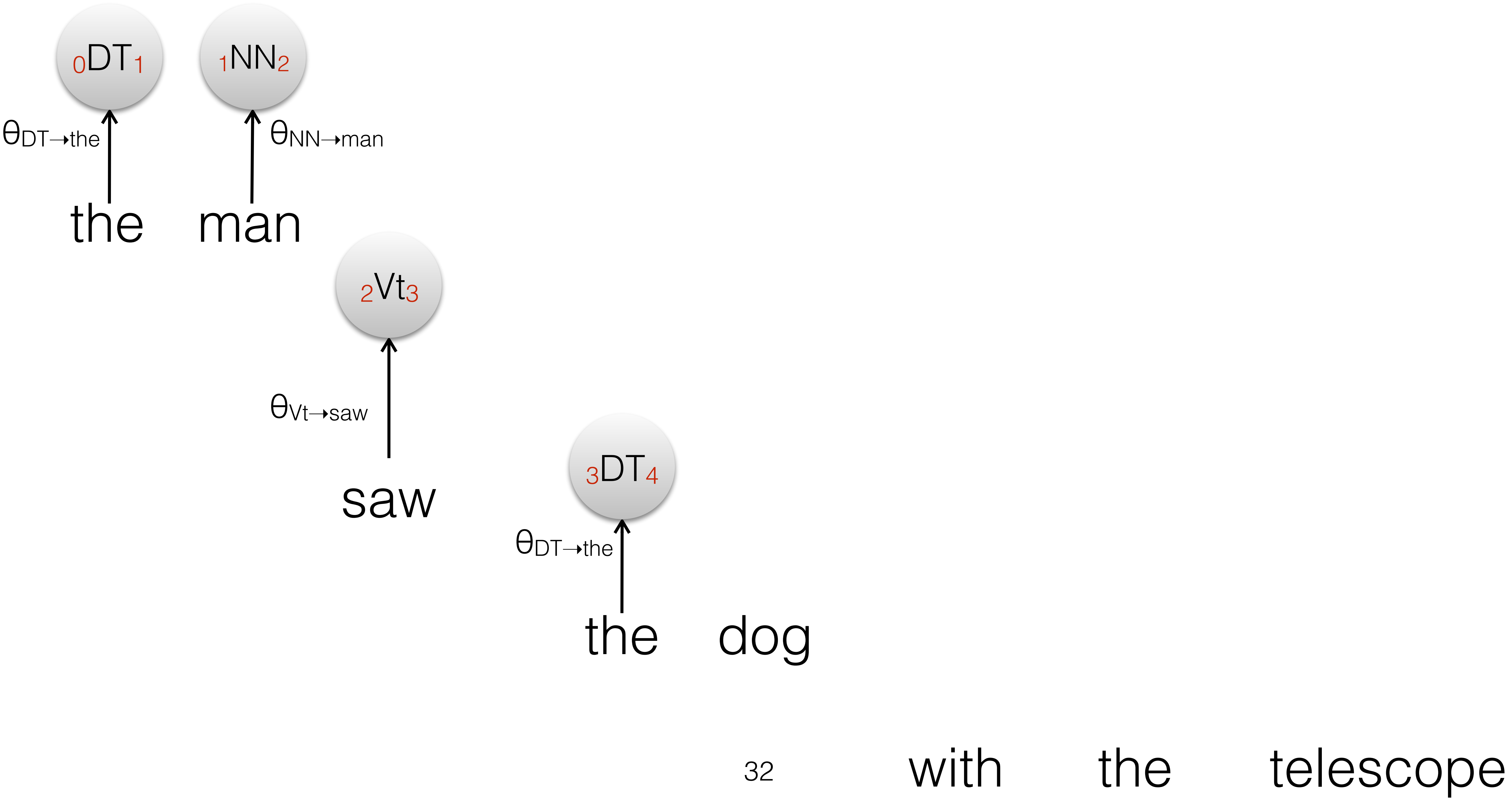
$\theta_{\mathrm{Vt}\rightarrow\mathrm{saw}}$

saw

the dog

with the telescope

Ambiguity



${}_0\mathrm{DT}_1$

${}_1\mathrm{NN}_2$

$\theta_{\mathrm{DT}\rightarrow\mathrm{the}}$

$\theta_{\mathrm{NN}\rightarrow\mathrm{man}}$

the man

${}_2\mathrm{Vt}_3$

$\theta_{\mathrm{Vt}\rightarrow\mathrm{saw}}$

saw

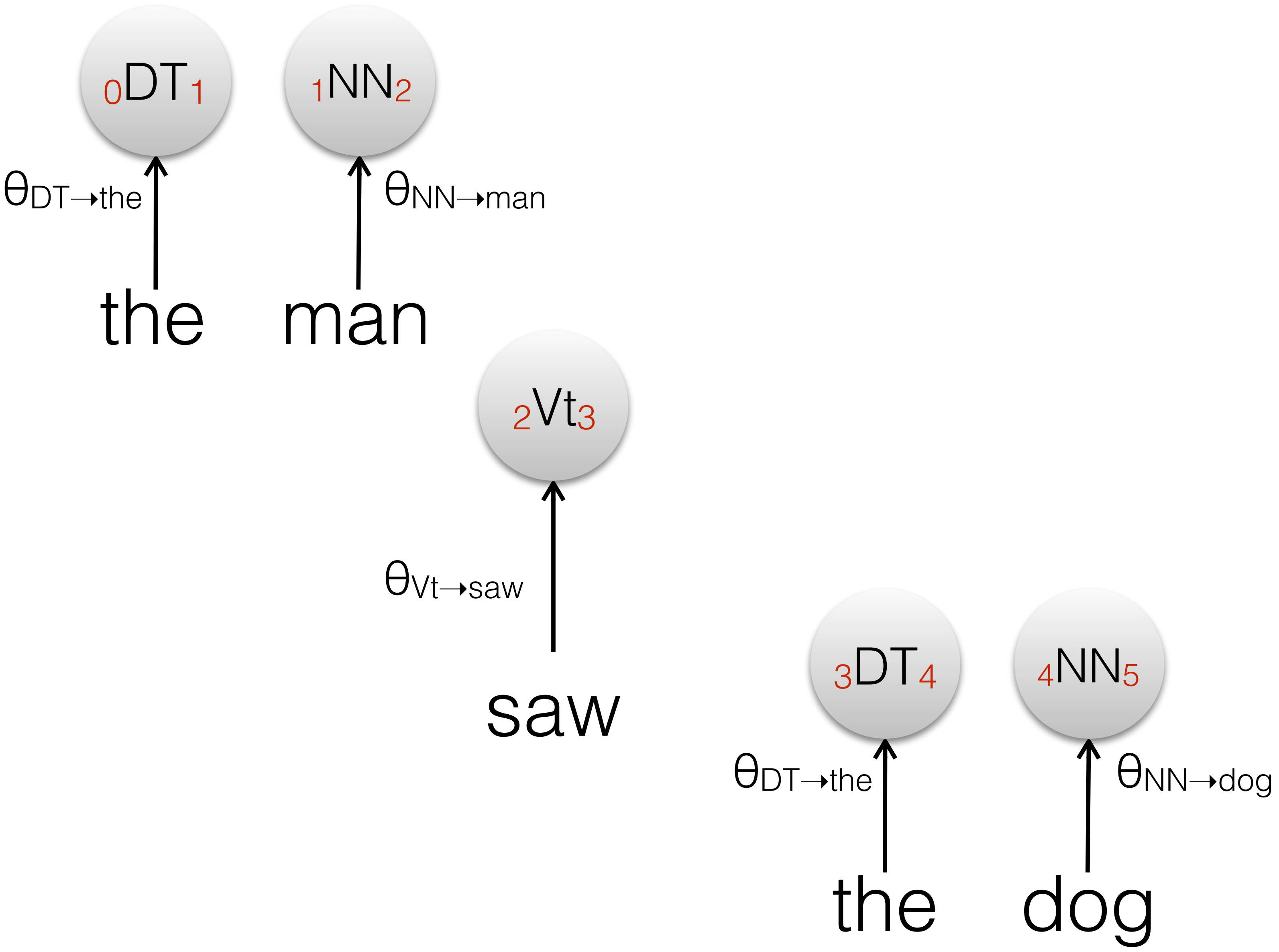
${}_3\mathrm{DT}_4$

$\theta_{\mathrm{DT}\rightarrow\mathrm{the}}$

the dog

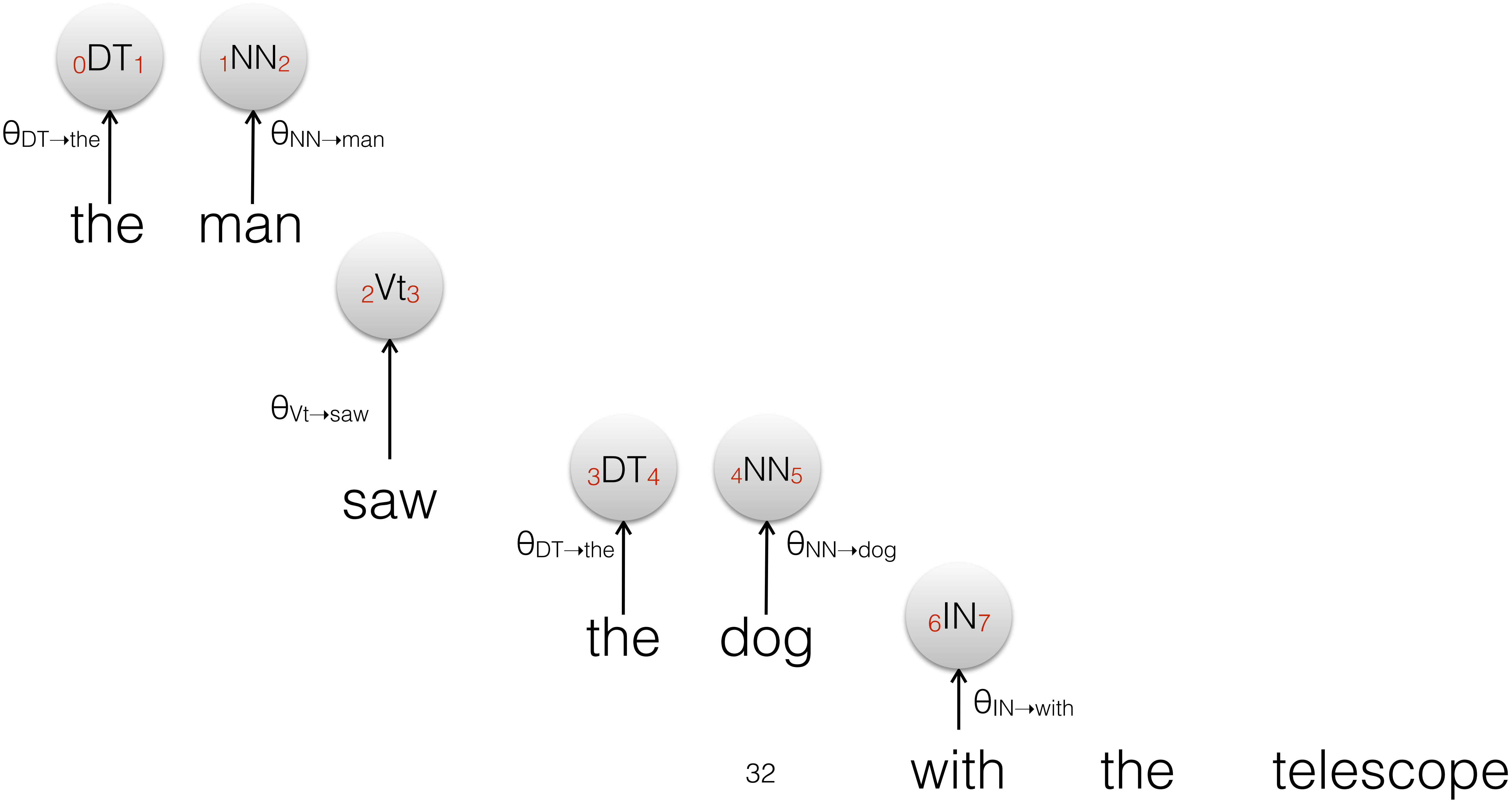
with the telescope

Ambiguity


$_0DT_1$
$_1NN_2$
$\theta_{DT\rightarrow the}$
$\theta_{NN\rightarrow man}$
the
man
$_2Vt_3$
$\theta_{Vt\rightarrow saw}$
saw
$_3DT_4$
$_4NN_5$
$\theta_{DT\rightarrow the}$
$\theta_{NN\rightarrow dog}$
the
dog

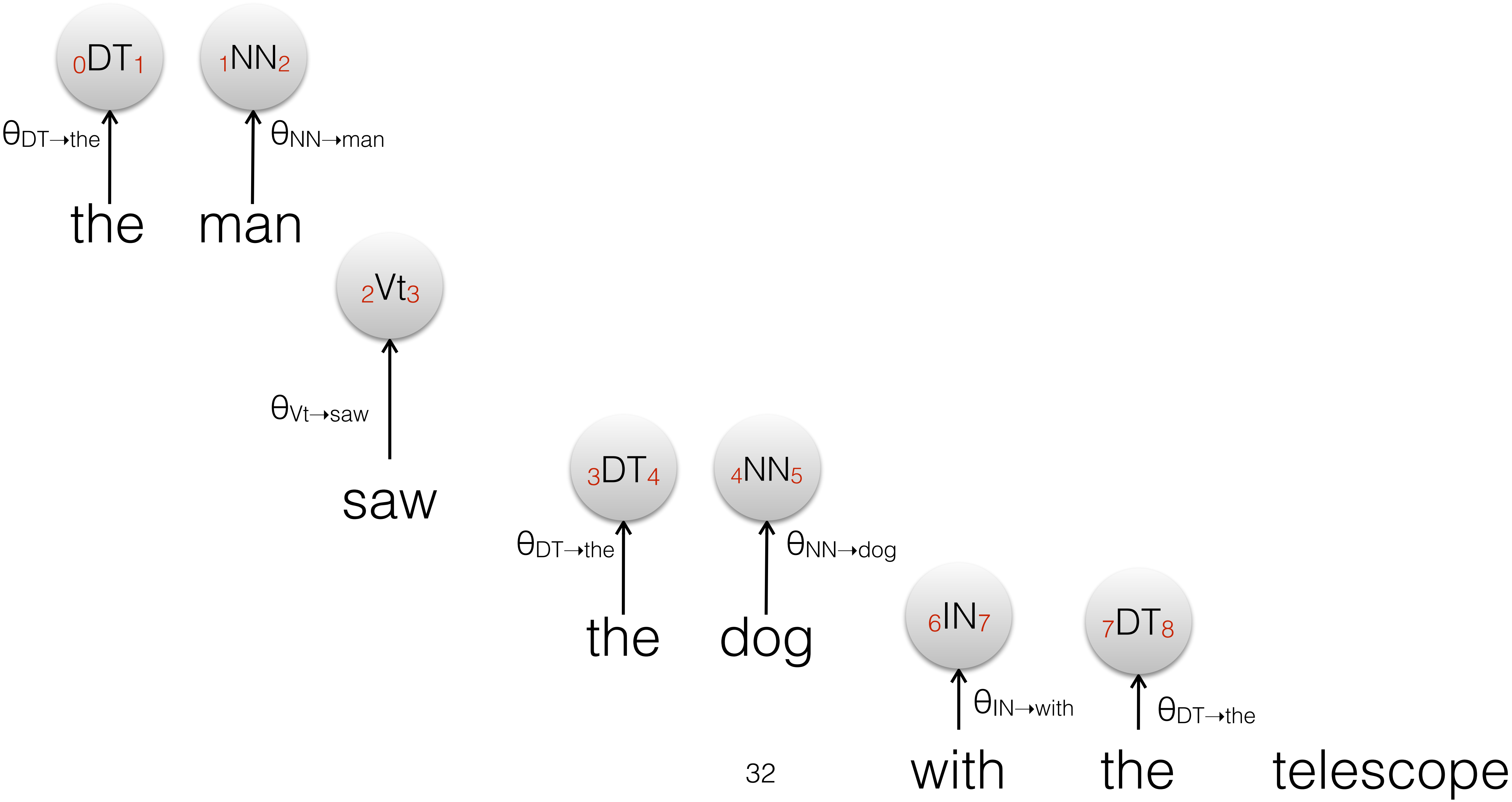
with the telescope

Ambiguity



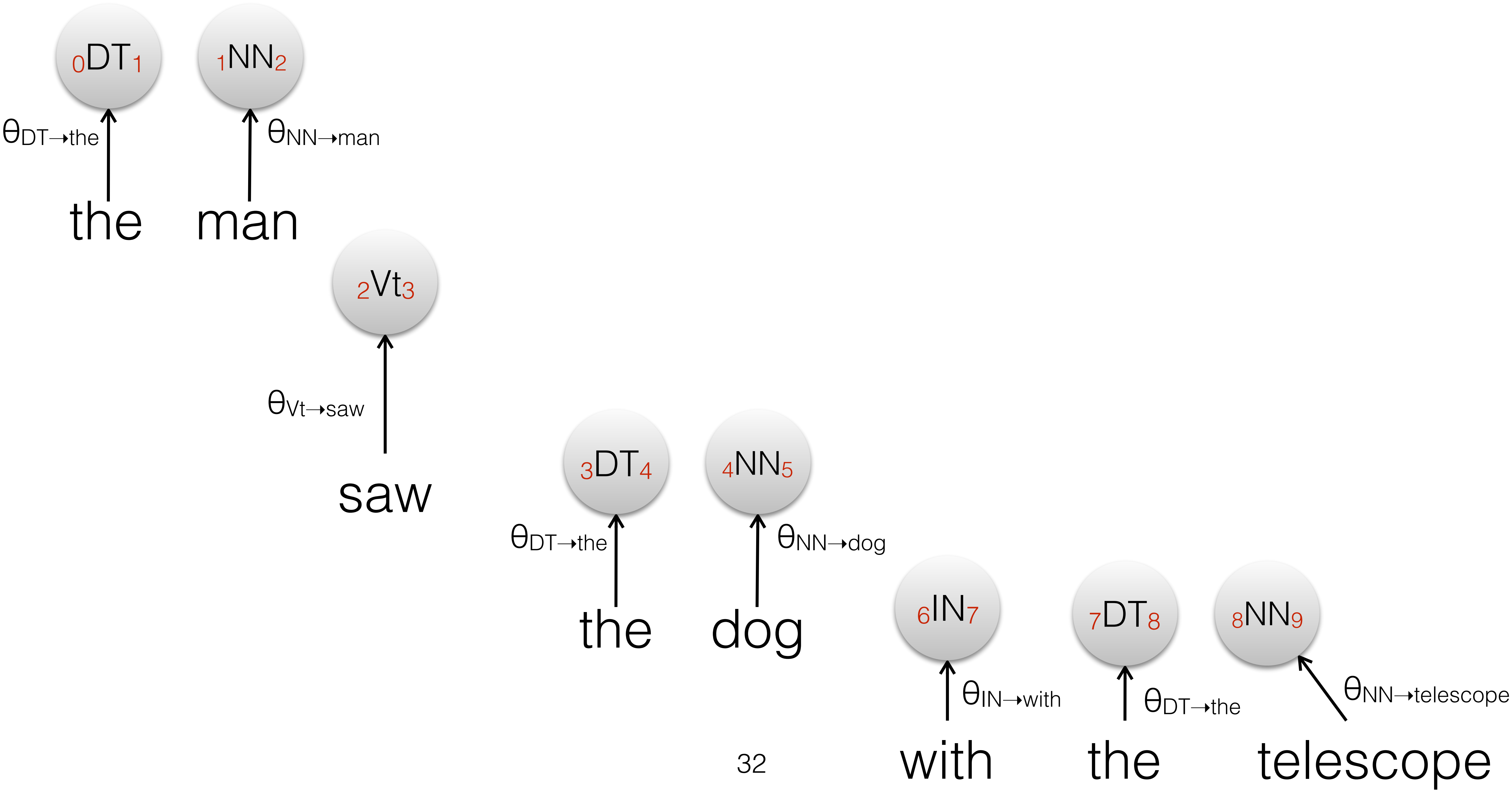
$_0$DT$_1$
$_1$NN$_2$
$\theta_{DT\to the}$
$\theta_{NN\to man}$
the
man
$_2$Vt$_3$
$\theta_{Vt\to saw}$
saw
$_3$DT$_4$
$_4$NN$_5$
$\theta_{DT\to the}$
$\theta_{NN\to dog}$
the
dog
$_6$IN$_7$
$\theta_{IN\to with}$
with
the
telescope

Ambiguity

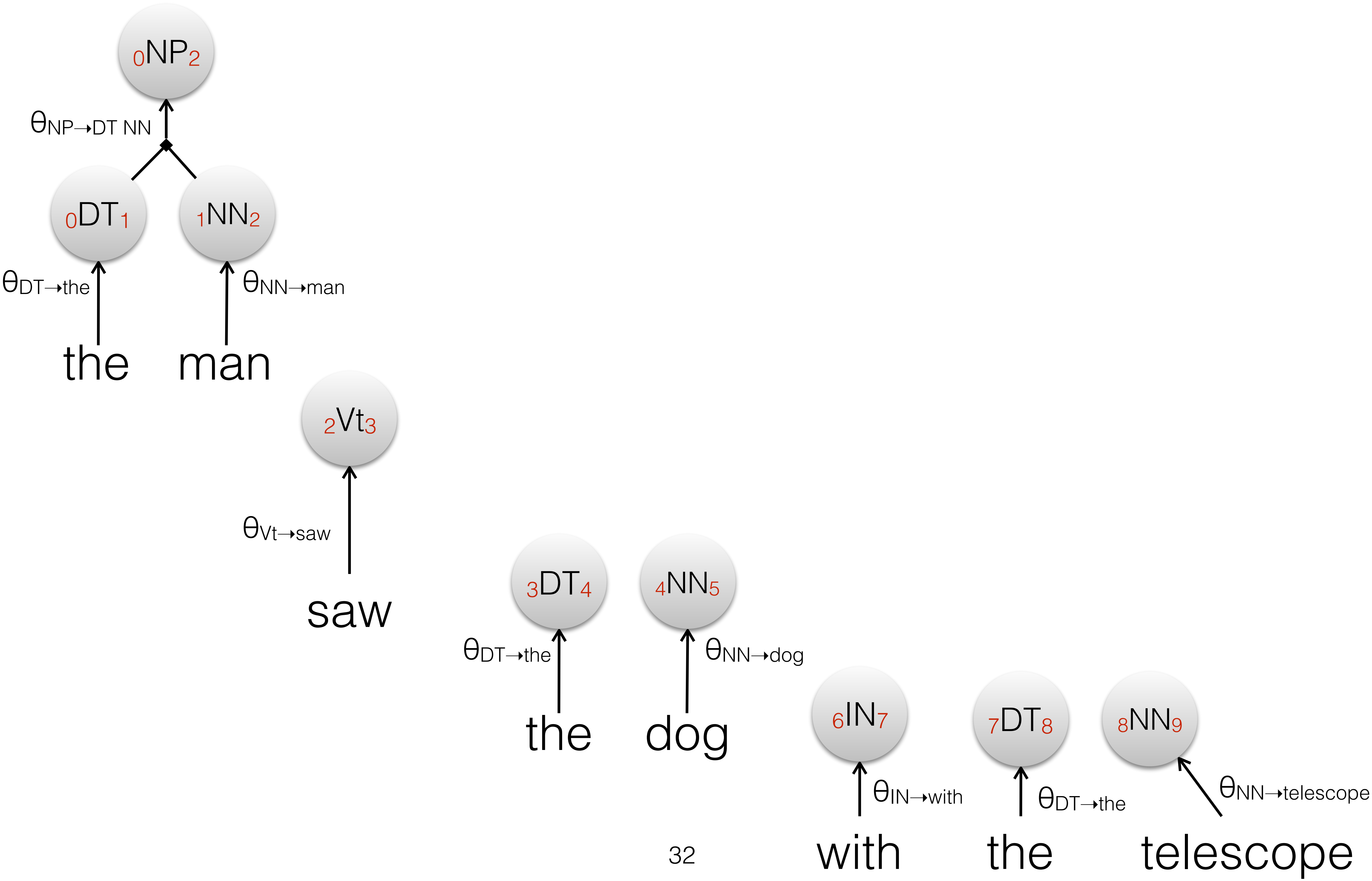


${}_0DT_1$
${}_1NN_2$
$\theta_{DT \rightarrow the}$
$\theta_{NN \rightarrow man}$
the
man
${}_2Vt_3$
$\theta_{Vt \rightarrow saw}$
saw
${}_3DT_4$
${}_4NN_5$
$\theta_{DT \rightarrow the}$
$\theta_{NN \rightarrow dog}$
the
dog
${}_6IN_7$
${}_7DT_8$
$\theta_{IN \rightarrow with}$
$\theta_{DT \rightarrow the}$
with
the
telescope

Ambiguity

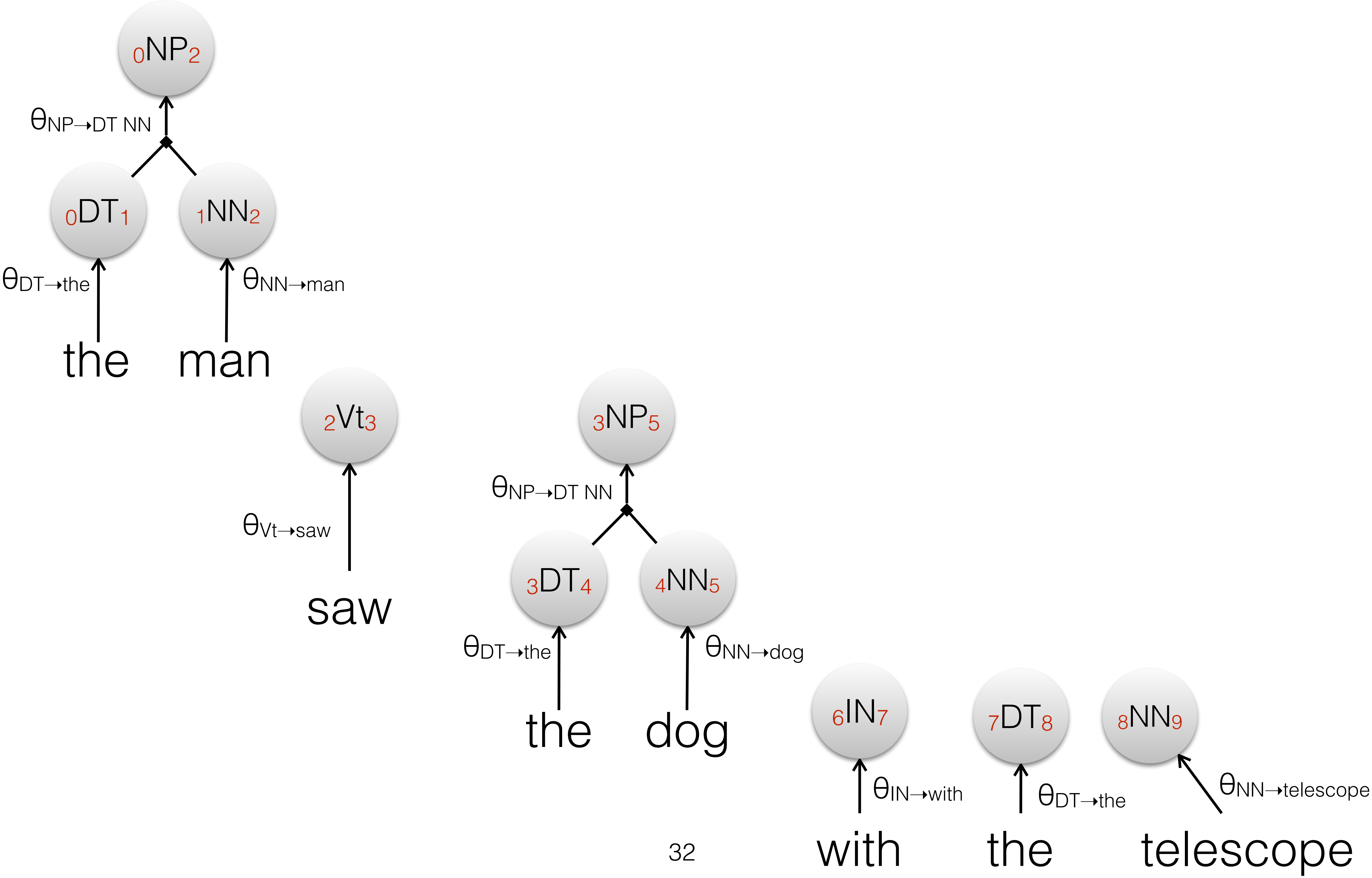

$_0$DT$_1$
$_1$NN$_2$
$\theta_{DT \to the}$
$\theta_{NN \to man}$
the
man
$_2$Vt$_3$
$\theta_{Vt \to saw}$
saw
$_3$DT$_4$
$_4$NN$_5$
$\theta_{DT \to the}$
$\theta_{NN \to dog}$
the
dog
$_6$IN$_7$
$_7$DT$_8$
$_8$NN$_9$
$\theta_{IN \to with}$
$\theta_{DT \to the}$
$\theta_{NN \to telescope}$
with
the
telescope

Ambiguity

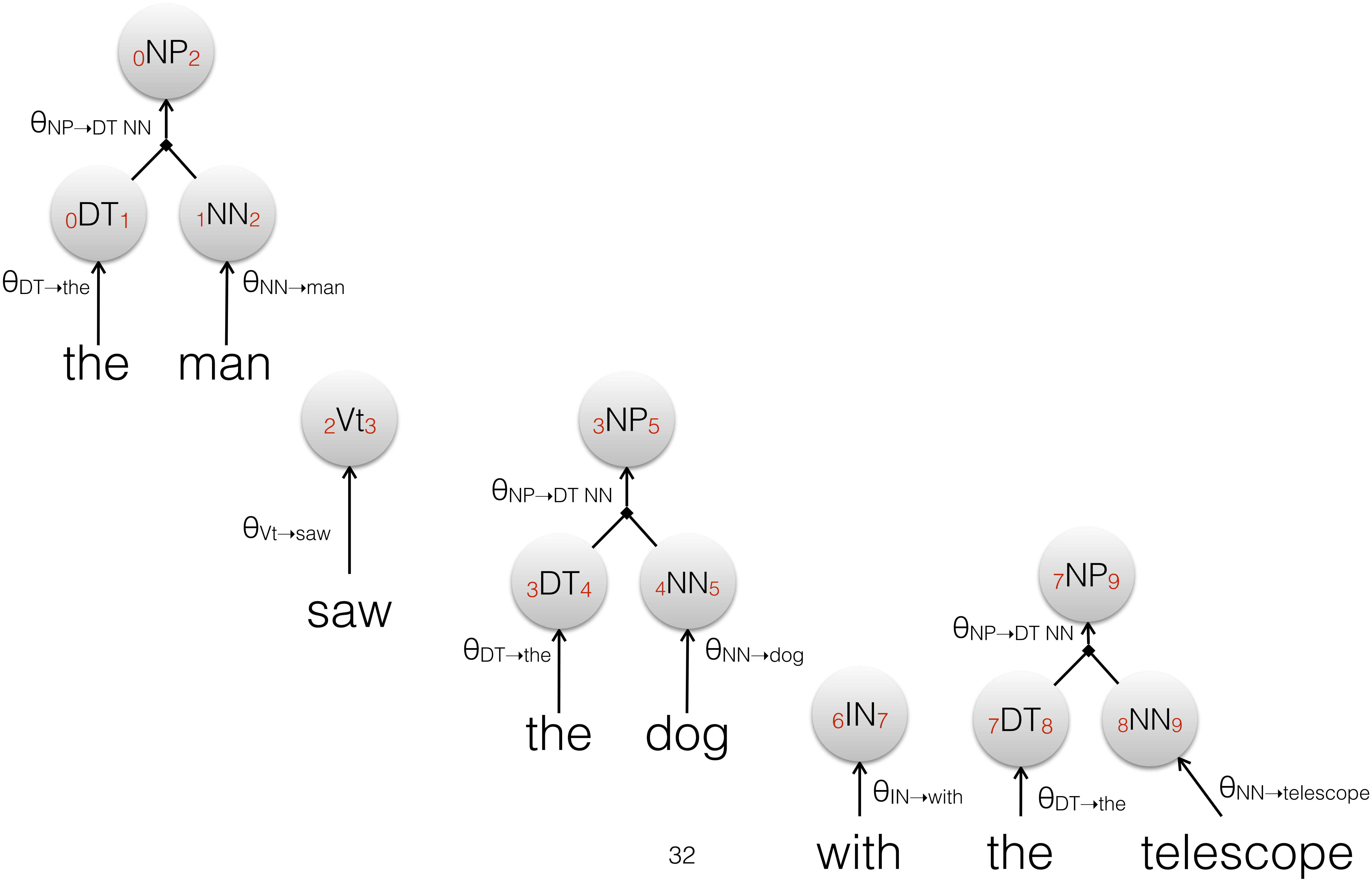


$_0NP_2$
$\theta_{NP \rightarrow DT\ NN}$
$_0DT_1$
$_1NN_2$
$\theta_{DT \rightarrow the}$
$\theta_{NN \rightarrow man}$
the
man
$_2Vt_3$
$\theta_{Vt \rightarrow saw}$
saw
$_3DT_4$
$_4NN_5$
$\theta_{DT \rightarrow the}$
$\theta_{NN \rightarrow dog}$
the
dog
$_6IN_7$
$_7DT_8$
$_8NN_9$
$\theta_{IN \rightarrow with}$
$\theta_{DT \rightarrow the}$
$\theta_{NN \rightarrow telescope}$
with
the
telescope

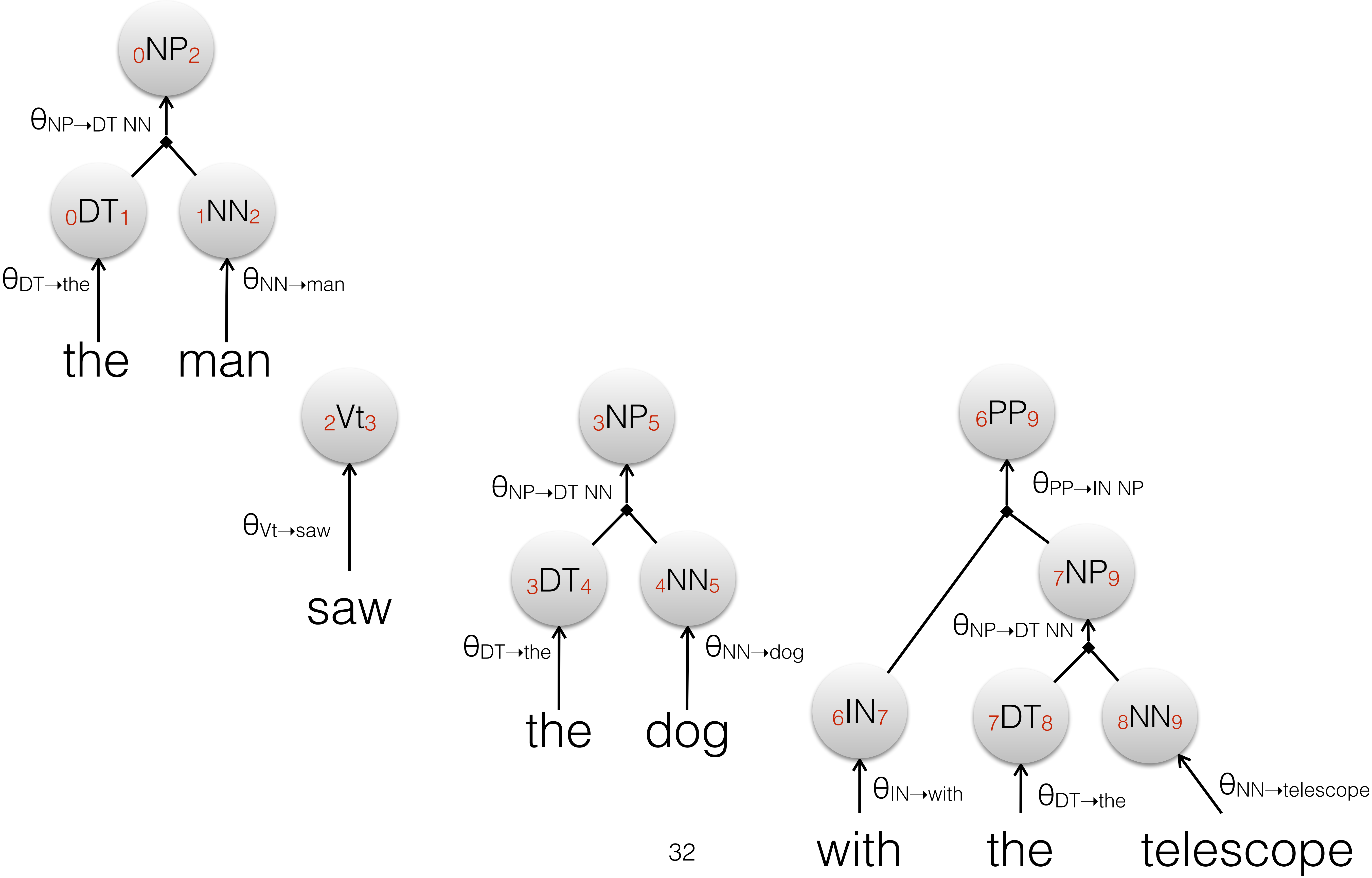
Ambiguity


$_0NP_2$
$\theta_{NP \to DT\ NN}$
$_0DT_1$
$_1NN_2$
$\theta_{DT \to the}$
$\theta_{NN \to man}$
the
man
$_2Vt_3$
$\theta_{Vt \to saw}$
saw
$_3NP_5$
$\theta_{NP \to DT\ NN}$
$_3DT_4$
$_4NN_5$
$\theta_{DT \to the}$
$\theta_{NN \to dog}$
the
dog
$_6IN_7$
$_7DT_8$
$_8NN_9$
$\theta_{IN \to with}$
$\theta_{DT \to the}$
$\theta_{NN \to telescope}$
with
the
telescope

Ambiguity

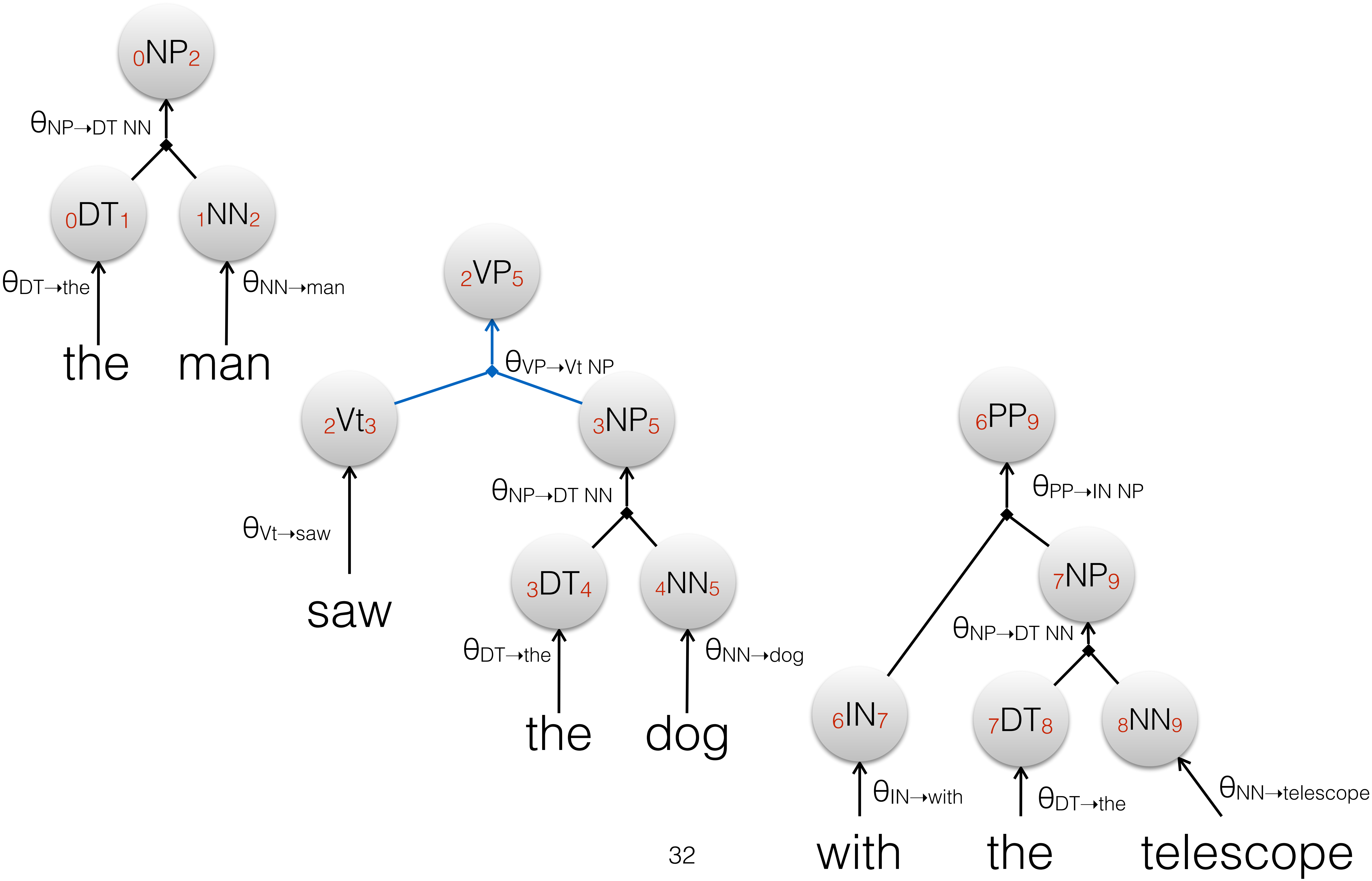

0NP2
θNP→DT NN
0DT1
1NN2
θDT→the
θNN→man
the
man
2Vt3
θVt→saw
saw
3NP5
θNP→DT NN
3DT4
4NN5
θDT→the
θNN→dog
the
dog
6IN7
θIN→with
with
7NP9
θNP→DT NN
7DT8
8NN9
θDT→the
θNN→telescope
the
telescope

Ambiguity

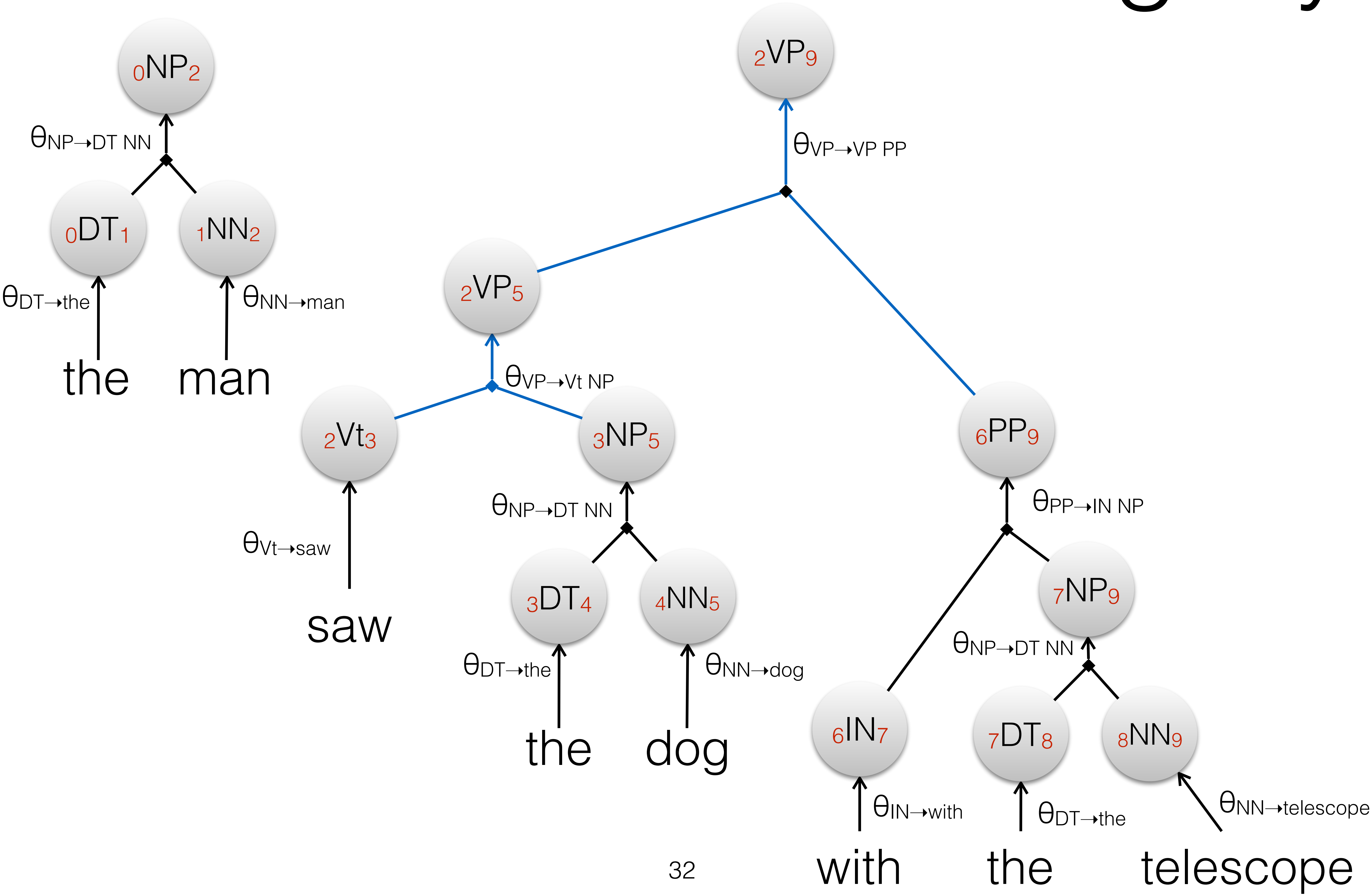


$_0NP_2$
$\theta_{NP \to DT\ NN}$
$_0DT_1$
$_1NN_2$
$\theta_{DT \to the}$
$\theta_{NN \to man}$
the
man
$_2Vt_3$
$\theta_{Vt \to saw}$
saw
$_3NP_5$
$\theta_{NP \to DT\ NN}$
$_3DT_4$
$_4NN_5$
$\theta_{DT \to the}$
$\theta_{NN \to dog}$
the
dog
$_6PP_9$
$\theta_{PP \to IN\ NP}$
$_7NP_9$
$\theta_{NP \to DT\ NN}$
$_6IN_7$
$_7DT_8$
$_8NN_9$
$\theta_{IN \to with}$
$\theta_{DT \to the}$
$\theta_{NN \to telescope}$
with
the
telescope

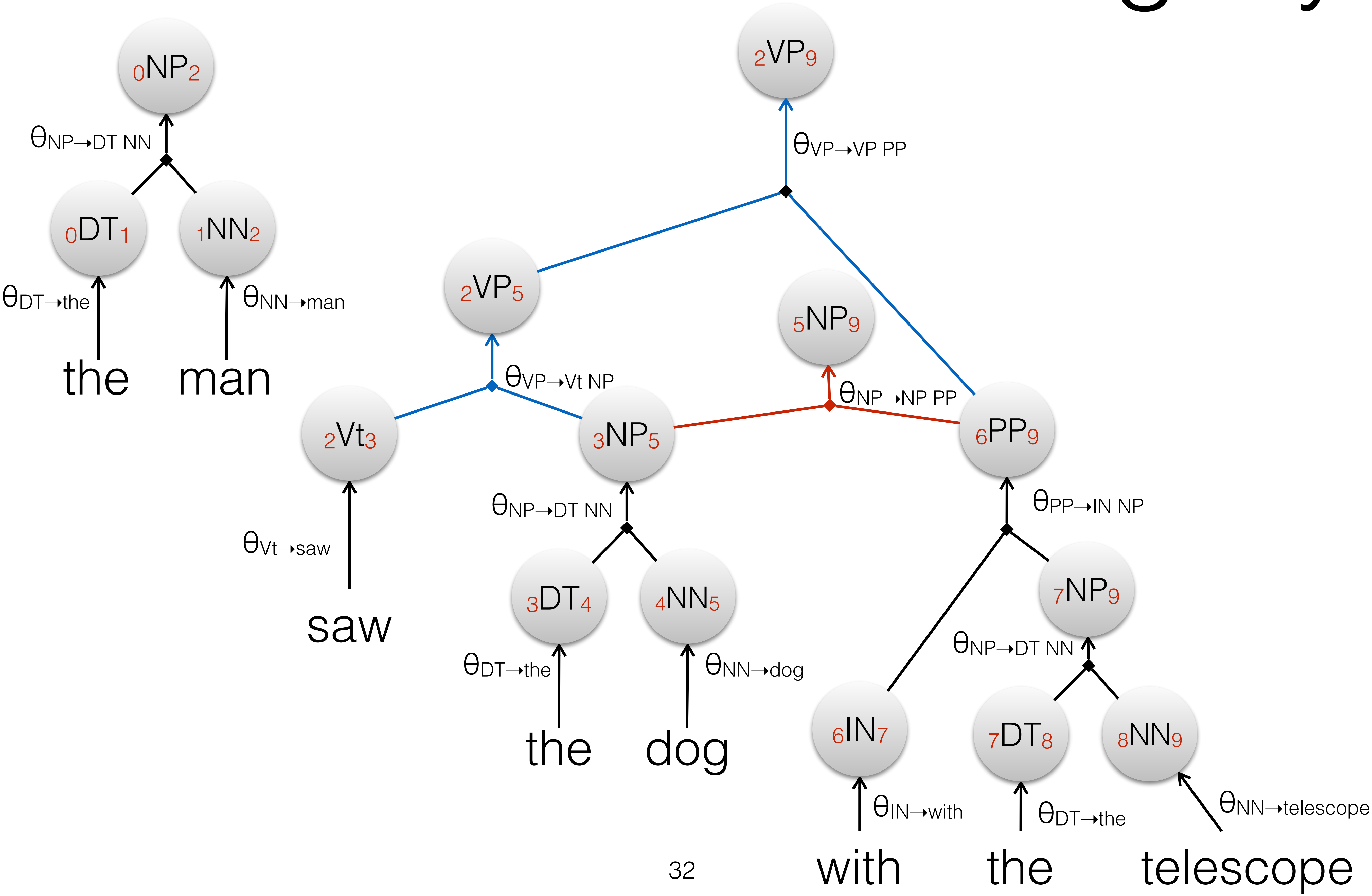
Ambiguity


$_0$NP$_2$
θNP→DT NN
$_0$DT$_1$
$_1$NN$_2$
θDT→the
θNN→man
the
man
$_2$VP$_5$
θVP→Vt NP
$_2$Vt$_3$
$_3$NP$_5$
θVt→saw
saw
θNP→DT NN
$_3$DT$_4$
$_4$NN$_5$
θDT→the
θNN→dog
the
dog
$_6$PP$_9$
θPP→IN NP
$_7$NP$_9$
θNP→DT NN
$_6$IN$_7$
$_7$DT$_8$
$_8$NN$_9$
θIN→with
θDT→the
θNN→telescope
with
the
telescope

Ambiguity

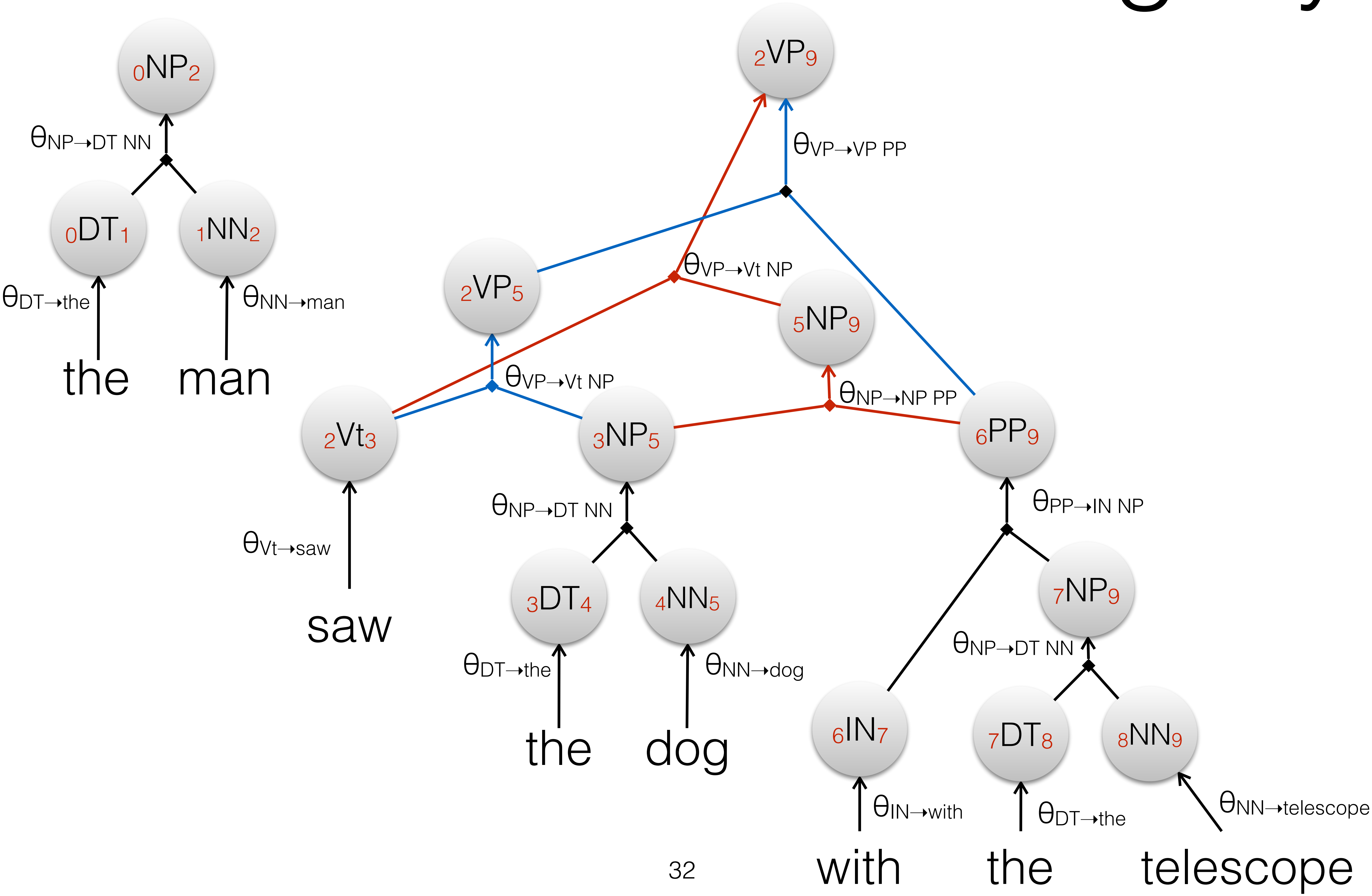

$_0NP_2$
$\theta_{NP \to DT\ NN}$
$_0DT_1$
$_1NN_2$
$\theta_{DT \to the}$
$\theta_{NN \to man}$
the
man
$_2VP_9$
$\theta_{VP \to VP\ PP}$
$_2VP_5$
$\theta_{VP \to Vt\ NP}$
$_2Vt_3$
$_3NP_5$
$_6PP_9$
$\theta_{Vt \to saw}$
saw
$\theta_{NP \to DT\ NN}$
$\theta_{PP \to IN\ NP}$
$_3DT_4$
$_4NN_5$
$_7NP_9$
$\theta_{DT \to the}$
$\theta_{NN \to dog}$
$\theta_{NP \to DT\ NN}$
the
dog
$_6IN_7$
$_7DT_8$
$_8NN_9$
$\theta_{IN \to with}$
$\theta_{DT \to the}$
$\theta_{NN \to telescope}$
with
the
telescope

Ambiguity



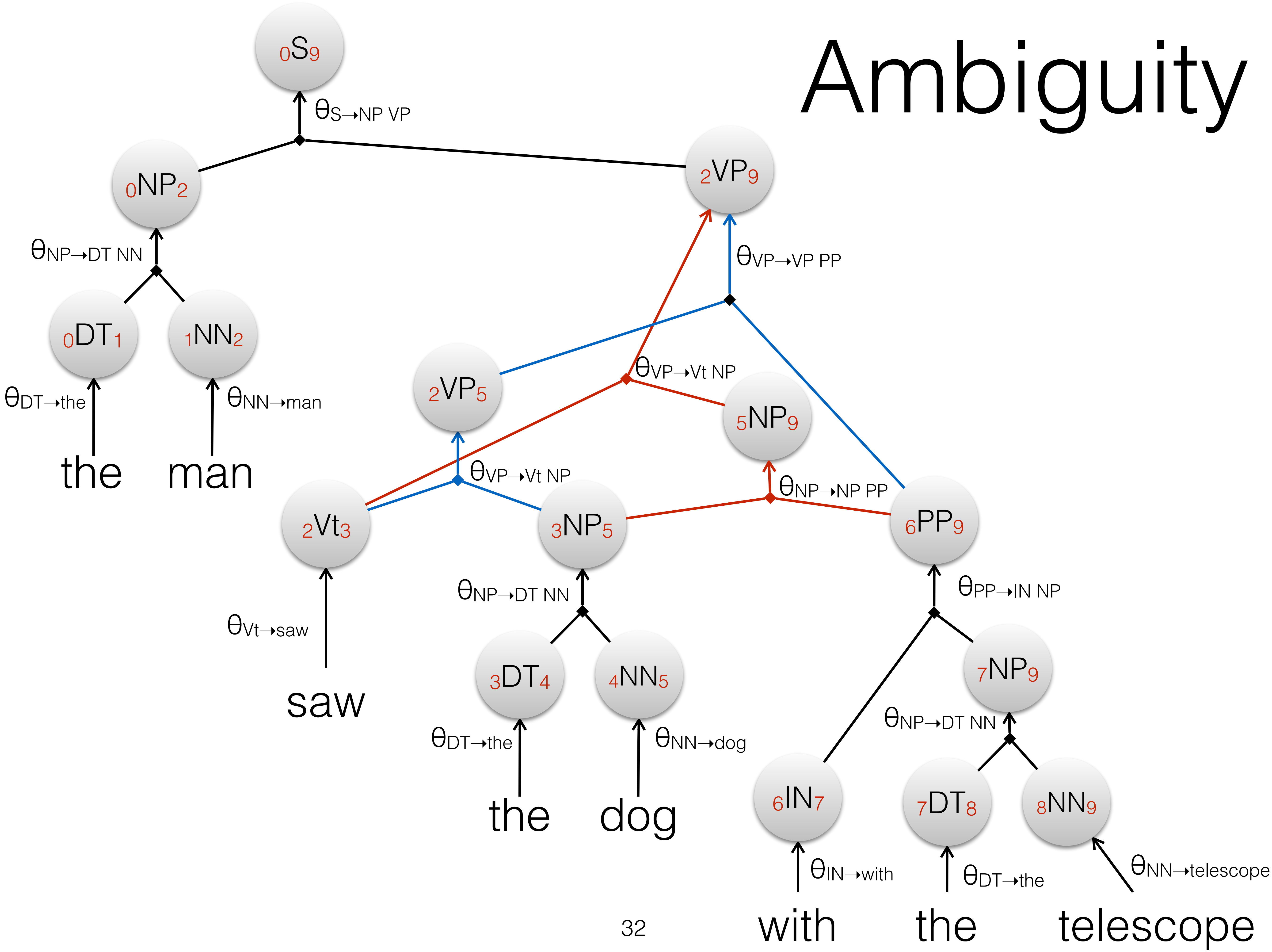
$_0$NP$_2$
θNP→DT NN
$_0$DT$_1$
$_1$NN$_2$
θDT→the
θNN→man
the
man
$_2$VP$_9$
θVP→VP PP
$_2$VP$_5$
$_5$NP$_9$
θVP→Vt NP
θNP→NP PP
$_2$Vt$_3$
$_3$NP$_5$
$_6$PP$_9$
θNP→DT NN
θPP→IN NP
θVt→saw
saw
$_3$DT$_4$
$_4$NN$_5$
$_7$NP$_9$
θNP→DT NN
θDT→the
θNN→dog
the
dog
$_6$IN$_7$
$_7$DT$_8$
$_8$NN$_9$
θIN→with
θDT→the
θNN→telescope
with
the
telescope

Ambiguity



$_{0}NP_{2}$
$\theta_{NP \rightarrow DT\ NN}$
$_{0}DT_{1}$
$_{1}NN_{2}$
$\theta_{DT \rightarrow the}$
$\theta_{NN \rightarrow man}$
the
man
$_{2}VP_{9}$
$\theta_{VP \rightarrow VP\ PP}$
$_{2}VP_{5}$
$\theta_{VP \rightarrow Vt\ NP}$
$_{5}NP_{9}$
$\theta_{VP \rightarrow Vt\ NP}$
$\theta_{NP \rightarrow NP\ PP}$
$_{2}Vt_{3}$
$_{3}NP_{5}$
$_{6}PP_{9}$
$\theta_{NP \rightarrow DT\ NN}$
$\theta_{PP \rightarrow IN\ NP}$
$\theta_{Vt \rightarrow saw}$
saw
$_{3}DT_{4}$
$_{4}NN_{5}$
$_{7}NP_{9}$
$\theta_{NP \rightarrow DT\ NN}$
$\theta_{DT \rightarrow the}$
$\theta_{NN \rightarrow dog}$
the
dog
$_{6}IN_{7}$
$_{7}DT_{8}$
$_{8}NN_{9}$
$\theta_{IN \rightarrow with}$
$\theta_{DT \rightarrow the}$
$\theta_{NN \rightarrow telescope}$
with
the
telescope

Ambiguity



$_0S_9$
$\theta_{S\rightarrow NP\ VP}$
$_0NP_2$
$_2VP_9$
$\theta_{NP\rightarrow DT\ NN}$
$\theta_{VP\rightarrow VP\ PP}$
$_0DT_1$
$_1NN_2$
$\theta_{VP\rightarrow Vt\ NP}$
$_2VP_5$
$\theta_{DT\rightarrow the}$
$\theta_{NN\rightarrow man}$
$_5NP_9$
the
man
$\theta_{VP\rightarrow Vt\ NP}$
$\theta_{NP\rightarrow NP\ PP}$
$_2Vt_3$
$_3NP_5$
$_6PP_9$
$\theta_{NP\rightarrow DT\ NN}$
$\theta_{PP\rightarrow IN\ NP}$
$\theta_{Vt\rightarrow saw}$
saw
$_3DT_4$
$_4NN_5$
$_7NP_9$
$\theta_{NP\rightarrow DT\ NN}$
$\theta_{DT\rightarrow the}$
$\theta_{NN\rightarrow dog}$
the
dog
$_6IN_7$
$_7DT_8$
$_8NN_9$
$\theta_{IN\rightarrow with}$
$\theta_{DT\rightarrow the}$
$\theta_{NN\rightarrow telescope}$
with
the
telescope

Complexity

Complexity

Item form: $[i, X \rightarrow \alpha \blacksquare \bullet \beta \square, j]$

Complexity

Item form: [i, X → α■•β□, j]

- Each rule segments the input w_1 .. w_n

Complexity

Item form: [i, X → α■•β□, j]

- Each rule segments the input w_1 .. w_n

Every CFG can be written in CNF (max arity = 2)

Complexity

Item form: [i, X → α■•β□, j]

- Each rule segments the input w_1 .. w_n

Every CFG can be written in CNF (max arity = 2)

- In total we get up to 3 indices ranging from 0 .. n

Complexity

Item form: [i, X → α■•β□, j]

- Each rule segments the input w_1 .. w_n

Every CFG can be written in CNF (max arity = 2)

- In total we get up to 3 indices ranging from 0 .. n
- $O(n^3)$ annotated rules

Bitext Parsing

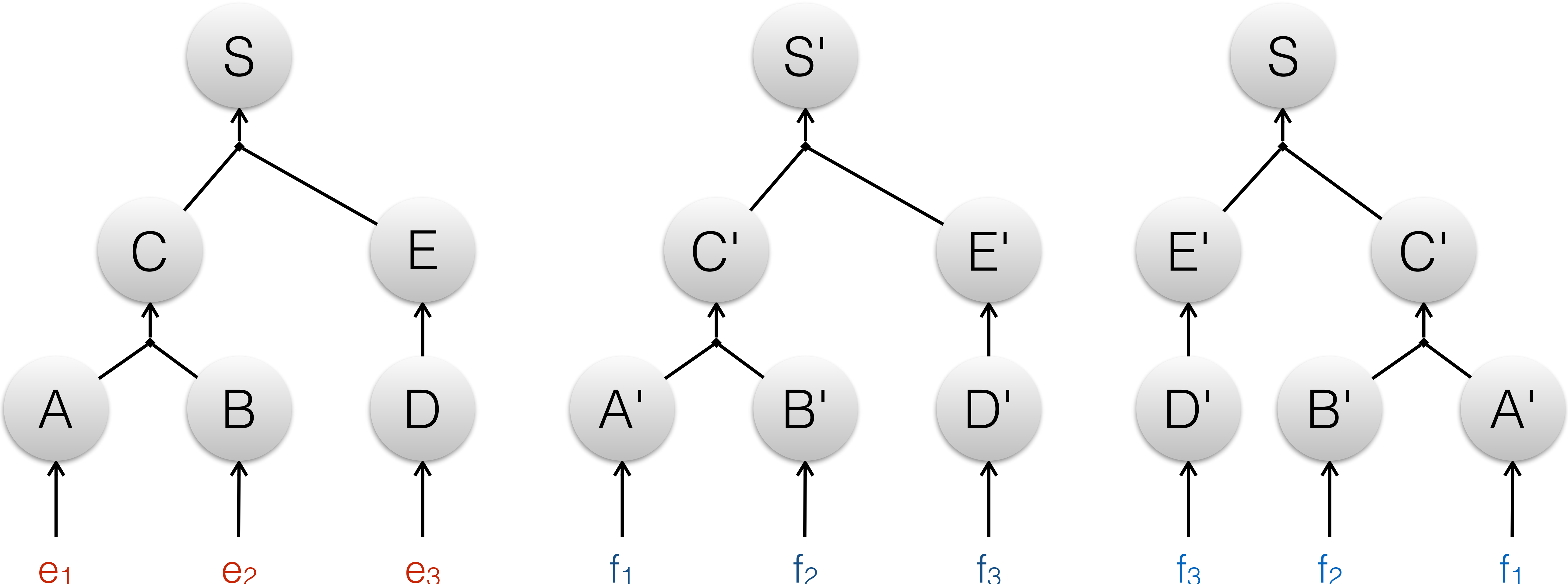
Imagine we have **two** streams of text

the man sleeps ⇔ dort l' homme

We want to parse both strings **simultaneously** such that their trees are **isomorphic**

- same structure up to
- relabelling and permutation of siblings

Isomorphic trees



Synchronous Grammar

A CFG **paired** with another

- RHS symbols map **one-to-one**

Synchronous Grammar

A CFG **paired** with another

- RHS symbols map **one-to-one**

English French

Synchronous Grammar

A CFG **paired** with another

- RHS symbols map **one-to-one**

	English	French	
X →	A	A	copy

Synchronous Grammar

A CFG **paired** with another

- RHS symbols map **one-to-one**

	English	French	
X →	A	A	copy
X →	B C	B C	copy

Synchronous Grammar

A CFG **paired** with another

- RHS symbols map **one-to-one**

	English	French	
X →	A	A	copy
X →	B C	B C	copy
		C B	invert

Synchronous Grammar

A CFG **paired** with another

- RHS symbols map **one-to-one**

	English	French	
X →	A	A	copy
X →	B C	B C	copy
		C B	invert
X →	e	f	transduce

Parse E

Parse with the English side of the grammar

${}_{0}\mathrm{S}_{3} \rightarrow {}_{0}\mathrm{NP}_{2}\ {}_{2}\mathrm{VP}_{3}$

${}_{0}\mathrm{NP}_{2} \rightarrow {}_{0}\mathrm{DT}_{1}\ {}_{1}\mathrm{NN}_{2}$

${}_{2}\mathrm{VP}_{3} \rightarrow {}_{2}\mathrm{Vi}_{3}$

${}_{0}\mathrm{DT}_{1} \rightarrow$ the

${}_{1}\mathrm{NN}_{2} \rightarrow$ man

${}_{2}\mathrm{Vi}_{3} \rightarrow$ sleeps

Projection

Projection

English	French

Projection

	English	French
${}_{0}S_{3} \rightarrow$	${}_{0}NP_{2}\ {}_{2}VP_{3}$	${}_{0}NP_{2}\ {}_{2}VP_{3}$

Projection

	English	French
${}_{0}S_{3} \rightarrow$	${}_{0}NP_{2}\ {}_{2}VP_{3}$	${}_{0}NP_{2}\ {}_{2}VP_{3}$
		${}_{2}VP_{3}\ {}_{0}NP_{2}$

Projection

	English	French
${}_0S_3 \rightarrow$	${}_0NP_2\ {}_2VP_3$	${}_0NP_2\ {}_2VP_3$
		${}_2VP_3\ {}_0NP_2$
${}_0NP_2 \rightarrow$	${}_0DT_1\ {}_1NN_2$	${}_0DT_1\ {}_1NN_2$

Projection

	English	French
${}_{0}\mathrm{S}_{3} \rightarrow$	${}_{0}\mathrm{NP}_{2}\ {}_{2}\mathrm{VP}_{3}$	${}_{0}\mathrm{NP}_{2}\ {}_{2}\mathrm{VP}_{3}$
		${}_{2}\mathrm{VP}_{3}\ {}_{0}\mathrm{NP}_{2}$
${}_{0}\mathrm{NP}_{2} \rightarrow$	${}_{0}\mathrm{DT}_{1}\ {}_{1}\mathrm{NN}_{2}$	${}_{0}\mathrm{DT}_{1}\ {}_{1}\mathrm{NN}_{2}$
		${}_{1}\mathrm{NN}_{2}\ {}_{0}\mathrm{DT}_{1}$

Projection

	English	French
${}_{0}\mathrm{S}_{3}$ →	${}_{0}\mathrm{NP}_{2}$ ${}_{2}\mathrm{VP}_{3}$	${}_{0}\mathrm{NP}_{2}$ ${}_{2}\mathrm{VP}_{3}$
		${}_{2}\mathrm{VP}_{3}$ ${}_{0}\mathrm{NP}_{2}$
${}_{0}\mathrm{NP}_{2}$ →	${}_{0}\mathrm{DT}_{1}$ ${}_{1}\mathrm{NN}_{2}$	${}_{0}\mathrm{DT}_{1}$ ${}_{1}\mathrm{NN}_{2}$
		${}_{1}\mathrm{NN}_{2}$ ${}_{0}\mathrm{DT}_{1}$
${}_{2}\mathrm{VP}_{3}$ →	${}_{2}\mathrm{Vi}_{3}$	${}_{2}\mathrm{Vi}_{3}$

Projection

	English	French
${}_0\mathrm{S}_3 \rightarrow$	${}_0\mathrm{NP}_2\ {}_2\mathrm{VP}_3$	${}_0\mathrm{NP}_2\ {}_2\mathrm{VP}_3$
		${}_2\mathrm{VP}_3\ {}_0\mathrm{NP}_2$
${}_0\mathrm{NP}_2 \rightarrow$	${}_0\mathrm{DT}_1\ {}_1\mathrm{NN}_2$	${}_0\mathrm{DT}_1\ {}_1\mathrm{NN}_2$
		${}_1\mathrm{NN}_2\ {}_0\mathrm{DT}_1$
${}_2\mathrm{VP}_3 \rightarrow$	${}_2\mathrm{Vi}_3$	${}_2\mathrm{Vi}_3$
${}_0\mathrm{DT}_1 \rightarrow$	the	le

Projection

	English	French
${}_{0}\mathrm{S}_{3} \rightarrow$	${}_{0}\mathrm{NP}_{2}\ {}_{2}\mathrm{VP}_{3}$	${}_{0}\mathrm{NP}_{2}\ {}_{2}\mathrm{VP}_{3}$
		${}_{2}\mathrm{VP}_{3}\ {}_{0}\mathrm{NP}_{2}$
${}_{0}\mathrm{NP}_{2} \rightarrow$	${}_{0}\mathrm{DT}_{1}\ {}_{1}\mathrm{NN}_{2}$	${}_{0}\mathrm{DT}_{1}\ {}_{1}\mathrm{NN}_{2}$
		${}_{1}\mathrm{NN}_{2}\ {}_{0}\mathrm{DT}_{1}$
${}_{2}\mathrm{VP}_{3} \rightarrow$	${}_{2}\mathrm{Vi}_{3}$	${}_{2}\mathrm{Vi}_{3}$
${}_{0}\mathrm{DT}_{1} \rightarrow$	the	le
		la

Projection

	English	French
${}_{0}\mathrm{S}_{3} \rightarrow$	${}_{0}\mathrm{NP}_{2}\ {}_{2}\mathrm{VP}_{3}$	${}_{0}\mathrm{NP}_{2}\ {}_{2}\mathrm{VP}_{3}$
		${}_{2}\mathrm{VP}_{3}\ {}_{0}\mathrm{NP}_{2}$
${}_{0}\mathrm{NP}_{2} \rightarrow$	${}_{0}\mathrm{DT}_{1}\ {}_{1}\mathrm{NN}_{2}$	${}_{0}\mathrm{DT}_{1}\ {}_{1}\mathrm{NN}_{2}$
		${}_{1}\mathrm{NN}_{2}\ {}_{0}\mathrm{DT}_{1}$
${}_{2}\mathrm{VP}_{3} \rightarrow$	${}_{2}\mathrm{Vi}_{3}$	${}_{2}\mathrm{Vi}_{3}$
${}_{0}\mathrm{DT}_{1} \rightarrow$	the	le
		la
		l'

Projection

	English	French
${}_0S_3 \rightarrow$	${}_0NP_2\ {}_2VP_3$	${}_0NP_2\ {}_2VP_3$
		${}_2VP_3\ {}_0NP_2$
${}_0NP_2 \rightarrow$	${}_0DT_1\ {}_1NN_2$	${}_0DT_1\ {}_1NN_2$
		${}_1NN_2\ {}_0DT_1$
${}_2VP_3 \rightarrow$	${}_2Vi_3$	${}_2Vi_3$
${}_0DT_1 \rightarrow$	the	le
		la
		l'
${}_1NN_2 \rightarrow$	man	homme

Projection

	English	French
${}_0\mathrm{S}_3 \rightarrow$	${}_0\mathrm{NP}_2\ {}_2\mathrm{VP}_3$	${}_0\mathrm{NP}_2\ {}_2\mathrm{VP}_3$
		${}_2\mathrm{VP}_3\ {}_0\mathrm{NP}_2$
${}_0\mathrm{NP}_2 \rightarrow$	${}_0\mathrm{DT}_1\ {}_1\mathrm{NN}_2$	${}_0\mathrm{DT}_1\ {}_1\mathrm{NN}_2$
		${}_1\mathrm{NN}_2\ {}_0\mathrm{DT}_1$
${}_2\mathrm{VP}_3 \rightarrow$	${}_2\mathrm{Vi}_3$	${}_2\mathrm{Vi}_3$
${}_0\mathrm{DT}_1 \rightarrow$	the	le
		la
		l'
${}_1\mathrm{NN}_2 \rightarrow$	man	homme
${}_2\mathrm{Vi}_3 \rightarrow$	sleeps	dort

French Grammar

	French
${}_0S_3 \rightarrow$	${}_0NP_2\ {}_2VP_3$
	${}_2VP_3\ {}_0NP_2$
${}_0NP_2 \rightarrow$	${}_0DT_1\ {}_1NN_2$
	${}_1NN_2\ {}_0DT_1$
${}_2VP_3 \rightarrow$	${}_2Vi_3$
${}_0DT_1 \rightarrow$	le
	la
	l'
${}_1NN_2 \rightarrow$	homme
${}_2Vi_3 \rightarrow$	dort

Parse F

	French
${}_{0}\mathrm{S}_{3} \rightarrow$	${}_{0}\mathrm{NP}_{2}\ {}_{2}\mathrm{VP}_{3}$
	${}_{2}\mathrm{VP}_{3}\ {}_{0}\mathrm{NP}_{2}$
${}_{0}\mathrm{NP}_{2} \rightarrow$	${}_{0}\mathrm{DT}_{1}\ {}_{1}\mathrm{NN}_{2}$
	${}_{1}\mathrm{NN}_{2}\ {}_{0}\mathrm{DT}_{1}$
${}_{2}\mathrm{VP}_{3} \rightarrow$	${}_{2}\mathrm{Vi}_{3}$
${}_{0}\mathrm{DT}_{1} \rightarrow$	le
	la
	l'
${}_{1}\mathrm{NN}_{2} \rightarrow$	homme
${}_{2}\mathrm{Vi}_{3} \rightarrow$	dort

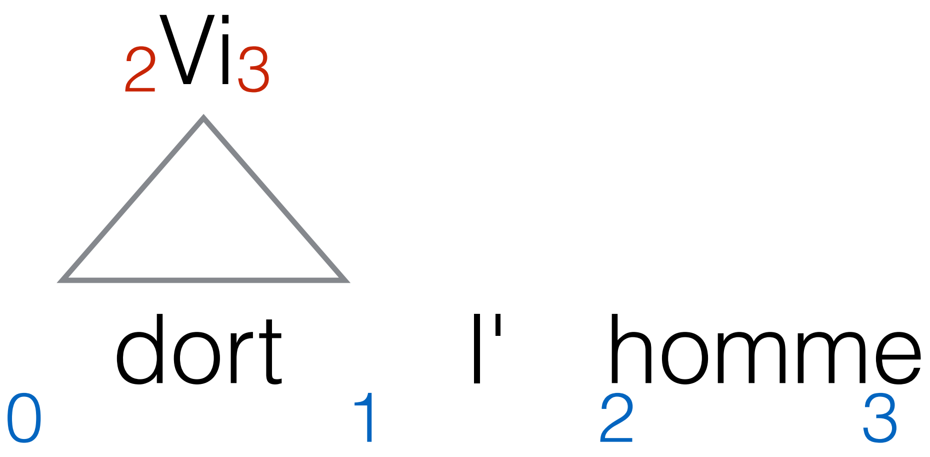
Parse F

	French
${}_0S_3 \rightarrow$	${}_0NP_2\ {}_2VP_3$
	${}_2VP_3\ {}_0NP_2$
${}_0NP_2 \rightarrow$	${}_0DT_1\ {}_1NN_2$
	${}_1NN_2\ {}_0DT_1$
${}_2VP_3 \rightarrow$	${}_2Vi_3$
${}_0DT_1 \rightarrow$	le
	la
	l'
${}_1NN_2 \rightarrow$	homme
${}_2Vi_3 \rightarrow$	dort

$_0$ dort $_1$ l' $_2$ homme $_3$

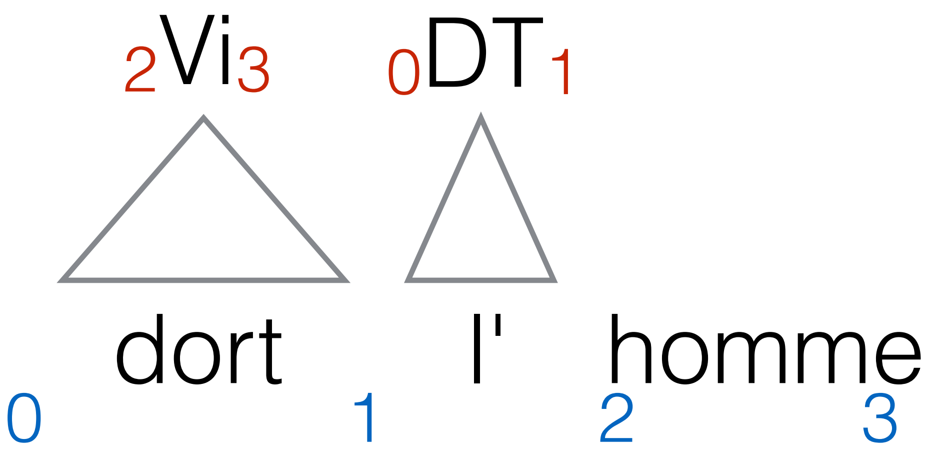
Parse F

	French
$_0S_3$ →	$_0NP_2\ _2VP_3$
	$_2VP_3\ _0NP_2$
$_0NP_2$ →	$_0DT_1\ _1NN_2$
	$_1NN_2\ _0DT_1$
$_2VP_3$ →	$_2Vi_3$
$_0DT_1$ →	le
	la
	l'
$_1NN_2$ →	homme
$_2Vi_3$ →	dort



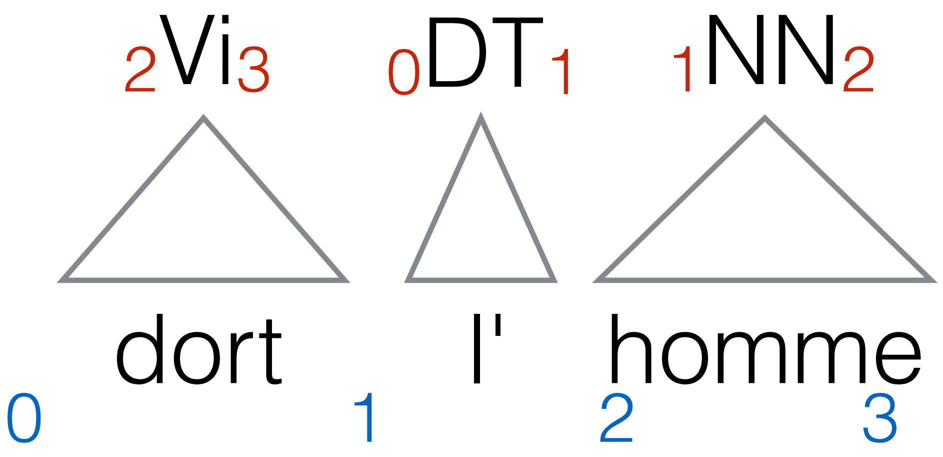
Parse F

	French
${}_0\mathrm{S}_3 \rightarrow$	${}_0\mathrm{NP}_2\ {}_2\mathrm{VP}_3$
	${}_2\mathrm{VP}_3\ {}_0\mathrm{NP}_2$
${}_0\mathrm{NP}_2 \rightarrow$	${}_0\mathrm{DT}_1\ {}_1\mathrm{NN}_2$
	${}_1\mathrm{NN}_2\ {}_0\mathrm{DT}_1$
${}_2\mathrm{VP}_3 \rightarrow$	${}_2\mathrm{Vi}_3$
${}_0\mathrm{DT}_1 \rightarrow$	le
	la
	l'
${}_1\mathrm{NN}_2 \rightarrow$	homme
${}_2\mathrm{Vi}_3 \rightarrow$	dort



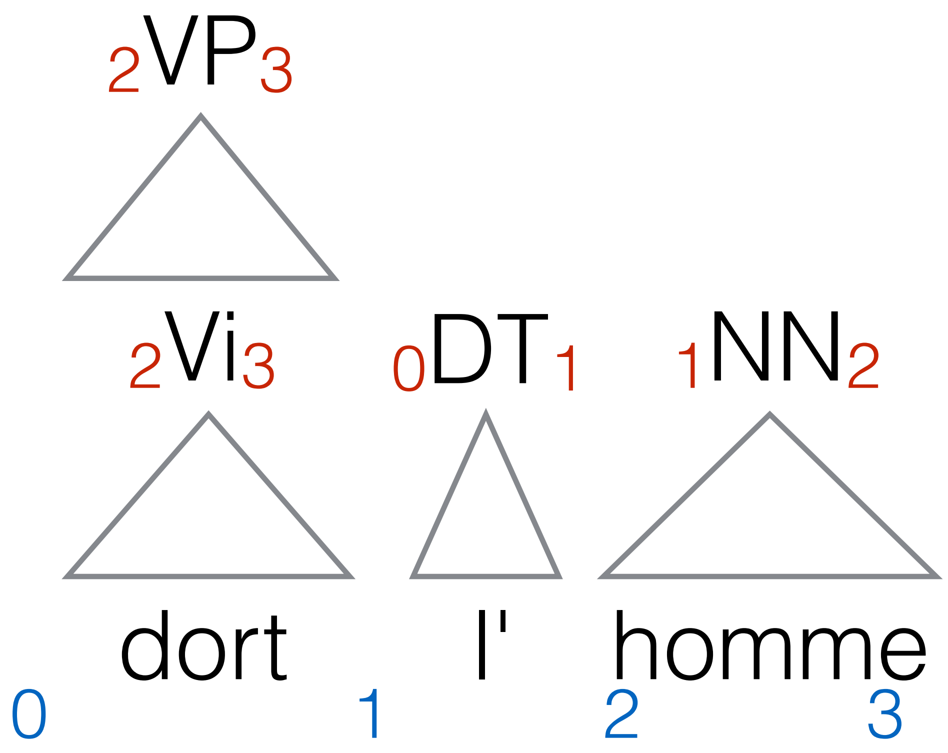
Parse F

	French
${}_0S_3 \rightarrow$	${}_0NP_2\ {}_2VP_3$
	${}_2VP_3\ {}_0NP_2$
${}_0NP_2 \rightarrow$	${}_0DT_1\ {}_1NN_2$
	${}_1NN_2\ {}_0DT_1$
${}_2VP_3 \rightarrow$	${}_2Vi_3$
${}_0DT_1 \rightarrow$	le
	la
	l'
${}_1NN_2 \rightarrow$	homme
${}_2Vi_3 \rightarrow$	dort



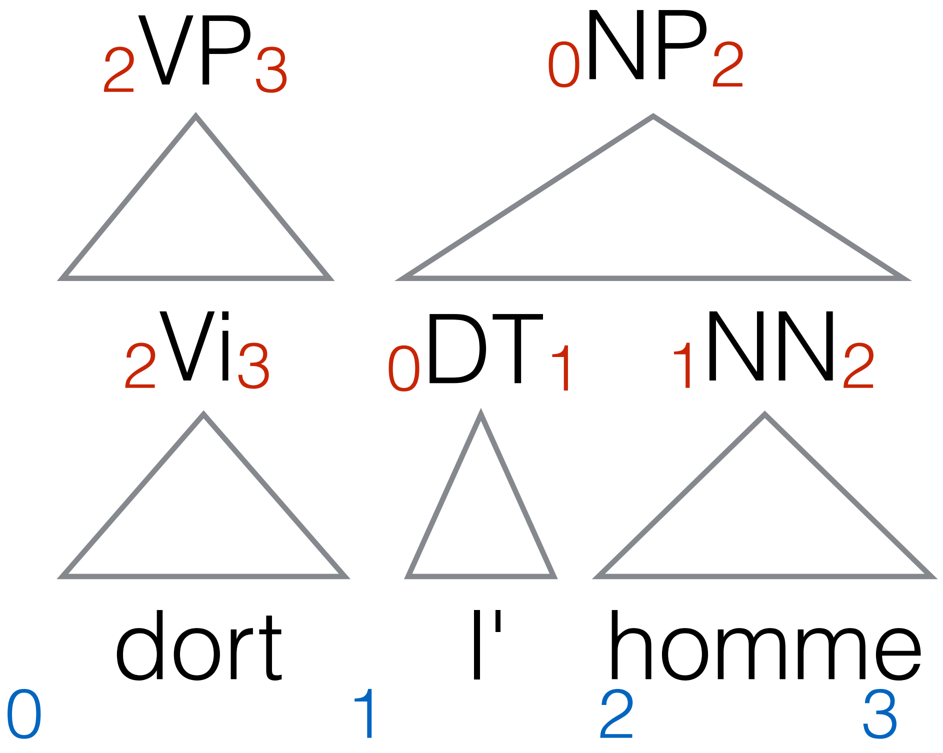
Parse F

	French
${}_0S_3$ →	${}_0NP_2$ ${}_2VP_3$
	${}_2VP_3$ ${}_0NP_2$
${}_0NP_2$ →	${}_0DT_1$ ${}_1NN_2$
	${}_1NN_2$ ${}_0DT_1$
${}_2VP_3$ →	${}_2Vi_3$
${}_0DT_1$ →	le
	la
	l'
${}_1NN_2$ →	homme
${}_2Vi_3$ →	dort



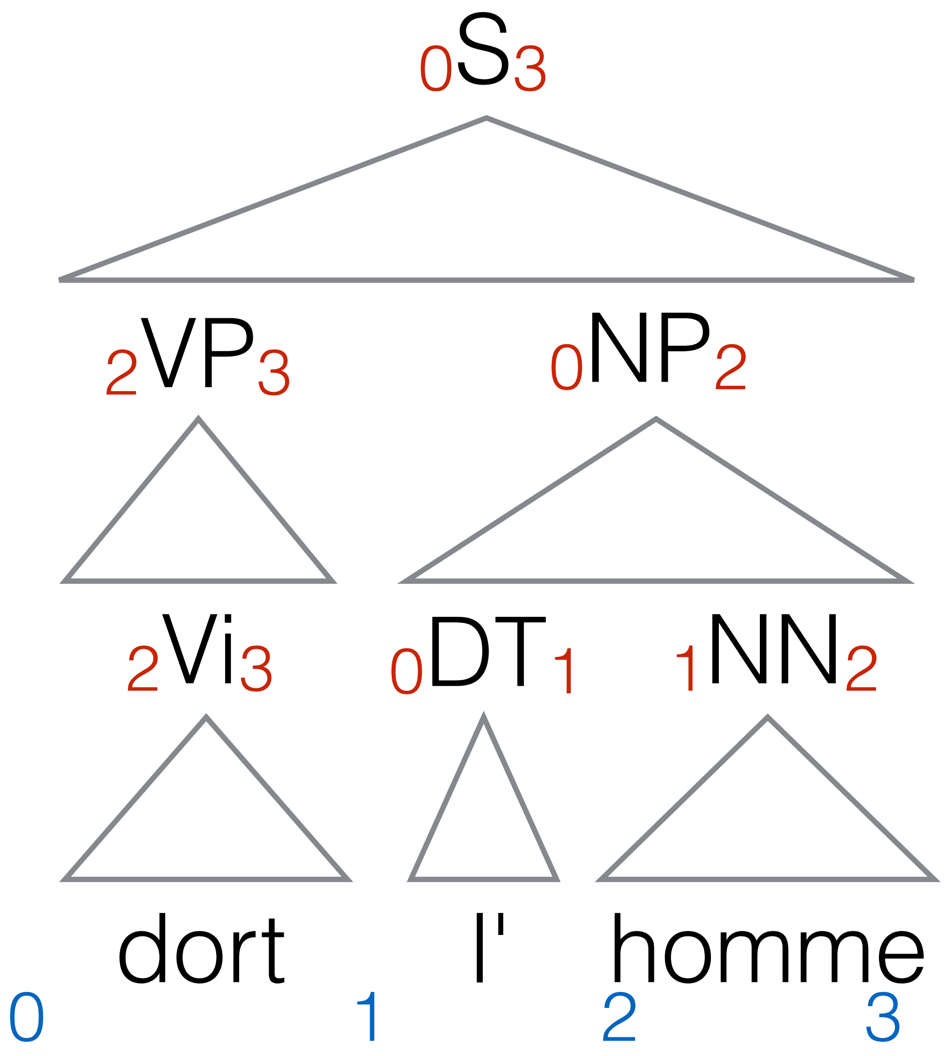
Parse F

	French
${}_0\mathrm{S}_3 \rightarrow$	${}_0\mathrm{NP}_2\ {}_2\mathrm{VP}_3$
	${}_2\mathrm{VP}_3\ {}_0\mathrm{NP}_2$
${}_0\mathrm{NP}_2 \rightarrow$	${}_0\mathrm{DT}_1\ {}_1\mathrm{NN}_2$
	${}_1\mathrm{NN}_2\ {}_0\mathrm{DT}_1$
${}_2\mathrm{VP}_3 \rightarrow$	${}_2\mathrm{Vi}_3$
${}_0\mathrm{DT}_1 \rightarrow$	le
	la
	l'
${}_1\mathrm{NN}_2 \rightarrow$	homme
${}_2\mathrm{Vi}_3 \rightarrow$	dort



Parse F

	French
${}_0S_3 \rightarrow$	${}_0NP_2$ ${}_2VP_3$
	${}_2VP_3$ ${}_0NP_2$
${}_0NP_2 \rightarrow$	${}_0DT_1$ ${}_1NN_2$
	${}_1NN_2$ ${}_0DT_1$
${}_2VP_3 \rightarrow$	${}_2Vi_3$
${}_0DT_1 \rightarrow$	le
	la
	l'
${}_1NN_2 \rightarrow$	homme
${}_2Vi_3 \rightarrow$	dort



Cascade of Monolingual Parsers

Cascade of Monolingual Parsers

CFG parsing can be seen as intersecting a CFG and an FSA [Bar-Hillel, 1961; Billot and Lang, 1989]

Cascade of Monolingual Parsers

CFG parsing can be seen as intersecting a CFG and an FSA [Bar-Hillel, 1961; Billot and Lang, 1989]

CFGs are closed under intersection [Hopcroft and Ullman, 1979]

Cascade of Monolingual Parsers

CFG parsing can be seen as intersecting a CFG and an FSA [Bar-Hillel, 1961; Billot and Lang, 1989]

CFGs are closed under intersection [Hopcroft and Ullman, 1979]

- L(CFG) ∩ L(FSA) is a context-free language

Cascade of Monolingual Parsers

CFG parsing can be seen as intersecting a CFG and an FSA [Bar-Hillel, 1961; Billot and Lang, 1989]

CFGs are closed under intersection [Hopcroft and Ullman, 1979]

- L(CFG) ∩ L(FSA) is a context-free language

This neat property makes cascading intersection operations (parsers) appealing [Dyer, 2010]

Cascade of Monolingual Parsers

CFG parsing can be seen as intersecting a CFG and an FSA [Bar-Hillel, 1961; Billot and Lang, 1989]

CFGs are closed under intersection [Hopcroft and Ullman, 1979]

- L(CFG) ∩ L(FSA) is a context-free language

This neat property makes cascading intersection operations (parsers) appealing [Dyer, 2010]

- e.g. bitext parsing

Biproduct: alignments

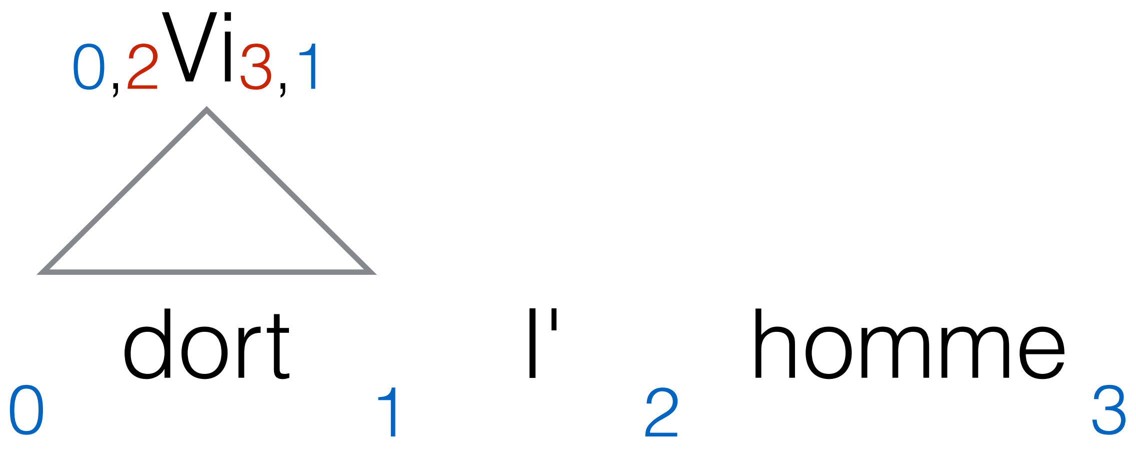
	French
${}_0S_3 \rightarrow$	${}_0NP_2\ {}_2VP_3$
	${}_2VP_3\ {}_0NP_2$
${}_0NP_2 \rightarrow$	${}_0DT_1\ {}_1NN_2$
	${}_1NN_2\ {}_0DT_1$
${}_2VP_3 \rightarrow$	${}_2Vi_3$
${}_0DT_1 \rightarrow$	le
	la
	l'
${}_1NN_2 \rightarrow$	${}_1NN_2 \rightarrow$ homme
${}_2Vi_3 \rightarrow$	${}_2Vi_3 \rightarrow$ dort

Biproduct: alignments

${}_0$ dort ${}_1$ l' ${}_2$ homme ${}_3$

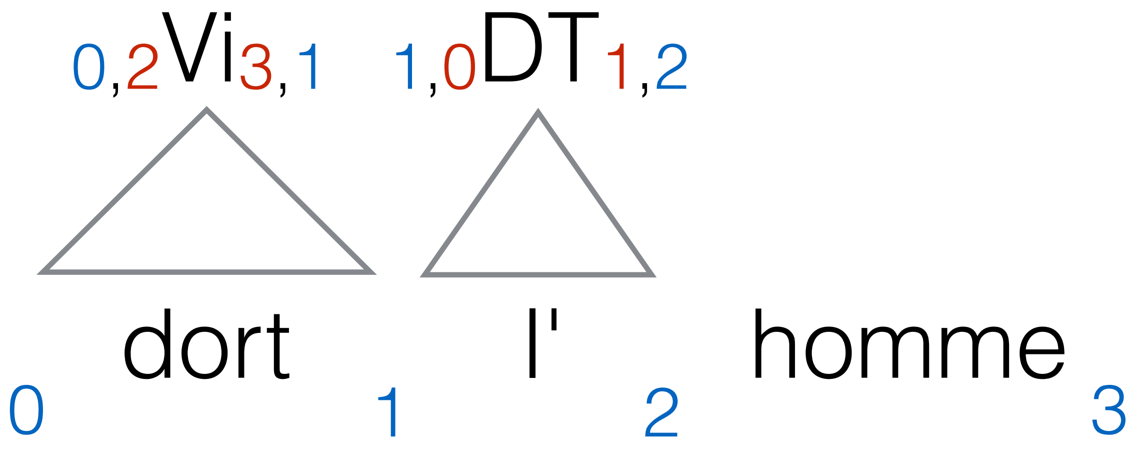
	French
${}_0\mathrm{S}_3 \rightarrow$	${}_0\mathrm{NP}_2\ {}_2\mathrm{VP}_3$
	${}_2\mathrm{VP}_3\ {}_0\mathrm{NP}_2$
${}_0\mathrm{NP}_2 \rightarrow$	${}_0\mathrm{DT}_1\ {}_1\mathrm{NN}_2$
	${}_1\mathrm{NN}_2\ {}_0\mathrm{DT}_1$
${}_2\mathrm{VP}_3 \rightarrow$	${}_2\mathrm{Vi}_3$
${}_0\mathrm{DT}_1 \rightarrow$	le
	la
	l'
${}_1\mathrm{NN}_2 \rightarrow$	${}_1\mathrm{NN}_2 \rightarrow$ homme
${}_2\mathrm{Vi}_3 \rightarrow$	${}_2\mathrm{Vi}_3 \rightarrow$ dort

Biproduct: alignments



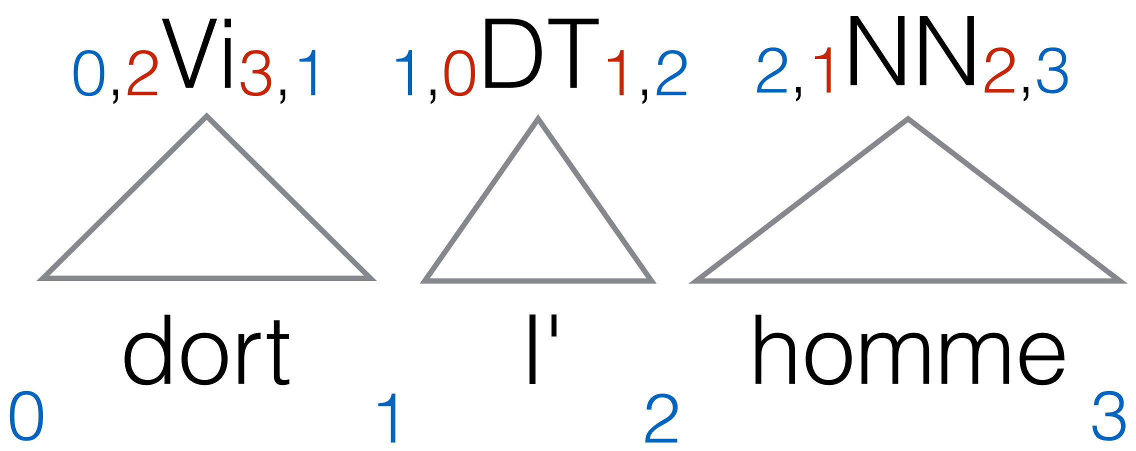
	French
$_0S_3 \rightarrow$	$_0NP_2\ _2VP_3$
	$_2VP_3\ _0NP_2$
$_0NP_2 \rightarrow$	$_0DT_1\ _1NN_2$
	$_1NN_2\ _0DT_1$
$_2VP_3 \rightarrow$	$_2Vi_3$
$_0DT_1 \rightarrow$	le
	la
	l'
$_1NN_2 \rightarrow$	$_1NN_2 \rightarrow$ homme
$_2Vi_3 \rightarrow$	$_2Vi_3 \rightarrow$ dort

Biproduct: alignments



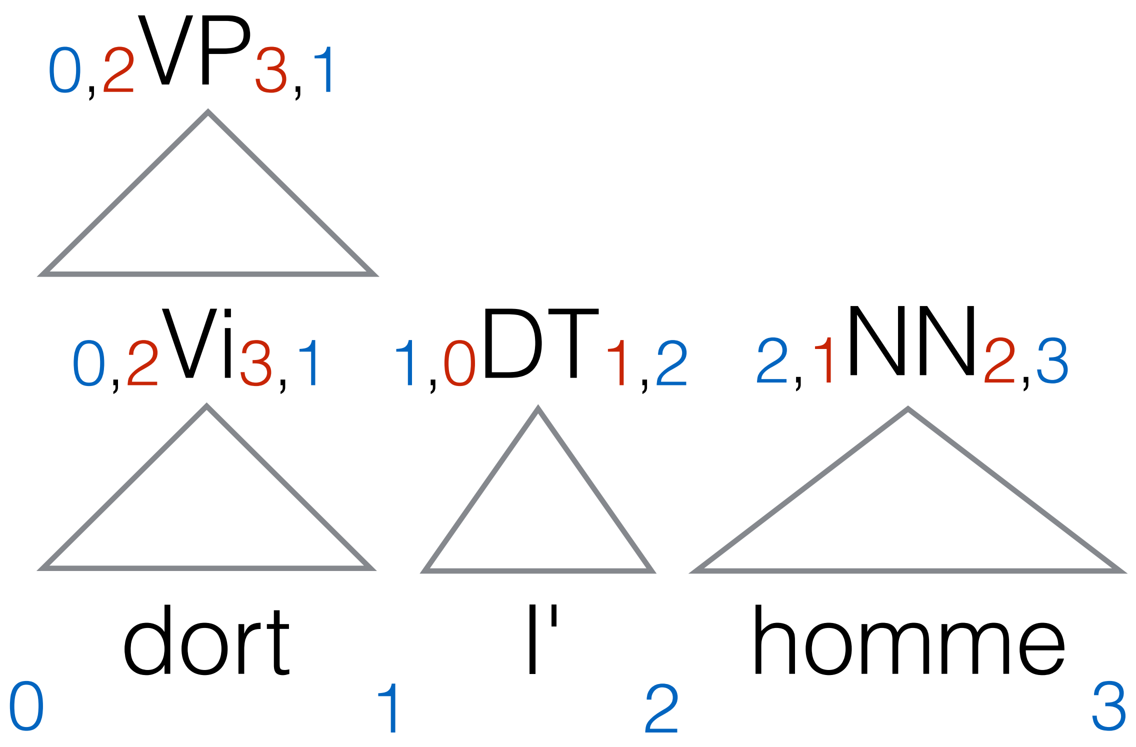
	French
$_0\mathrm{S}_3$ →	$_0\mathrm{NP}_2\ _2\mathrm{VP}_3$
	$_2\mathrm{VP}_3\ _0\mathrm{NP}_2$
$_0\mathrm{NP}_2$ →	$_0\mathrm{DT}_1\ _1\mathrm{NN}_2$
	$_1\mathrm{NN}_2\ _0\mathrm{DT}_1$
$_2\mathrm{VP}_3$ →	$_2\mathrm{Vi}_3$
$_0\mathrm{DT}_1$ →	le
	la
	l'
$_1\mathrm{NN}_2$ →	$_1\mathrm{NN}_2$ → homme
$_2\mathrm{Vi}_3$ →	$_2\mathrm{Vi}_3$ → dort

Biproduct: alignments


${}_{0,2}\mathrm{Vi}_{3,1}$
${}_{1,0}\mathrm{DT}_{1,2}$
${}_{2,1}\mathrm{NN}_{2,3}$
dort
l'
homme
0
1
2
3

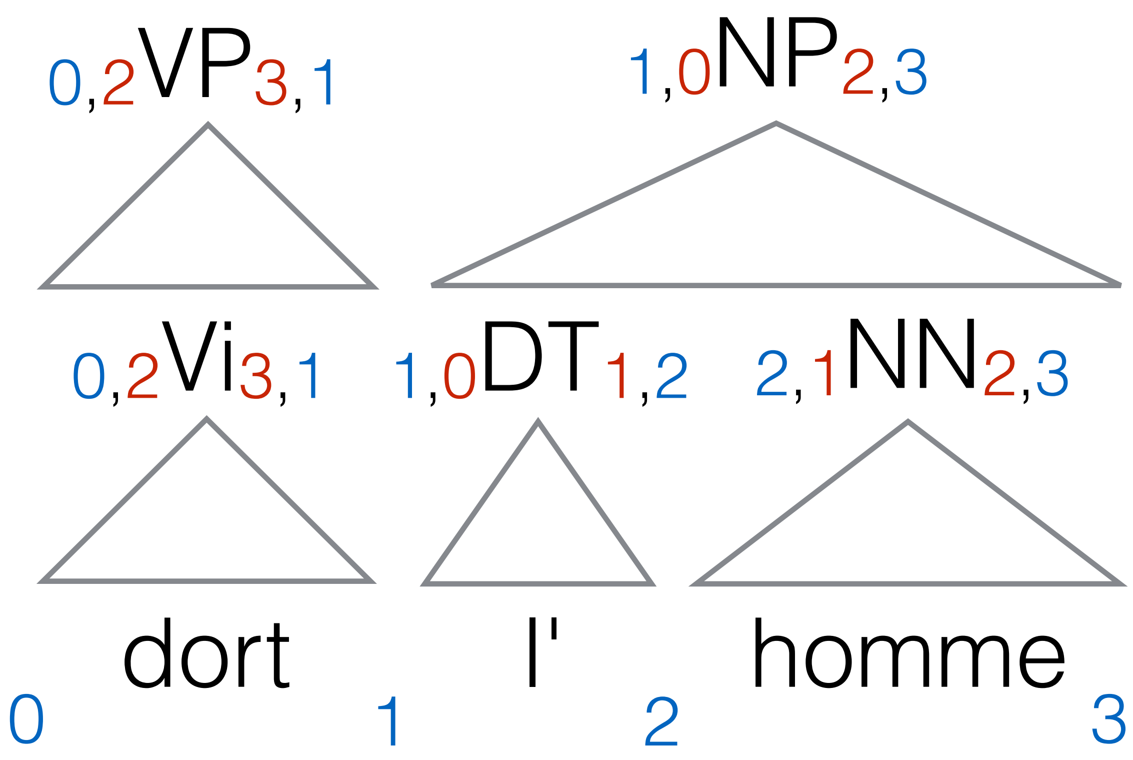
	French
${}_0\mathrm{S}_3 \rightarrow$	${}_0\mathrm{NP}_2\ {}_2\mathrm{VP}_3$
	${}_2\mathrm{VP}_3\ {}_0\mathrm{NP}_2$
${}_0\mathrm{NP}_2 \rightarrow$	${}_0\mathrm{DT}_1\ {}_1\mathrm{NN}_2$
	${}_1\mathrm{NN}_2\ {}_0\mathrm{DT}_1$
${}_2\mathrm{VP}_3 \rightarrow$	${}_2\mathrm{Vi}_3$
${}_0\mathrm{DT}_1 \rightarrow$	le
	la
	l'
${}_1\mathrm{NN}_2 \rightarrow$	${}_1\mathrm{NN}_2 \rightarrow$ homme
${}_2\mathrm{Vi}_3 \rightarrow$	${}_2\mathrm{Vi}_3 \rightarrow$ dort

Biproduct: alignments



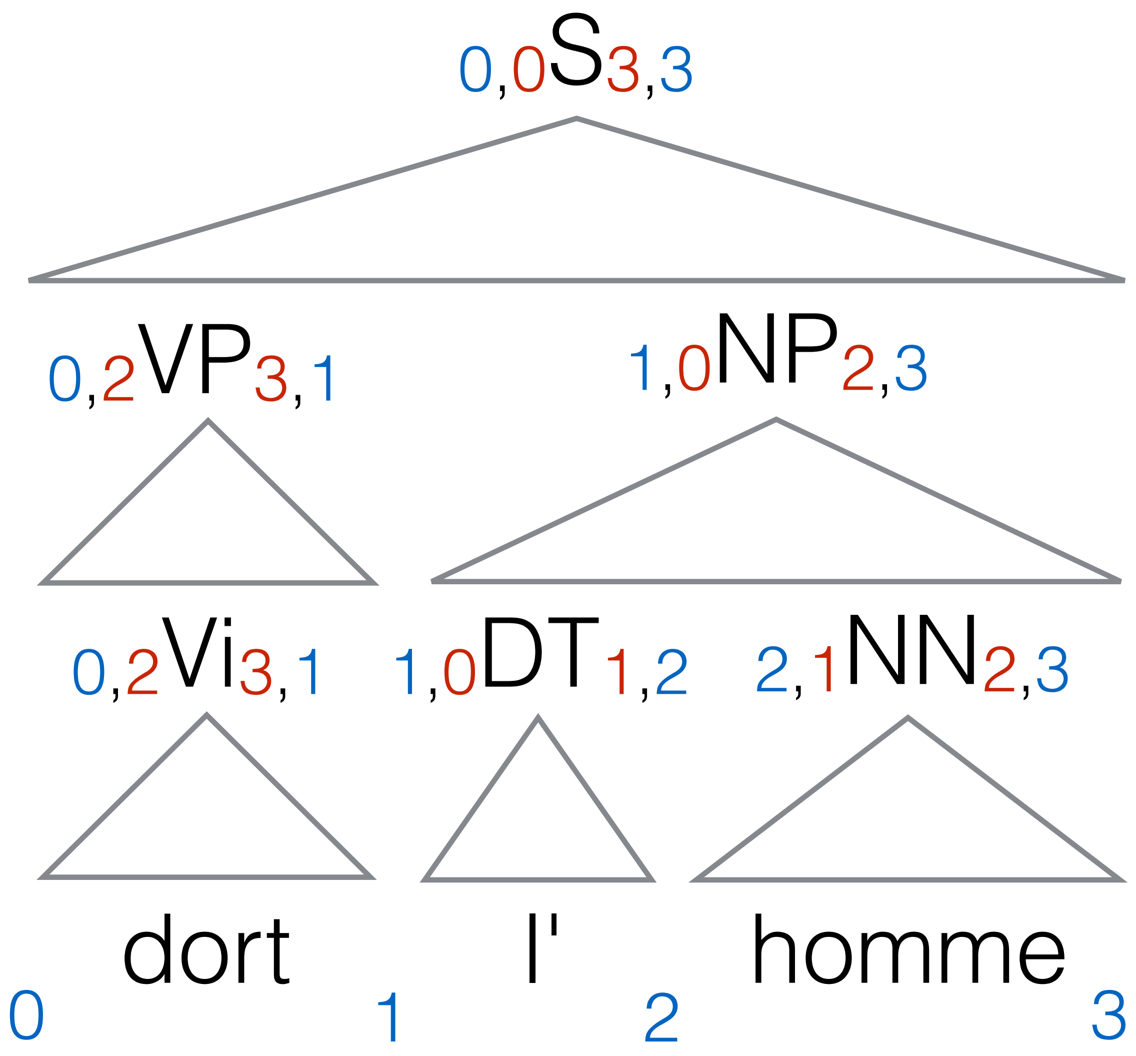
	French
${}_0S_3 \to$	${}_0NP_2\ {}_2VP_3$
	${}_2VP_3\ {}_0NP_2$
${}_0NP_2 \to$	${}_0DT_1\ {}_1NN_2$
	${}_1NN_2\ {}_0DT_1$
${}_2VP_3 \to$	${}_2Vi_3$
${}_0DT_1 \to$	le
	la
	l'
${}_1NN_2 \to$	${}_1NN_2 \to$ homme
${}_2Vi_3 \to$	${}_2Vi_3 \to$ dort

Biproduct: alignments



	French
$_0S_3$ →	$_0NP_2\ _2VP_3$
	$_2VP_3\ _0NP_2$
$_0NP_2$ →	$_0DT_1\ _1NN_2$
	$_1NN_2\ _0DT_1$
$_2VP_3$ →	$_2Vi_3$
$_0DT_1$ →	le
	la
	l'
$_1NN_2$ →	$_1NN_2$ → homme
$_2Vi_3$ →	$_2Vi_3$ → dort

Biproduct: alignments



	French
$_0S_3 \rightarrow$	$_0NP_2\ _2VP_3$
	$_2VP_3\ _0NP_2$
$_0NP_2 \rightarrow$	$_0DT_1\ _1NN_2$
	$_1NN_2\ _0DT_1$
$_2VP_3 \rightarrow$	$_2Vi_3$
$_0DT_1 \rightarrow$	le
	la
	l'
$_1NN_2 \rightarrow$	$_1NN_2 \rightarrow$ homme
$_2Vi_3 \rightarrow$	$_2Vi_3 \rightarrow$ dort

Complexity

- $O(l^3 \times m^3)$
    - where l is the length of the English string
    - and m is the length of the French string
- Joint parsing or cascade of parsers has the same theoretical complexity
    - Can cascading be more efficient on average? Why?

Bibliography

- Hopcroft, John E. and Ullman, Jeffrey D. 1979. Introduction To Automata Theory, Languages, And Computation.

- Shieber, S. and Schabes, Y. and Pereira, F. 1995. Principles and implementation of deductive parsing. In *Journal of Logic Programming*

- Bar-Hillel, Y. and Perles, M. and Shamir, E. 1961. On formal properties of simple phrase structure grammars.

- Billot, S. and Lang, B. 1989. The Structure of Shared Forests in Ambiguous ParsingThe Structure of Shared Forests in Ambiguous Parsing. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*.

- Dyer, C. 2010. Two monolingual parses are better than one (synchronous parse). In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.