Master in Artificial Intelligence

Trees and Grammars

Constituency Parsing

Dependency Parsing

Advanced Human Language Technologies

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

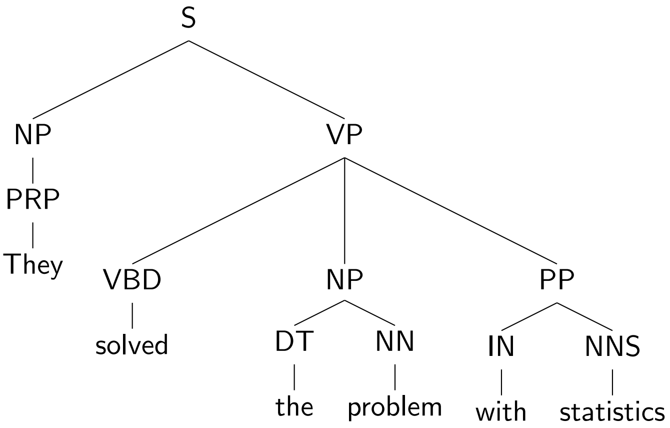
Outline

1 Trees and Grammars

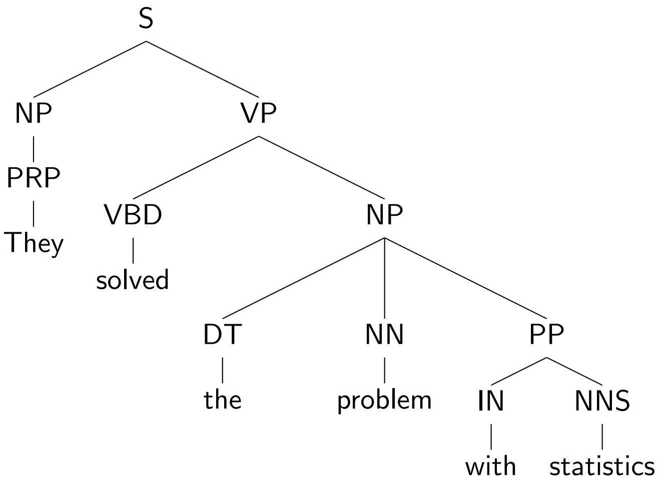
2 Constituency Parsing
- CKY Algorithm
- Earley Algorithm

3 Dependency Parsing
- Dependency Trees
- Arc-factored Dependency Parsing
- Parsing Projective Structures
- Parsing non-Projective Structures
- Transition-Based parsers

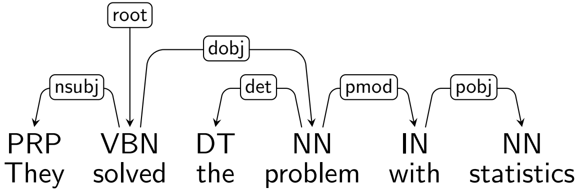
A Syntactic Tree



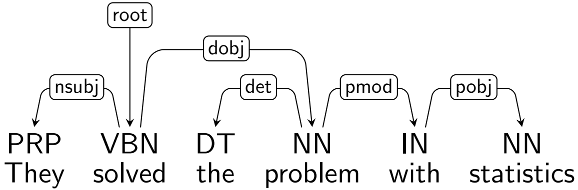
Another Syntactic Tree

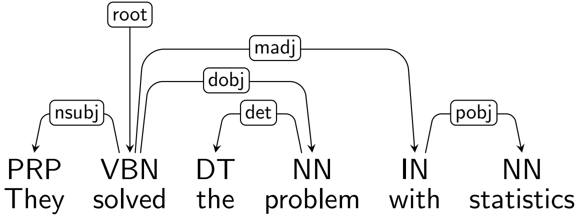


Dependency Trees

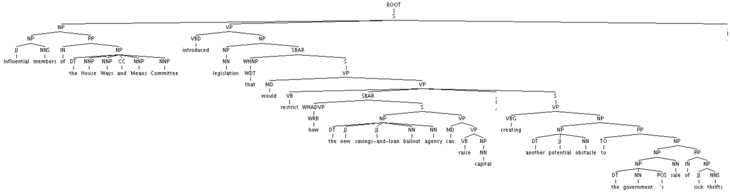


Dependency Trees





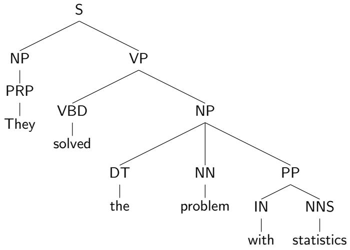
A “real” sentence



Influential members of the House Ways and Means Committee introduced legislation that would restrict how the new savings-and-loan bailout agency can raise capital, creating another potential obstacle to the government's sale of sick thrifts.

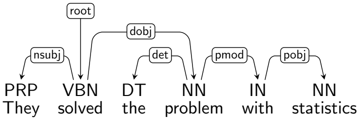
Theories of Syntactic Structure

Constituent Trees



- Main element: constituents (or phrases, or bracketings)
- Constituents = abstract linguistic units
- Results in nested trees

Dependency Trees



- Main element: dependency
- Focus on relations between words
- Handles *free word order* nicely.

Context Free Grammars (CFGs)

A context-free grammar is defined as a tuple $G = \langle N, \Sigma, R, S \rangle$ where:

- N is a set of non-terminal symbols
- $S \in N$ is a distinguished start symbol
- Σ is a set of terminal symbols
- R is a set of rules of the form $X \to Y_1 Y_2 \ldots Y_n$ where $n \geq 0,\ X \in N,\ Y_i \in N \cup \Sigma$

Context Free Grammars, Example

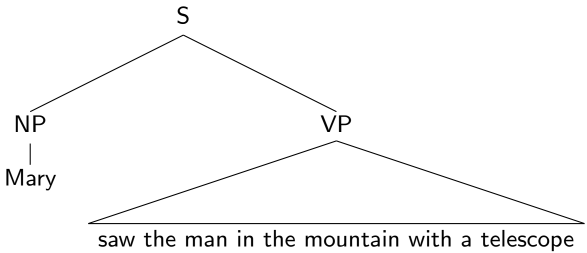
$$
\begin{aligned}
N &= \{\mathrm{S}, \mathrm{VP}, \mathrm{NP}, \mathrm{PP}, \mathrm{DT}, \mathrm{Vi}, \mathrm{Vt}, \mathrm{NN}, \mathrm{IN}\}^{1} \\
S &= \{\mathrm{S}\} \\
\Sigma &= \{\mathrm{sleeps}, \mathrm{saw}, \mathrm{man}, \mathrm{woman}, \mathrm{telescope}, \mathrm{the}, \mathrm{with}, \mathrm{in}\} \\
R &= \left\{ \begin{array}{ll}
\mathrm{S} \rightarrow \mathrm{NP\ VP} & \mathrm{Vi} \rightarrow \mathrm{sleeps} \\
\mathrm{NP} \rightarrow \mathrm{DT\ NN} & \mathrm{Vt} \rightarrow \mathrm{saw} \\
\mathrm{NP} \rightarrow \mathrm{NP\ PP} & \mathrm{NN} \rightarrow \mathrm{man} \\
\mathrm{PP} \rightarrow \mathrm{IN\ NP} & \mathrm{NN} \rightarrow \mathrm{woman} \\
\mathrm{VP} \rightarrow \mathrm{Vi} & \mathrm{NN} \rightarrow \mathrm{telescope} \\
\mathrm{VP} \rightarrow \mathrm{Vt\ NP} & \mathrm{DT} \rightarrow \mathrm{the} \\
\mathrm{VP} \rightarrow \mathrm{VP\ PP} & \mathrm{IN} \rightarrow \mathrm{with} \\
 & \mathrm{IN} \rightarrow \mathrm{in}
\end{array} \right\}
\end{aligned}
$$

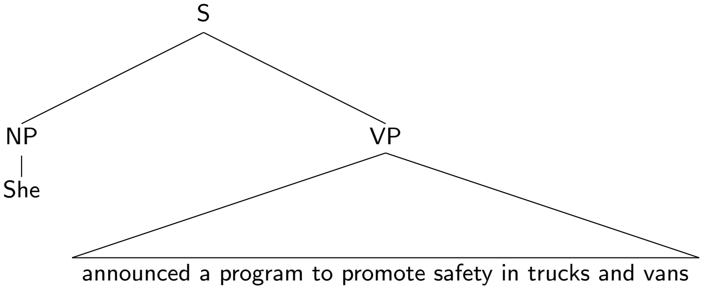
[1] S=sentence, VP=verb phrase, NP=noun phrase, PP=prepositional phrase, DT=determiner, Vi=intransitive verb, Vt=transitive verb, NN=noun, IN=preposition

Properties of CFGs

- A CFG defines a set of possible *derivations* (i.e. unique trees)
- A sequence of terminals $s \in \Sigma^*$ is *generated* by the CFG (or *recognized* by it, or *belongs* to the language defined by it) if there is at least a derivation that produces s.
- Some sequences of terminals generated by the CFG may have more than one derivation (*ambiguity*).

Ambiguity


S
NP
VP
Mary
saw the man in the mountain with a telescope


S
NP
VP
She
announced a program to promote safety in trucks and vans

Ambiguity

Some trees are more likely than others...

Ambiguity

Some trees are more likely than others...

Can we model that?

Context Free Grammar (CFGs)

A context-free grammar is defined as a tuple $G = \langle N, \Sigma, R, S \rangle$ where:

- N is a set of non-terminal symbols
- $S \in N$ is a distinguished start symbol
- Σ is a set of terminal symbols
- R is a set of rules of the form $X \to Y_1 Y_2 \ldots Y_n$ where $n \geq 0,\ X \in N,\ Y_i \in N \cup \Sigma$

Context Free Grammar (CFGs)

A context-free grammar is defined as a tuple $G = \langle N, \Sigma, R, S \rangle$ where:

- N is a set of non-terminal symbols
- $S \in N$ is a distinguished start symbol
- Σ is a set of terminal symbols
- R is a set of rules of the form $X \to Y_1 Y_2 \ldots Y_n$ where $n \geq 0,\ X \in N,\ Y_i \in N \cup \Sigma$

Probabilistic Context Free Grammar (**P**CFGs)

A context-free grammar is defined as a tuple $G = \langle N, \Sigma, R, S \rangle$ where:

- N is a set of non-terminal symbols
- $S \in N$ is a distinguished start symbol
- Σ is a set of terminal symbols
- R is a set of rules of the form $X \rightarrow Y_1 Y_2 \ldots Y_n$ where $n \geq 0,\ X \in N,\ Y_i \in N \cup \Sigma$

Probabilistic Context Free Grammar (**P**CFGs)

A probabilistic context-free grammar is defined as a tuple $G = \langle N, \Sigma, R, S \quad \rangle$ where:

- N is a set of non-terminal symbols
- $S \in N$ is a distinguished start symbol
- Σ is a set of terminal symbols
- R is a set of rules of the form $X \rightarrow Y_1 Y_2 \ldots Y_n$ where $n \geq 0,\ X \in N,\ Y_i \in N \cup \Sigma$

Probabilistic Context Free Grammar (**P**CFGs)

A probabilistic context-free grammar is defined as a tuple $G = \langle N, \Sigma, R, S, q \rangle$ where:

- N is a set of non-terminal symbols
- $S \in N$ is a distinguished start symbol
- Σ is a set of terminal symbols
- R is a set of rules of the form $X \rightarrow Y_1 Y_2 \ldots Y_n$ where $n \geq 0,\ X \in N,\ Y_i \in N \cup \Sigma$

Probabilistic Context Free Grammar (**P**CFGs)

A probabilistic context-free grammar is defined as a tuple $G = \langle N, \Sigma, R, S, q \rangle$ where:

- N is a set of non-terminal symbols
- $S \in N$ is a distinguished start symbol
- Σ is a set of terminal symbols
- R is a set of rules of the form $X \to Y_1 Y_2 \dots Y_n$ where $n \geq 0,\ X \in N,\ Y_i \in N \cup \Sigma$
- q is a set of non-negative parameters, one for each rule $X \to \alpha \in R$ such that, for any $X \in N$,

$$\sum_{(X \to \alpha) \in R} q(X \to \alpha) = 1$$

Context Free Grammars, Example

$$
\begin{array}{rcl}
N & = & \{\mathrm{S}, \mathrm{VP}, \mathrm{NP}, \mathrm{PP}, \mathrm{DT}, \mathrm{Vi}, \mathrm{Vt}, \mathrm{NN}, \mathrm{IN}\}^{1} \\
S & = & \{\mathrm{S}\} \\
\Sigma & = & \{\text{sleeps}, \text{saw}, \text{man}, \text{woman}, \text{telescope}, \text{the}, \text{with}, \text{in}\} \\
R & = & \left\{ \begin{array}{ll}
\mathrm{S} \rightarrow \mathrm{NP}\ \mathrm{VP} & \mathrm{Vi} \rightarrow \text{sleeps} \\
\mathrm{NP} \rightarrow \mathrm{DT}\ \mathrm{NN} & \mathrm{Vt} \rightarrow \text{saw} \\
\mathrm{NP} \rightarrow \mathrm{NP}\ \mathrm{PP} & \mathrm{NN} \rightarrow \text{man} \\
\mathrm{PP} \rightarrow \mathrm{IN}\ \mathrm{NP} & \mathrm{NN} \rightarrow \text{woman} \\
\mathrm{VP} \rightarrow \mathrm{Vi} & \mathrm{NN} \rightarrow \text{telescope} \\
\mathrm{VP} \rightarrow \mathrm{Vt}\ \mathrm{NP} & \mathrm{DT} \rightarrow \text{the} \\
\mathrm{VP} \rightarrow \mathrm{VP}\ \mathrm{PP} & \mathrm{IN} \rightarrow \text{with} \\
 & \mathrm{IN} \rightarrow \text{in}
\end{array} \right\}
\end{array}
$$

[1] S=sentence, VP=verb phrase, NP=noun phrase, PP=prepositional phrase, DT=determiner, Vi=intransitive verb, Vt=transitive verb, NN=noun, IN=preposition

Probabilistic Context Free Grammars, Example

$$
\begin{aligned}
N &= \{\mathrm{S}, \mathrm{VP}, \mathrm{NP}, \mathrm{PP}, \mathrm{DT}, \mathrm{Vi}, \mathrm{Vt}, \mathrm{NN}, \mathrm{IN}\}^{1} \\
S &= \{\mathrm{S}\} \\
\Sigma &= \{\mathrm{sleeps}, \mathrm{saw}, \mathrm{man}, \mathrm{woman}, \mathrm{telescope}, \mathrm{the}, \mathrm{with}, \mathrm{in}\} \\
R &= \left\{ \begin{array}{llll}
\mathrm{S} \rightarrow \mathrm{NP\ VP} & 1.0 & \mathrm{Vi} \rightarrow \mathrm{sleeps} & 1.0 \\
\mathrm{NP} \rightarrow \mathrm{DT\ NN} & 0.4 & \mathrm{Vt} \rightarrow \mathrm{saw} & 1.0 \\
\mathrm{NP} \rightarrow \mathrm{NP\ PP} & 0.6 & \mathrm{NN} \rightarrow \mathrm{man} & 0.7 \\
\mathrm{PP} \rightarrow \mathrm{IN\ NP} & 1.0 & \mathrm{NN} \rightarrow \mathrm{woman} & 0.2 \\
\mathrm{VP} \rightarrow \mathrm{Vi} & 0.5 & \mathrm{NN} \rightarrow \mathrm{telescope} & 0.1 \\
\mathrm{VP} \rightarrow \mathrm{Vt\ NP} & 0.4 & \mathrm{DT} \rightarrow \mathrm{the} & 1.0 \\
\mathrm{VP} \rightarrow \mathrm{VP\ PP} & 0.1 & \mathrm{IN} \rightarrow \mathrm{with} & 0.5 \\
 & & \mathrm{IN} \rightarrow \mathrm{in} & 0.5
\end{array} \right\}
\end{aligned}
$$

[1] S=sentence, VP=verb phrase, NP=noun phrase, PP=prepositional phrase, DT=determiner, Vi=intransitive verb, Vt=transitive verb, NN=noun, IN=preposition

Properties of PCFGs

- The probability of a parse tree $t \in \mathcal{T}_G$ is computed as:

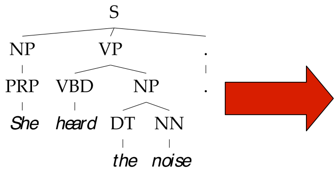
$$p(t) = \prod_{r \in t} q(r)$$

- If there is more than one tree for a sentence, we can rank them by probability.
- The most likely tree for a sentence s is:

$$\arg\max_{t \in \mathcal{T}(s)} p(t)$$

Learning Treebank Grammars

- Read the grammar rules from a treebank



S → NP VP .	1
NP → PRP	0.5
NP → DT NN	0.5
VP → VBD NP	1
PRP → She	1
...	

- Set rule weights by maximum likelihood

$$q(\alpha \to \beta) = \frac{\text{Count}(\alpha \to \beta)}{\text{Count}(\alpha)}$$

- Smoothing issues apply
- Having the appropriate CFG is critical to success

Outline

1. Trees and Grammars
2. Constituency Parsing
   - CKY Algorithm
   - Earley Algorithm
3. Dependency Parsing
   - Dependency Trees
   - Arc-factored Dependency Parsing
   - Parsing Projective Structures
   - Parsing non-Projective Structures
   - Transition-Based parsers

Parsing Natural Language Sentences

Goal of a parser:

- Find all possible trees

Parsing Natural Language Sentences

Goal of a parser:

- Find all possible trees
- Find all possible trees, ranked by probability

Parsing Natural Language Sentences

Goal of a parser:

- Find all possible trees
- Find all possible trees, ranked by probability
- Find most likely tree

Parsing Natural Language Sentences

Goal of a parser:

- Find all possible trees
- Find all possible trees, ranked by probability
- Find most likely tree

- Many of the possible trees will share subtrees that we don't need to re-parse.

Parsing Natural Language Sentences

Goal of a parser:

- Find all possible trees
- Find all possible trees, ranked by probability
- Find most likely tree

- Many of the possible trees will share subtrees that we don't need to re-parse.
- Define a dynammic programming table (*aka* **chart**) to store intermediate results.

Outline

1 Trees and Grammars

2 Constituency Parsing
- CKY Algorithm
- Earley Algorithm

3 Dependency Parsing
- Dependency Trees
- Arc-factored Dependency Parsing
- Parsing Projective Structures
- Parsing non-Projective Structures
- Transition-Based parsers

CKY Algorithm

- Bottom-up
- Requires a grammar in Chomsky Normal Form (CNF).
- Dynammic programming: Store partial results that can be reused in different candidate solutions.
- Analogous to Viterbi in HMMs.
- Intermediate results stored in a *chart* structure.

CKY Algorithm

Chart content:

- Maximum probability of a subtree with root X spanning words $i \dots j$:
$$\pi(i, j, X)$$
- Backpath to recover which rules produced the maximum probability tree:
$$\psi(i, j, X)$$

The goal is to compute:

- $\max_{t \in \mathcal{T}(s)} p(t) = \pi(1, n, S)$
- $\psi(1, n, S)$
- It is possible to use it without probabilities to get all parse trees (with higher complexity)

CKY Algorithm

Base case: Tree leaves

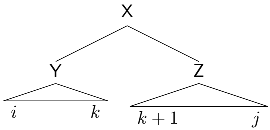
- $\forall i = 1 \ldots n$, $\forall X \rightarrow w_i \in R, \quad \pi(i, i, X) = q(X \rightarrow w_i)$

Recursive case: Non-terminal nodes

- $\forall i = 1 \ldots n$, $\forall j = (i+1) \ldots n$, $\forall X \in N$

$$\pi(i, j, X) = \max_{\substack{X \rightarrow Y Z \in R \\ k : i \leq k < j}} q(X \rightarrow Y Z) \times \pi(i, k, Y) \times \pi(k+1, j, Z)$$

$$\psi(i, j, X) = \arg \max_{\substack{X \rightarrow Y Z \in R \\ k : i \leq k < j}} q(X \rightarrow Y Z) \times \pi(i, k, Y) \times \pi(k+1, j, Z)$$



Output:

- Return $\pi(1, n, S)$ and recover backpath trough $\psi(1, n, S)$

CKY Algorithm - Example

$$N = \{\mathrm{S}, \mathrm{VP}, \mathrm{NP}, \mathrm{PP}, \mathrm{DT}, \mathrm{Vi}, \mathrm{Vt}, \mathrm{NN}, \mathrm{IN}\}^1$$

$$S = \{\mathrm{S}\}$$

$$\Sigma = \{\mathrm{sleeps}, \mathrm{saw}, \mathrm{man}, \mathrm{woman}, \mathrm{telescope}, \mathrm{the}, \mathrm{with}, \mathrm{in}\}$$

$$R = \left\{ \begin{array}{llll} \mathrm{S} \rightarrow \mathrm{NP\ VP} & 1.0 & \mathrm{Vi} \rightarrow \mathrm{sleeps} & 1.0 \\ \mathrm{NP} \rightarrow \mathrm{DT\ NN} & 0.4 & \mathrm{Vt} \rightarrow \mathrm{saw} & 1.0 \\ \mathrm{NP} \rightarrow \mathrm{NP\ PP} & 0.6 & \mathrm{NN} \rightarrow \mathrm{man} & 0.7 \\ \mathrm{PP} \rightarrow \mathrm{IN\ NP} & 1.0 & \mathrm{NN} \rightarrow \mathrm{woman} & 0.2 \\ \mathrm{VP} \rightarrow \mathrm{Vi} & 0.5 & \mathrm{NN} \rightarrow \mathrm{telescope} & 0.1 \\ \mathrm{VP} \rightarrow \mathrm{Vt\ NP} & 0.4 & \mathrm{DT} \rightarrow \mathrm{the} & 1.0 \\ \mathrm{VP} \rightarrow \mathrm{VP\ PP} & 0.1 & \mathrm{IN} \rightarrow \mathrm{with} & 0.5 \\ & & \mathrm{IN} \rightarrow \mathrm{in} & 0.5 \end{array} \right\}$$

[1] S=sentence, VP=verb phrase, NP=noun phrase, PP=prepositional phrase, DT=determiner, Vi=intransitive verb, Vt=transitive verb, NN=noun, IN=preposition

CKY Algorithm - Example - CNF

$$
\begin{aligned}
N &= \{\mathrm{S}, \mathrm{VP}, \mathrm{NP}, \mathrm{PP}, \mathrm{DT}, \mathrm{Vi}, \mathrm{Vt}, \mathrm{NN}, \mathrm{IN}\}^{1} \\
S &= \{\mathrm{S}\} \\
\Sigma &= \{\text{sleeps}, \text{saw}, \text{man}, \text{woman}, \text{telescope}, \text{the}, \text{with}, \text{in}\} \\
R &= \left\{ \begin{array}{llll}
\mathrm{S} \rightarrow \mathrm{NP}\ \mathrm{VP} & 1.0 & \mathrm{Vi} \rightarrow \text{sleeps} & 1.0 \\
\mathrm{NP} \rightarrow \mathrm{DT}\ \mathrm{NN} & 0.4 & \mathrm{Vt} \rightarrow \text{saw} & 1.0 \\
\mathrm{NP} \rightarrow \mathrm{NP}\ \mathrm{PP} & 0.6 & \mathrm{NN} \rightarrow \text{man} & 0.7 \\
\mathrm{PP} \rightarrow \mathrm{IN}\ \mathrm{NP} & 1.0 & \mathrm{NN} \rightarrow \text{woman} & 0.2 \\
\mathrm{VP} \rightarrow \mathrm{Vi} & 0.5 & \mathrm{NN} \rightarrow \text{telescope} & 0.1 \\
\mathrm{VP} \rightarrow \mathrm{Vt}\ \mathrm{NP} & 0.4 & \mathrm{DT} \rightarrow \text{the} & 1.0 \\
\mathrm{VP} \rightarrow \mathrm{VP}\ \mathrm{PP} & 0.1 & \mathrm{IN} \rightarrow \text{with} & 0.5 \\
 & & \mathrm{IN} \rightarrow \text{in} & 0.5
\end{array} \right\}
\end{aligned}
$$

[1] S=sentence, VP=verb phrase, NP=noun phrase, PP=prepositional phrase, DT=determiner, Vi=intransitive verb, Vt=transitive verb, NN=noun, IN=preposition

CKY Algorithm - Example - CNF

$$N = \{\text{S}, \text{VP}, \text{NP}, \text{PP}, \text{DT}, \text{Vi}, \text{Vt}, \text{NN}, \text{IN}\}^1$$

$$S = \{\text{S}\}$$

$$\Sigma = \{\text{sleeps}, \text{saw}, \text{man}, \text{woman}, \text{telescope}, \text{the}, \text{with}, \text{in}\}$$

$$R = \left\{ \begin{array}{llll} \text{S} \rightarrow \text{NP VP} & 0.5 & \text{Vi} \rightarrow \text{sleeps} & 1.0 \\ \text{S} \rightarrow \text{NP Vi} & 0.5 & \text{Vt} \rightarrow \text{saw} & 1.0 \\ \text{NP} \rightarrow \text{DT NN} & 0.4 & \text{NN} \rightarrow \text{man} & 0.7 \\ \text{NP} \rightarrow \text{NP PP} & 0.6 & \text{NN} \rightarrow \text{woman} & 0.2 \\ \text{PP} \rightarrow \text{IN NP} & 1.0 & \text{NN} \rightarrow \text{telescope} & 0.1 \\ \text{VP} \rightarrow \text{Vt NP} & 0.4 & \text{DT} \rightarrow \text{the} & 1.0 \\ \text{VP} \rightarrow \text{VP PP} & 0.1 & \text{IN} \rightarrow \text{with} & 0.5 \\ \text{VP} \rightarrow \text{Vi PP} & 0.5 & \text{IN} \rightarrow \text{in} & 0.5 \end{array} \right\}$$

[1] S=sentence, VP=verb phrase, NP=noun phrase, PP=prepositional phrase, DT=determiner, Vi=intransitive verb, Vt=transitive verb, NN=noun, IN=preposition

CKY Algorithm - Example

DT 1.0 The 11	NN 0.2 woman 22	Vt 1.0 saw 33	DT 1.0 the 44	NN 0.7 man 55	IN 0.5 with 66	DT 1.0 the 77	NN 0.1 telescope 88

CKY Algorithm - Example

NP → $DT_{11}NN_{22}$ 0.4*1.0*0.2=0.08 12	23	34	NP → $DT_{44}NN_{55}$ 0.4*1.0*0.7=0.28 45	56	67	NP → $DT_{44}NN_{55}$ 0.4*1.0*0.1=0.04 78

DT 1.0 The 11	NN 0.2 woman 22	Vt 1.0 saw 33	DT 1.0 the 44	NN 0.7 man 55	IN 0.5 with 66	DT 1.0 the 77	NN 0.1 telescope 88

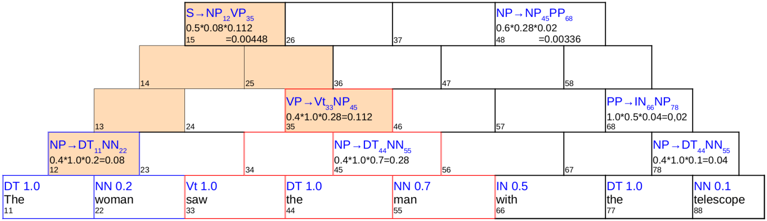
CKY Algorithm - Example

13	24	$VP \rightarrow Vt_{33} NP_{45}$ 0.4*1.0*0.28=0.112 35	46	57	$PP \rightarrow IN_{66} NP_{78}$ 1.0*0.5*0.04=0,02 68		
$NP \rightarrow DT_{11} NN_{22}$ 0.4*1.0*0.2=0.08 12	23	34	$NP \rightarrow DT_{44} NN_{55}$ 0.4*1.0*0.7=0.28 45	56	67	$NP \rightarrow DT_{44} NN_{55}$ 0.4*1.0*0.1=0.04 78	
DT 1.0 The 11	NN 0.2 woman 22	Vt 1.0 saw 33	DT 1.0 the 44	NN 0.7 man 55	IN 0.5 with 66	DT 1.0 the 77	NN 0.1 telescope 88

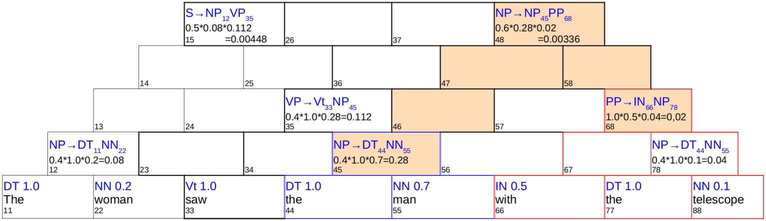
CKY Algorithm - Example

14	25	36	47	58			
13	24	$VP \rightarrow Vt_{33} NP_{45}$ 0.4*1.0*0.28=0.112 35	46	57	$PP \rightarrow IN_{66} NP_{78}$ 1.0*0.5*0.04=0,02 68		
$NP \rightarrow DT_{11} NN_{22}$ 0.4*1.0*0.2=0.08 12	23	34	$NP \rightarrow DT_{44} NN_{55}$ 0.4*1.0*0.7=0.28 45	56	67	$NP \rightarrow DT_{44} NN_{55}$ 0.4*1.0*0.1=0.04 78	
DT 1.0 The 11	NN 0.2 woman 22	Vt 1.0 saw 33	DT 1.0 the 44	NN 0.7 man 55	IN 0.5 with 66	DT 1.0 the 77	NN 0.1 telescope 88

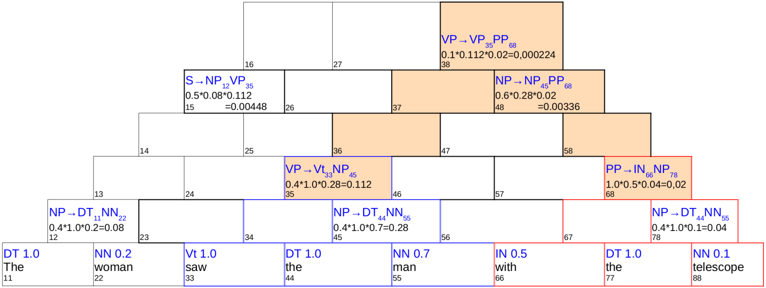
CKY Algorithm - Example



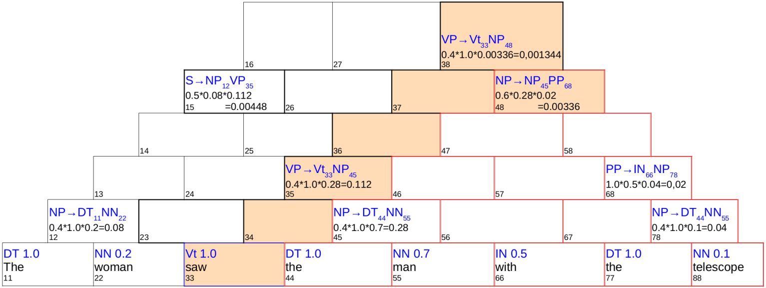
CKY Algorithm - Example



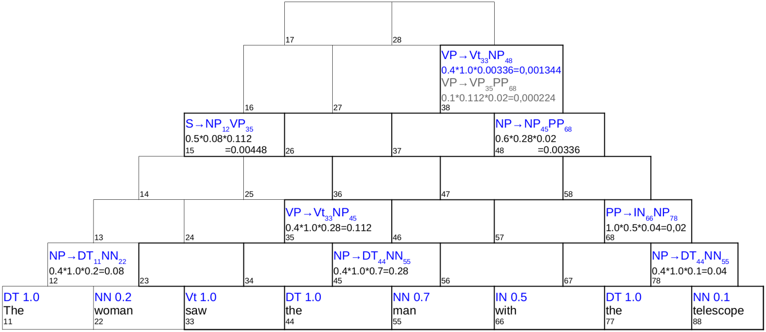
CKY Algorithm - Example



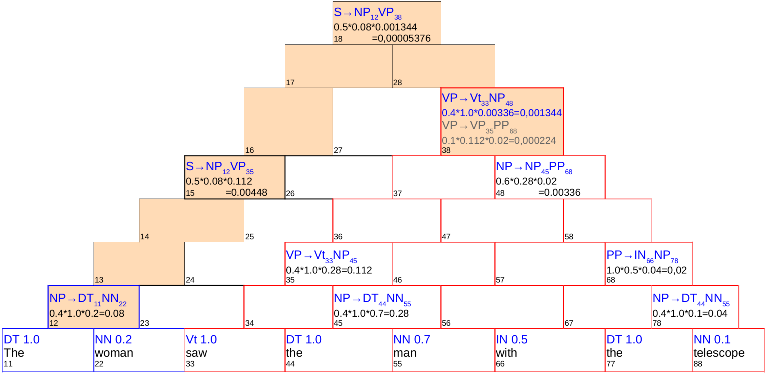
CKY Algorithm - Example



CKY Algorithm - Example



CKY Algorithm - Example



Outline

1 Trees and Grammars

2 Constituency Parsing
- CKY Algorithm
- Earley Algorithm

3 Dependency Parsing
- Dependency Trees
- Arc-factored Dependency Parsing
- Parsing Projective Structures
- Parsing non-Projective Structures
- Transition-Based parsers

Earley Algorithm

- Top-down
- Can deal with any CFG (even left-recursive)
- Dynammic programming: Store partial results that can be reused in different candidate solutions.
- Intermediate results stored in a *chart* structure.

Earley Algorithm

Chart content:

- Set of items (aka *states*), each describing the applicability status of each rule after each word:

$$[i, j, X \to \alpha \bullet \beta]$$

- Backpath to recover which rules produced the complete tree:

$$\psi(i, j, X)$$

The goal is:

- Find if it is possible to reach $[1, n, S \to \alpha \cdot]$
- Recover $\psi(0, n, S)$ if it is
- Probabilistic versions exist, though not as straightforward as in CKY

Earley Algorithm

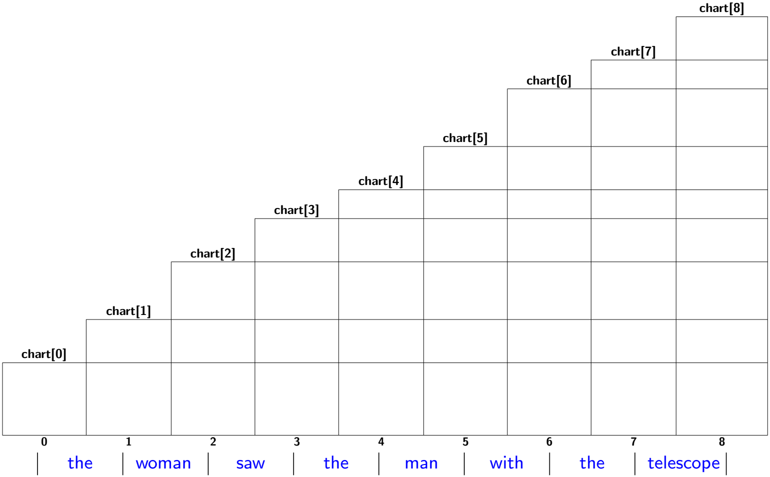
Parsing state examples:

$[0, 0, \mathrm{S} \rightarrow \bullet \ \mathrm{NP} \ \mathrm{VP}]$	A NP is expected at the beginning of the sentence
$[1, 2, \mathrm{NP} \rightarrow \mathrm{DT} \bullet \mathrm{NN}]$	A NP has been partially matched (DT was found between positions 1 and 2)
$[0, 3, \mathrm{VP} \rightarrow \mathrm{V} \ \mathrm{NP} \ \bullet]$	A VP has been completed between positions 0 and 3

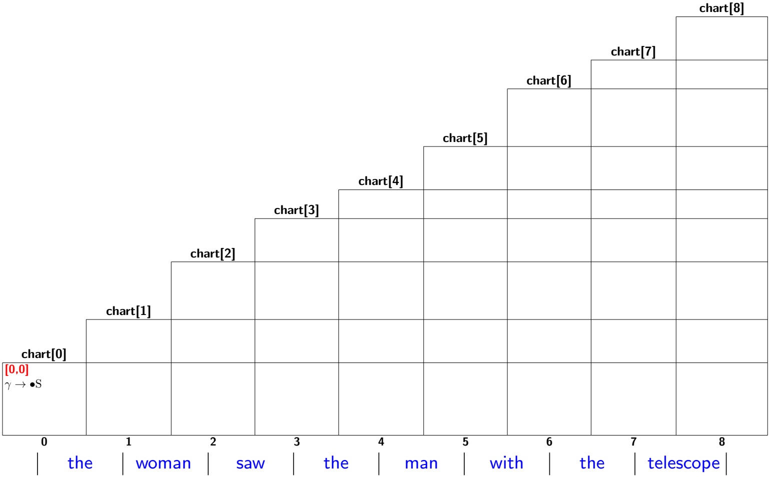
Earley Algorithm

```
def Earley(words,grammar):
    chart = [ [ ] for i in range(len(words)+1) ]
    chart[0].append([0,0,γ → •S])
    for i in range(len(words)+1) :
        for state in chart[i] :
            if state.complete() : Complete(state)
            elif is_PoS(state.next()) : Scan(state)
            else : Predict(state)
    return chart

def Scan([i,j,A → α • Bβ]):
    if B in words[j].PoS() : chart[j+1].append([j,j+1,B →word[j]•])

def Predict([i,j,A → α • Bβ]):
    for B → γ in grammar : chart[j].append([j,j,B → •γ])

def Complete([k,j,B → γ•]):
    for [i,k,A → α • Bβ] in chart[k] : chart[j].append([i,j,A → αB • β])
```

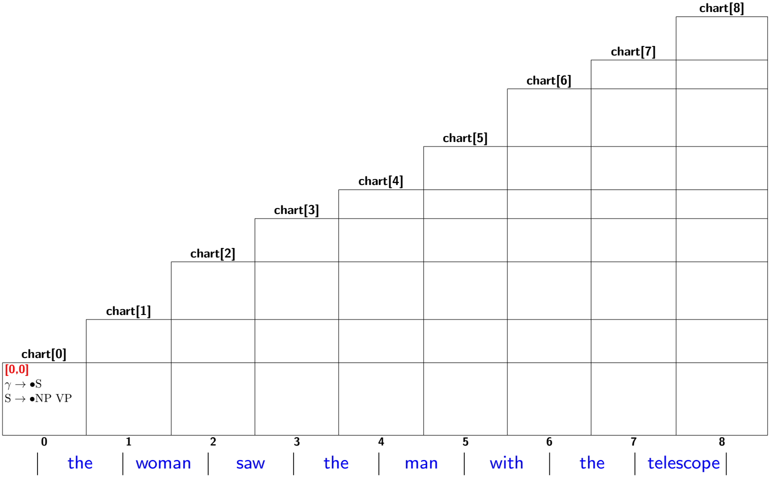
Earley Algorithm



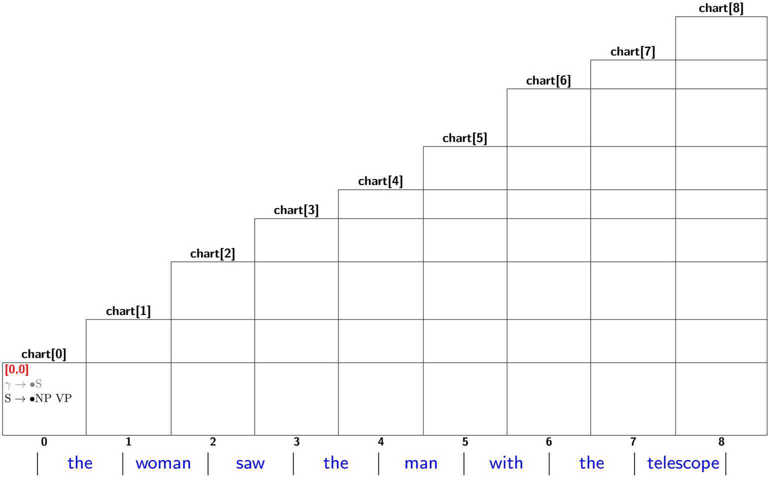
Earley Algorithm



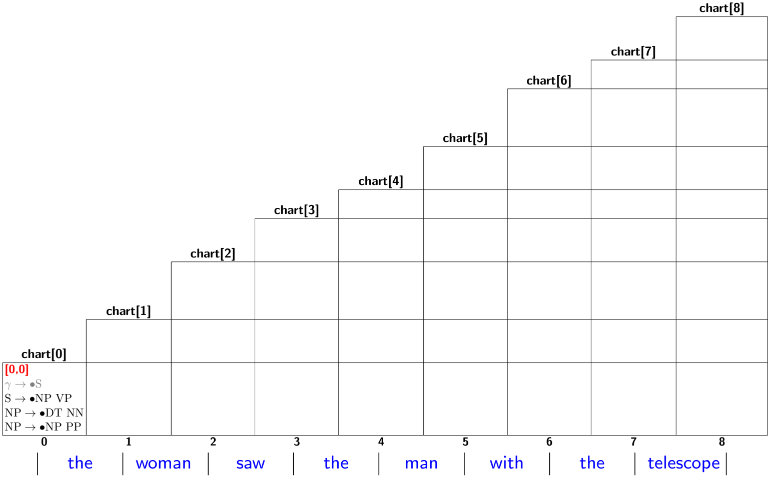
Earley Algorithm



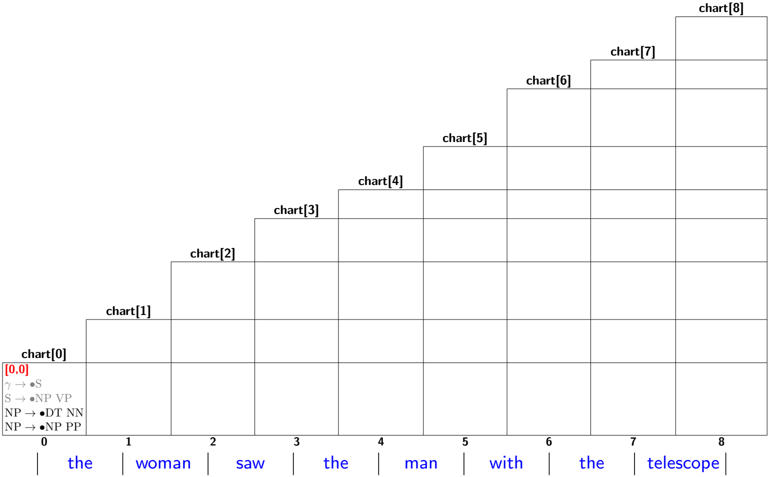
Earley Algorithm



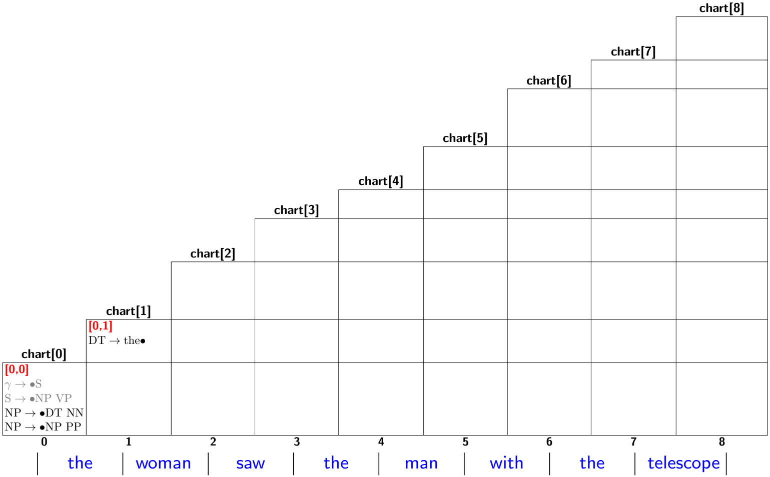
Earley Algorithm



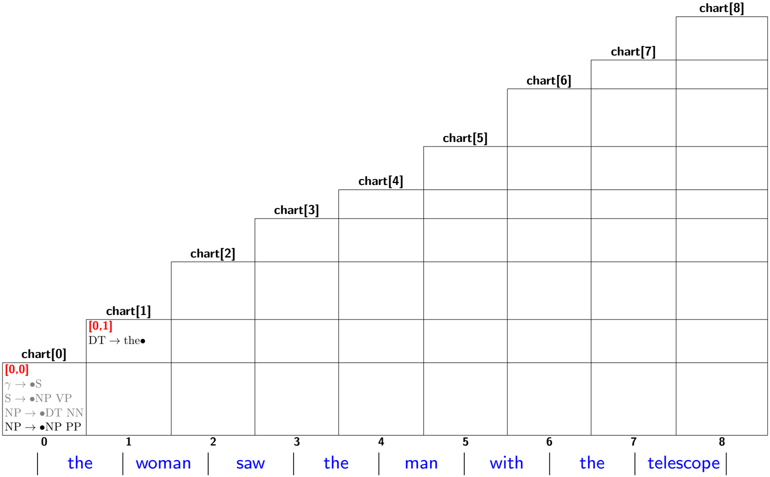
Earley Algorithm



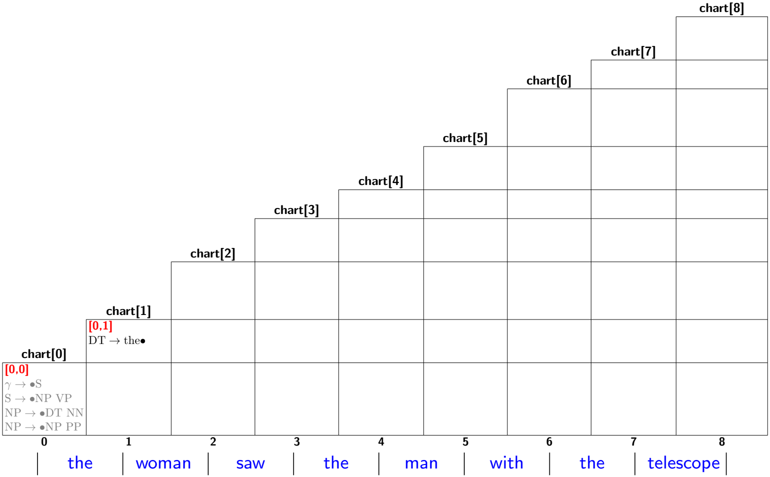
Earley Algorithm



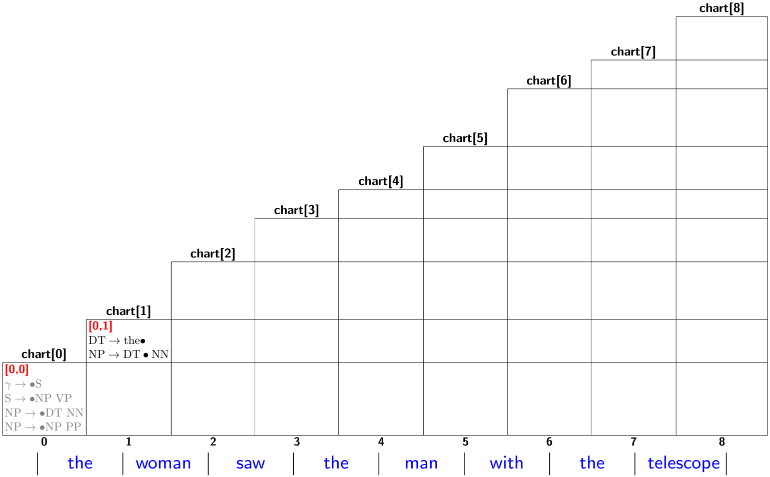
Earley Algorithm



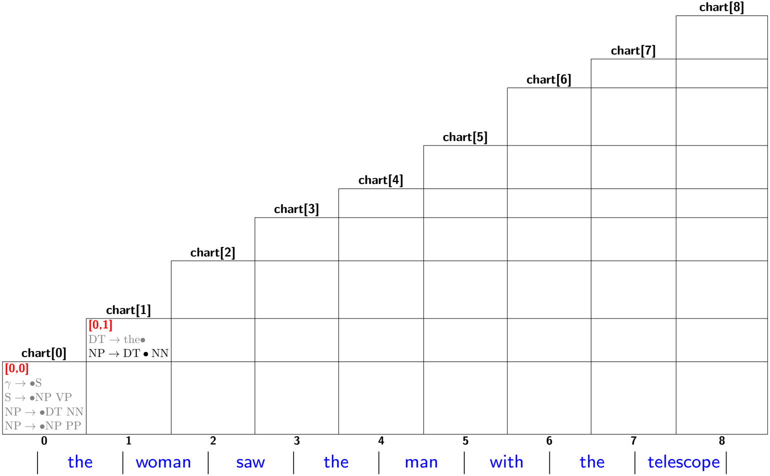
Earley Algorithm



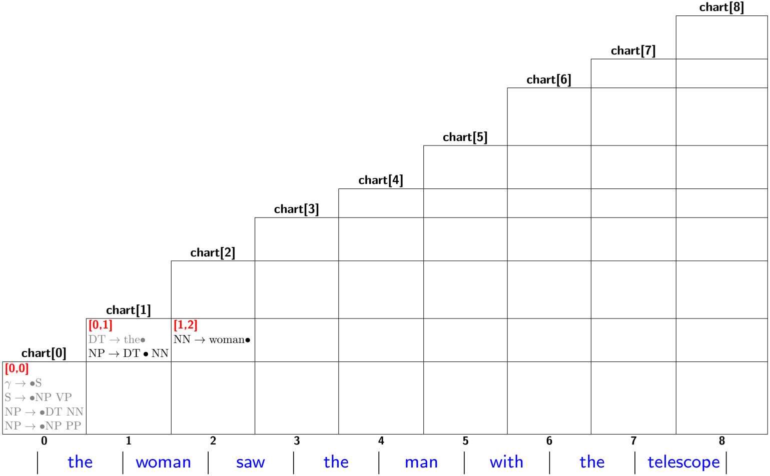
Earley Algorithm



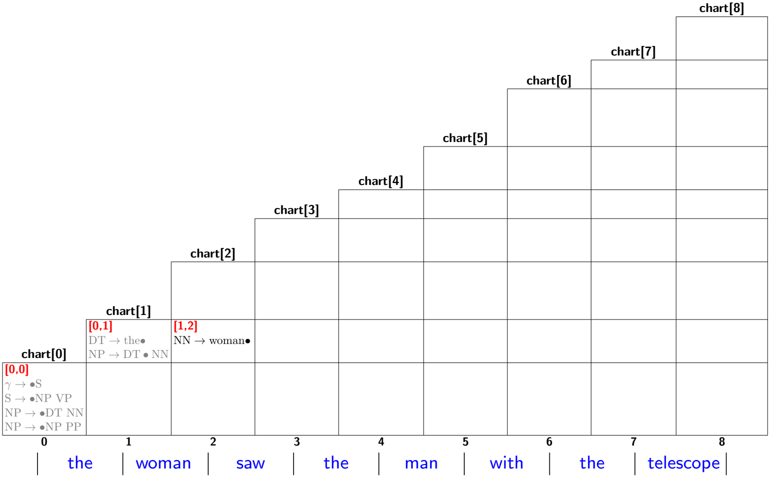
Earley Algorithm



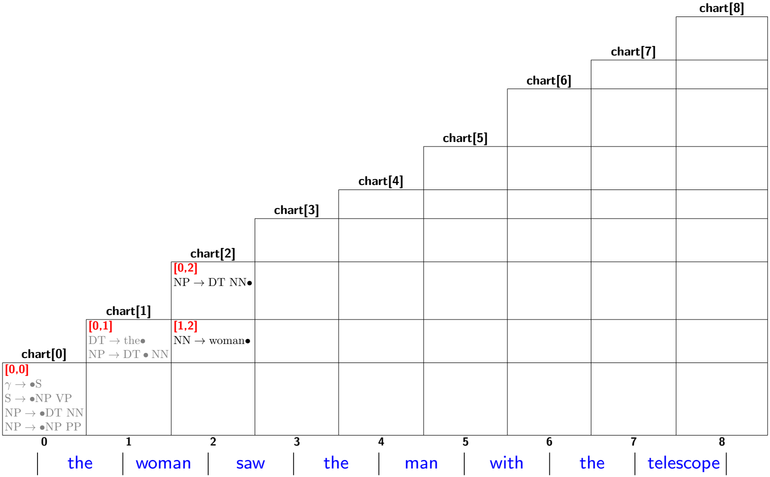
Earley Algorithm



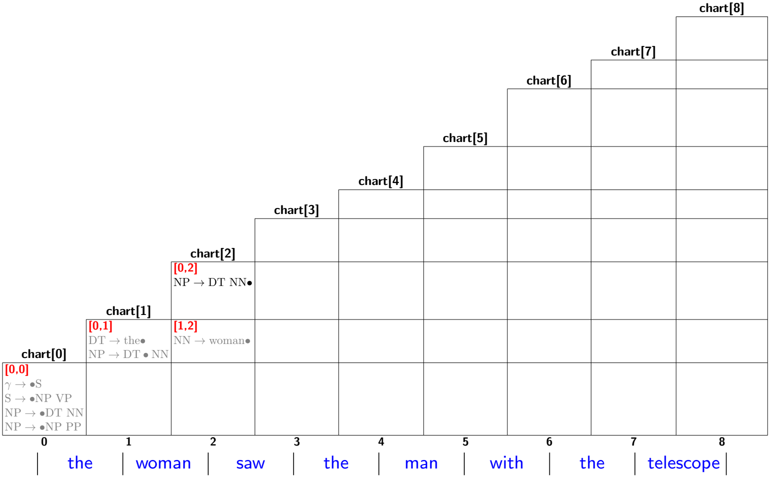
Earley Algorithm



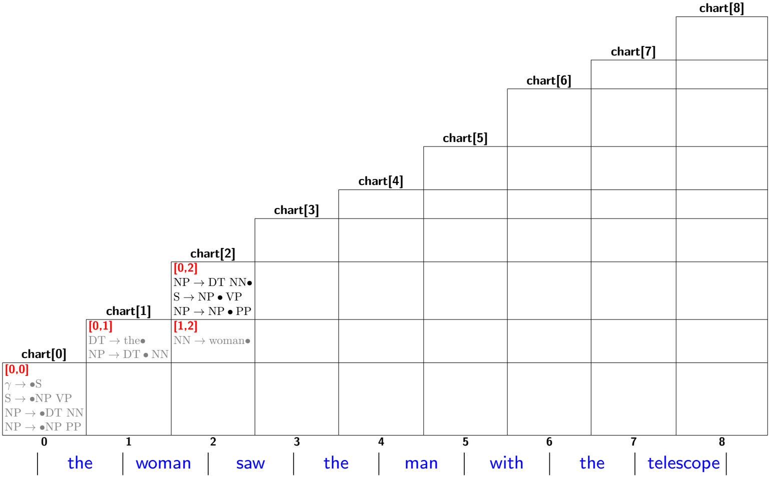
Earley Algorithm



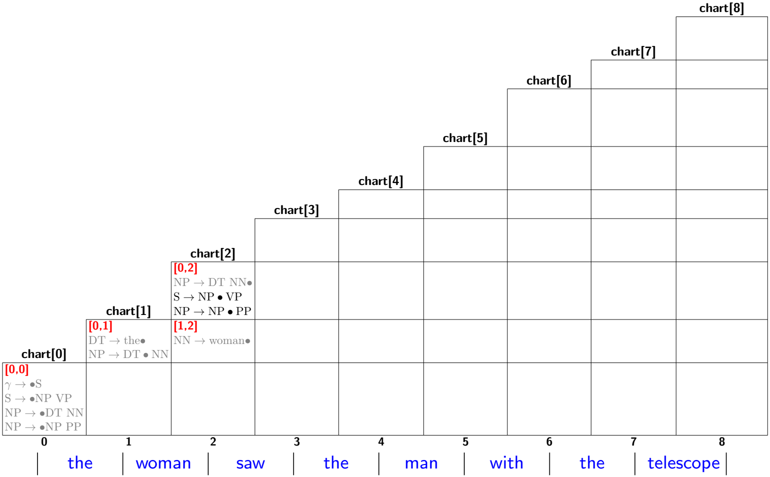
Earley Algorithm



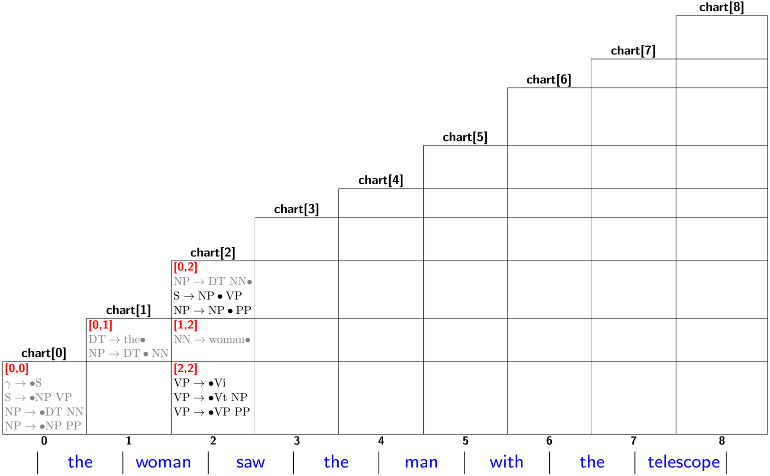
Earley Algorithm



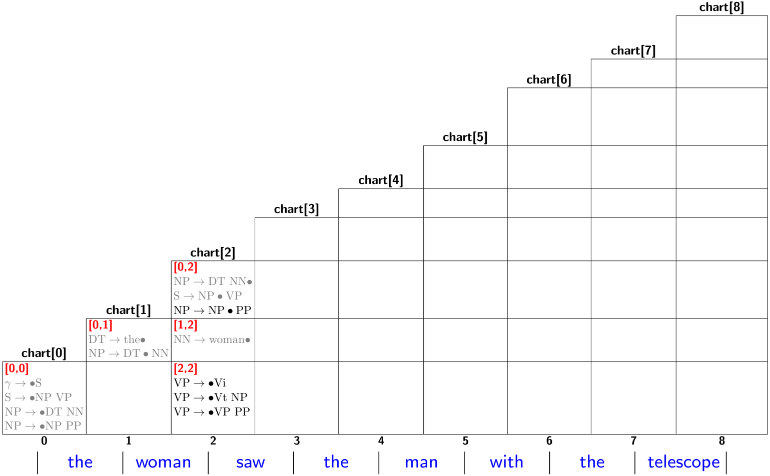
Earley Algorithm



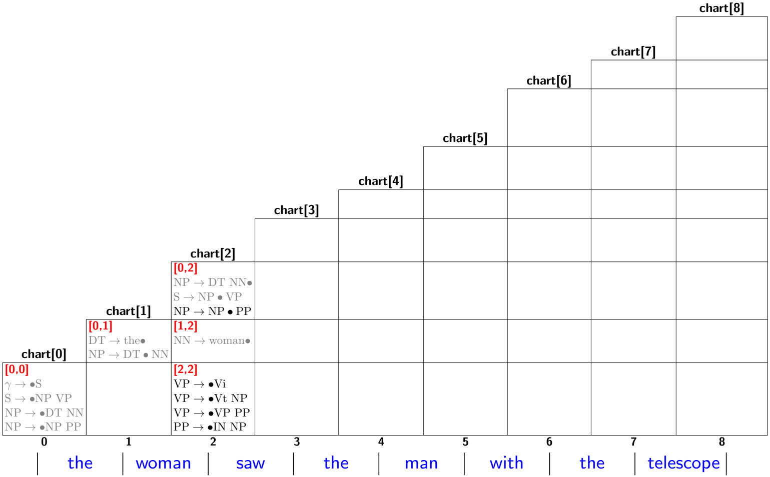
Earley Algorithm



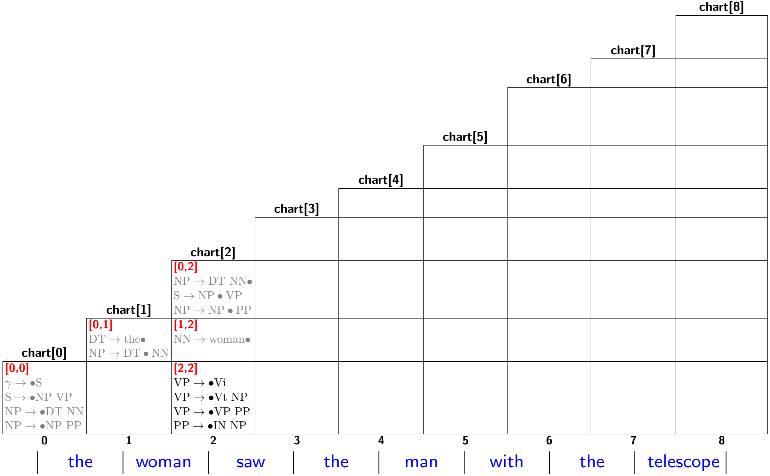
Earley Algorithm



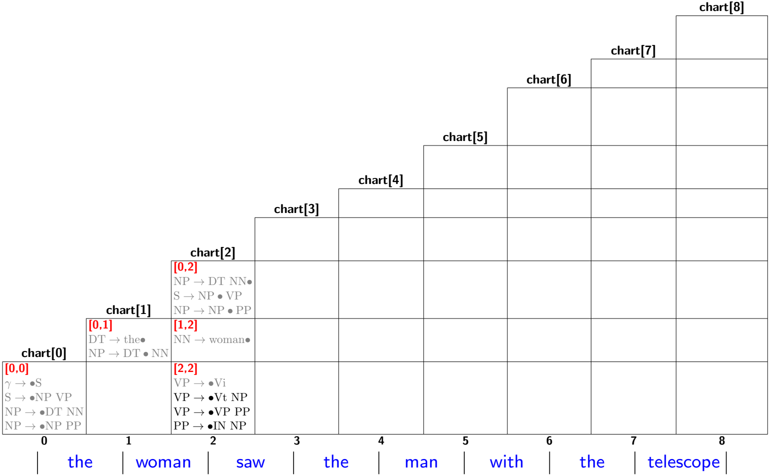
Earley Algorithm



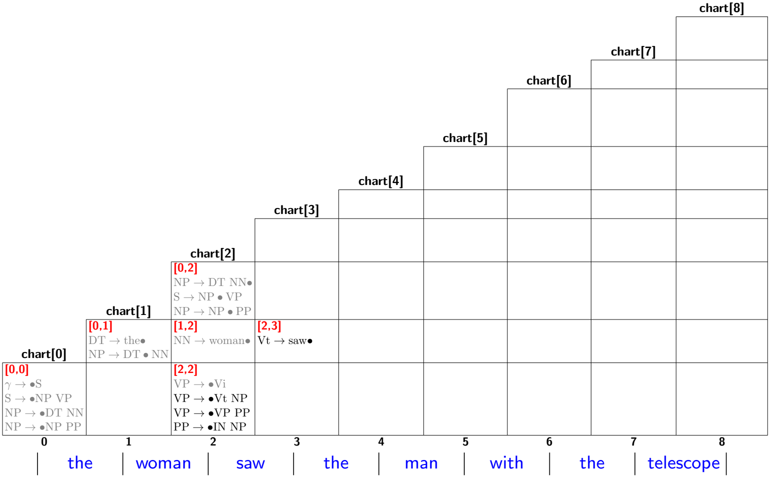
Earley Algorithm



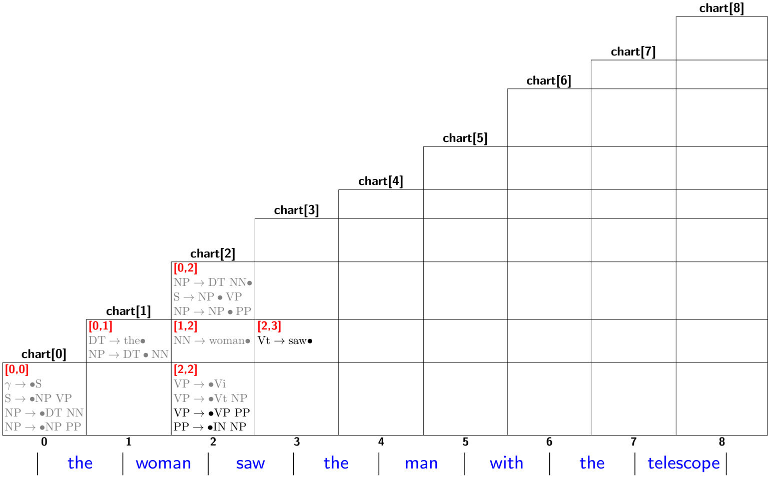
Earley Algorithm



Earley Algorithm



Earley Algorithm

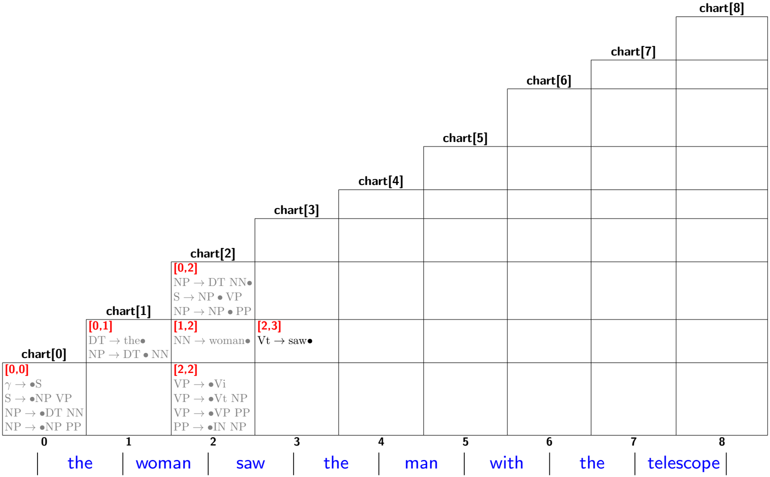


Earley Algorithm

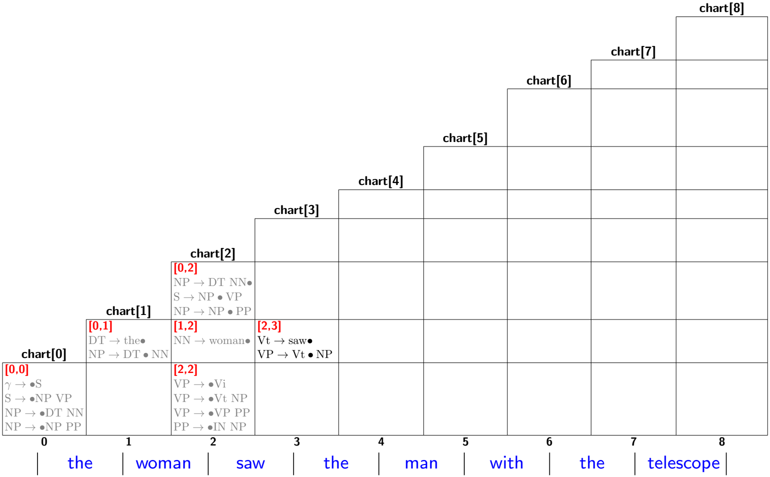
	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
			[0,2] NP $\rightarrow$ DT NN$\bullet$ S $\rightarrow$ NP $\bullet$ VP NP $\rightarrow$ NP $\bullet$ PP						
		[0,1] DT $\rightarrow$ the$\bullet$ NP $\rightarrow$ DT $\bullet$ NN	[1,2] NN $\rightarrow$ woman$\bullet$	[2,3] Vt $\rightarrow$ saw$\bullet$					
	[0,0] $\gamma \rightarrow \bullet$S S $\rightarrow$ $\bullet$NP VP NP $\rightarrow$ $\bullet$DT NN NP $\rightarrow$ $\bullet$NP PP		[2,2] VP $\rightarrow$ $\bullet$Vi VP $\rightarrow$ $\bullet$Vt NP VP $\rightarrow$ $\bullet$VP PP PP $\rightarrow$ $\bullet$IN NP						
	0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

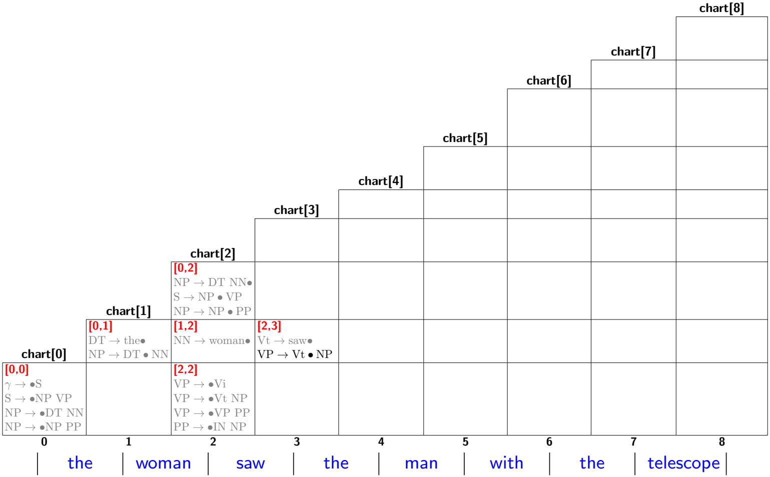
Earley Algorithm



Earley Algorithm



Earley Algorithm

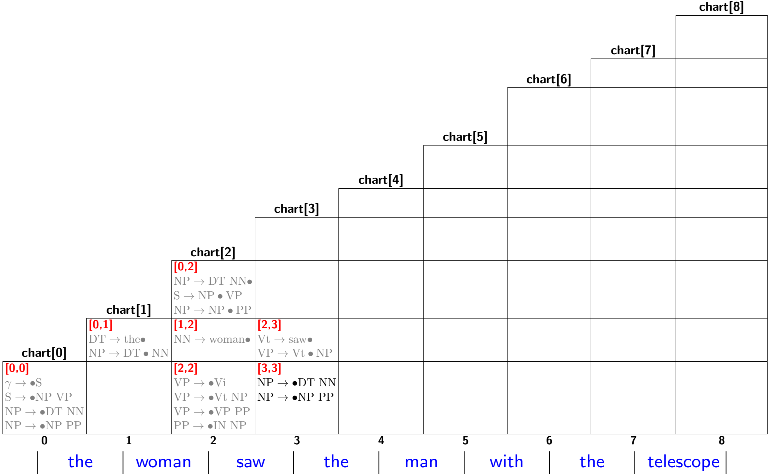


Earley Algorithm

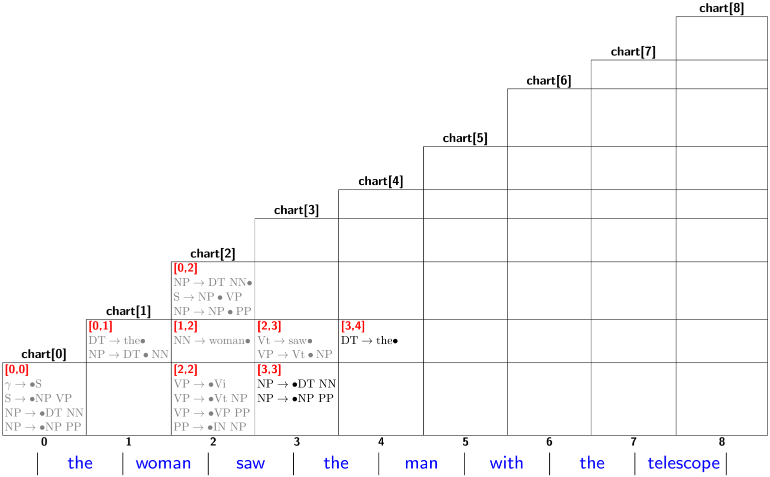
	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
			[0,2] NP $\to$ DT NN$\bullet$ S $\to$ NP $\bullet$ VP NP $\to$ NP $\bullet$ PP						
		[0,1] DT $\to$ the$\bullet$ NP $\to$ DT $\bullet$ NN	**[1,2]** NN $\to$ woman$\bullet$	**[2,3]** Vt $\to$ saw$\bullet$ VP $\to$ Vt $\bullet$ NP					
	[0,0] $\gamma \to \bullet$S S $\to$ $\bullet$NP VP NP $\to$ $\bullet$DT NN NP $\to$ $\bullet$NP PP		**[2,2]** VP $\to$ $\bullet$Vi VP $\to$ $\bullet$Vt NP VP $\to$ $\bullet$VP PP PP $\to$ $\bullet$IN NP	**[3,3]** NP $\to$ $\bullet$DT NN NP $\to$ $\bullet$NP PP					

0 the 1 woman 2 saw 3 the 4 man 5 with 6 the 7 telescope 8

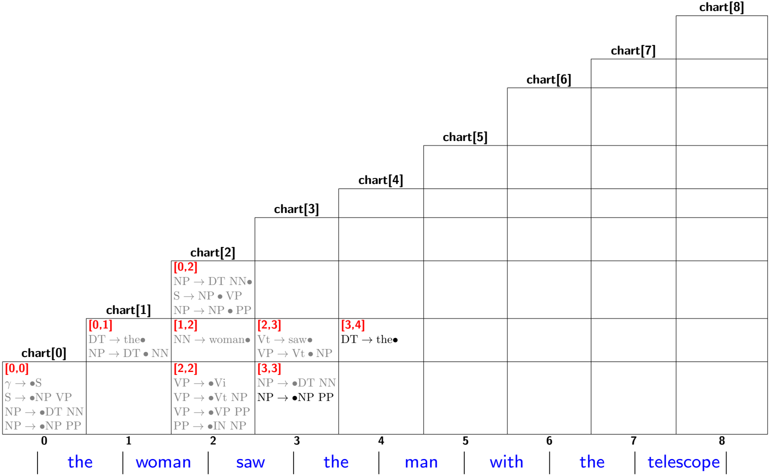
Earley Algorithm



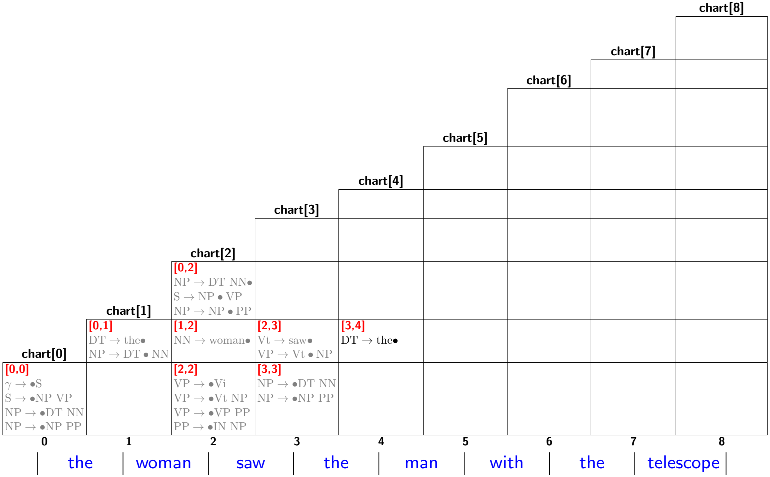
Earley Algorithm



Earley Algorithm



Earley Algorithm

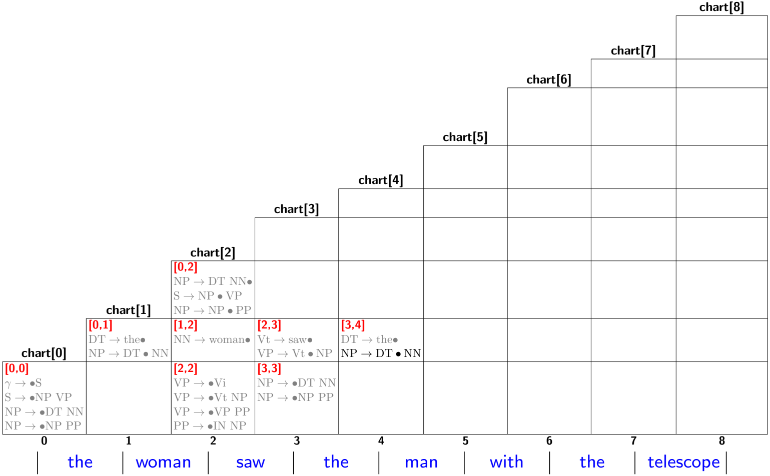


Earley Algorithm

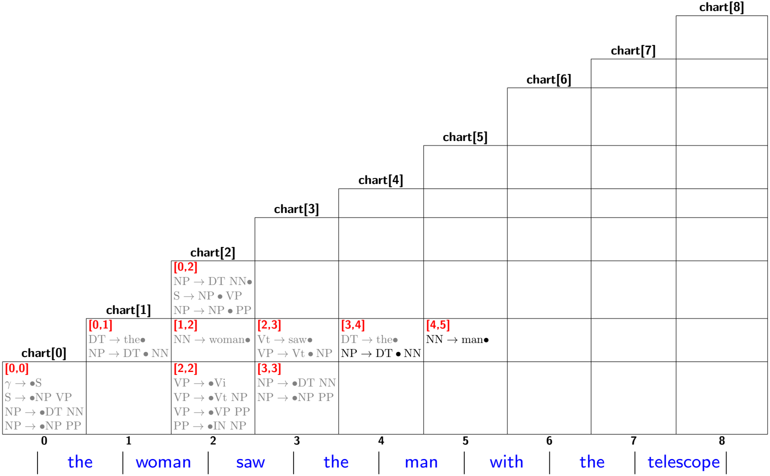
chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
		[0,2] NP → DT NN• S → NP • VP NP → NP • PP						
	[0,1] DT → the• NP → DT • NN	**[1,2]** NN → woman•	**[2,3]** Vt → saw• VP → Vt • NP	**[3,4]** DT → the• NP → DT • NN				
[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		**[2,2]** VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	**[3,3]** NP → •DT NN NP → •NP PP					
0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

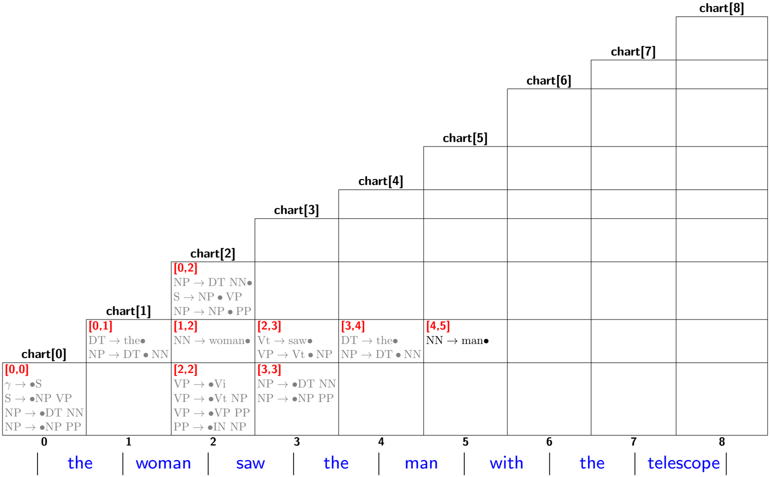
Earley Algorithm



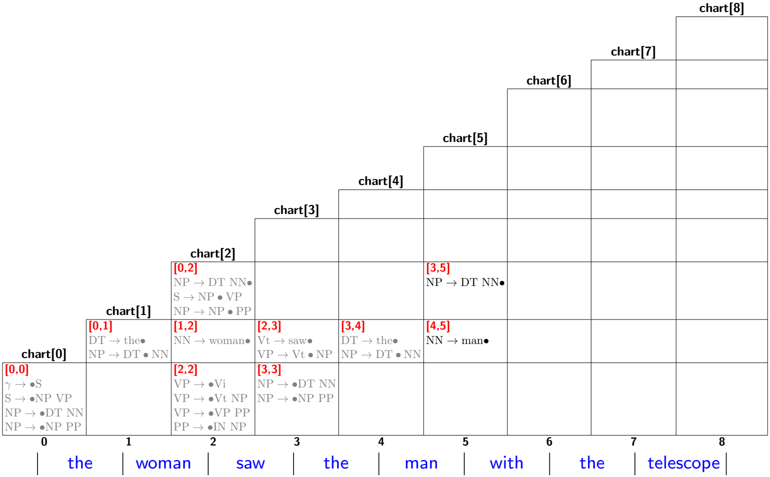
Earley Algorithm



Earley Algorithm



Earley Algorithm

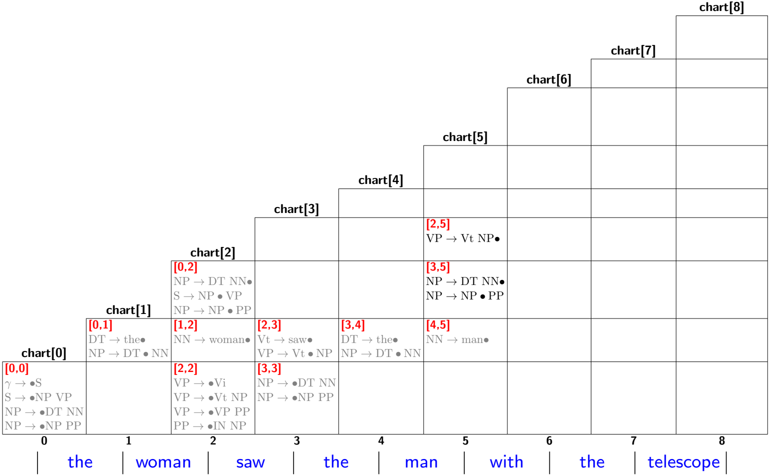


Earley Algorithm

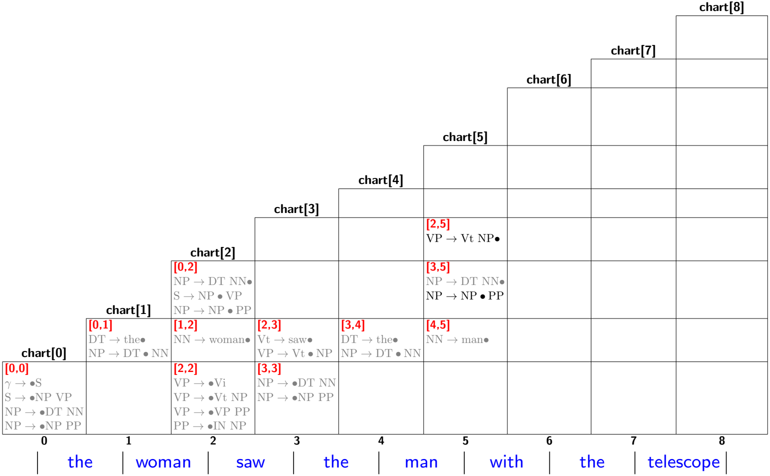
chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
		[0,2] NP → DT NN• S → NP • VP NP → NP • PP			**[3,5]** NP → DT NN•			
	[0,1] DT → the• NP → DT • NN	**[1,2]** NN → woman•	**[2,3]** Vt → saw• VP → Vt • NP	**[3,4]** DT → the• NP → DT • NN	**[4,5]** NN → man•			
[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		**[2,2]** VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	**[3,3]** NP → •DT NN NP → •NP PP					
0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm



Earley Algorithm

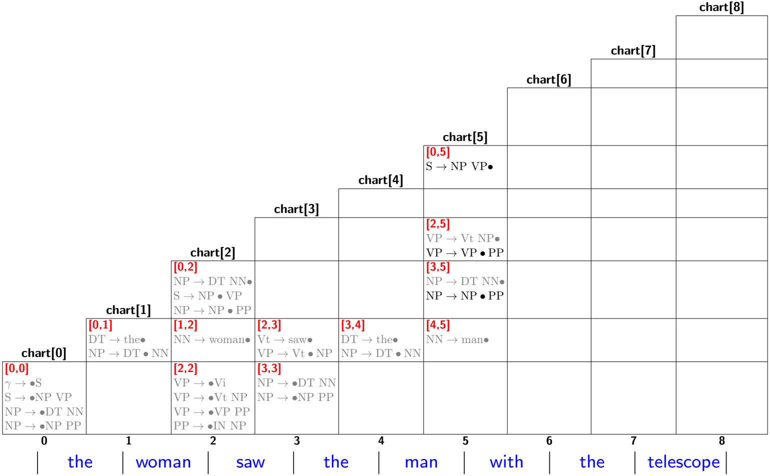


Earley Algorithm

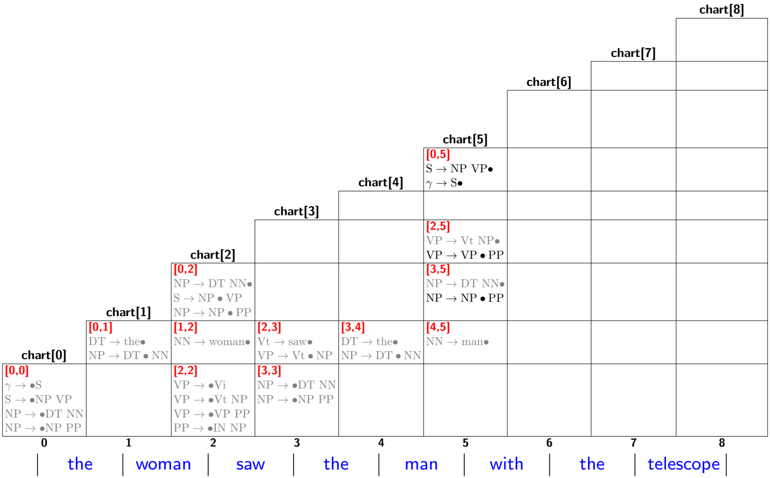
chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
					[0,5] S → NP VP•			
					[2,5] VP → Vt NP• VP → VP • PP			
		[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[3,5] NP → DT NN• NP → NP • PP			
	[0,1] DT → the• NP → DT • NN	[1,2] NN → woman•	[2,3] Vt → saw• VP → Vt • NP	[3,4] DT → the• NP → DT • NN	[4,5] NN → man•			
[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[3,3] NP → •DT NN NP → •NP PP					
0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

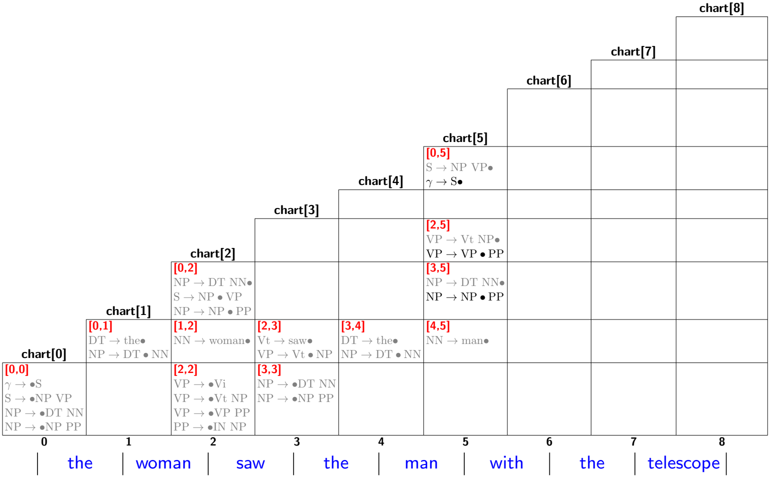
Earley Algorithm



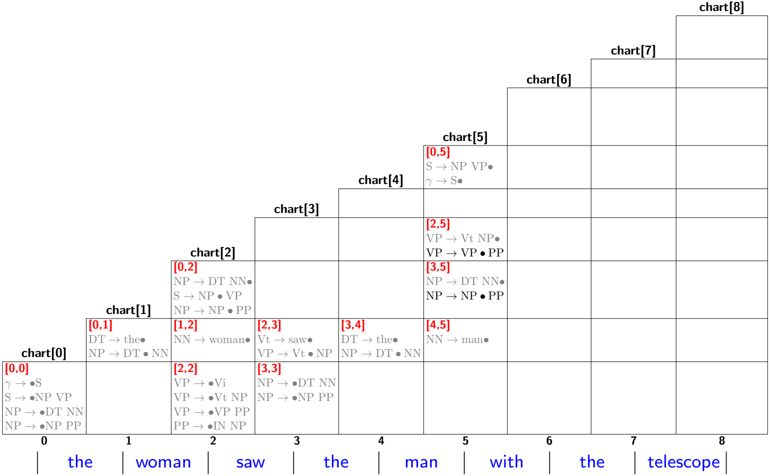
Earley Algorithm



Earley Algorithm



Earley Algorithm



Earley Algorithm

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
					[0,5] S → NP VP• γ → S•			
					[2,5] VP → Vt NP• VP → VP • PP			
		[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[3,5] NP → DT NN• NP → NP • PP			
	[0,1] DT → the• NP → DT • NN	[1,2] NN → woman•	[2,3] Vt → saw• VP → Vt • NP	[3,4] DT → the• NP → DT • NN	[4,5] NN → man•			
[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[3,3] NP → •DT NN NP → •NP PP		[5,5] PP → •IN NP			
0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
					[0,5] S → NP VP• γ → S•			
					[2,5] VP → Vt NP• VP → VP • PP			
		[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[3,5] NP → DT NN• NP → NP • PP			
	[0,1] DT → the• NP → DT • NN	[1,2] NN → woman•	[2,3] Vt → saw• VP → Vt • NP	[3,4] DT → the• NP → DT • NN	[4,5] NN → man•			
[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[3,3] NP → •DT NN NP → •NP PP		[5,5] PP → •IN NP			

0 | the | 1 | woman | 2 | saw | 3 | the | 4 | man | 5 | with | 6 | the | 7 | telescope | 8

Trees and Grammars
Constituency Parsing
Earley Algorithm
Dependency Parsing

Earley Algorithm

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
					[0,5] $S \rightarrow NP\ VP \bullet$ $\gamma \rightarrow S \bullet$			
					[2,5] $VP \rightarrow Vt\ NP \bullet$ $VP \rightarrow VP \bullet PP$			
		[0,2] $NP \rightarrow DT\ NN \bullet$ $S \rightarrow NP \bullet VP$ $NP \rightarrow NP \bullet PP$			[3,5] $NP \rightarrow DT\ NN \bullet$ $NP \rightarrow NP \bullet PP$			
	[0,1] $DT \rightarrow the \bullet$ $NP \rightarrow DT \bullet NN$	[1,2] $NN \rightarrow woman \bullet$	[2,3] $Vt \rightarrow saw \bullet$ $VP \rightarrow Vt \bullet NP$	[3,4] $DT \rightarrow the \bullet$ $NP \rightarrow DT \bullet NN$	[4,5] $NN \rightarrow man \bullet$			
[0,0] $\gamma \rightarrow \bullet S$ $S \rightarrow \bullet NP\ VP$ $NP \rightarrow \bullet DT\ NN$ $NP \rightarrow \bullet NP\ PP$		[2,2] $VP \rightarrow \bullet Vi$ $VP \rightarrow \bullet Vt\ NP$ $VP \rightarrow \bullet VP\ PP$ $PP \rightarrow \bullet IN\ NP$	[3,3] $NP \rightarrow \bullet DT\ NN$ $NP \rightarrow \bullet NP\ PP$		[5,5] $PP \rightarrow \bullet IN\ NP$			
0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
						[0,5] S → NP VP• γ → S•			
						[2,5] VP → Vt NP• VP → VP • PP			
			[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[3,5] NP → DT NN• NP → NP • PP			
		[0,1] DT → the• NP → DT • NN	[1,2] NN → woman•	[2,3] Vt → saw• VP → Vt • NP	[3,4] DT → the• NP → DT • NN	[4,5] NN → man•	[5,6] IN → with•		
	[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[3,3] NP → •DT NN NP → •NP PP		[5,5] PP → •IN NP			
	0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

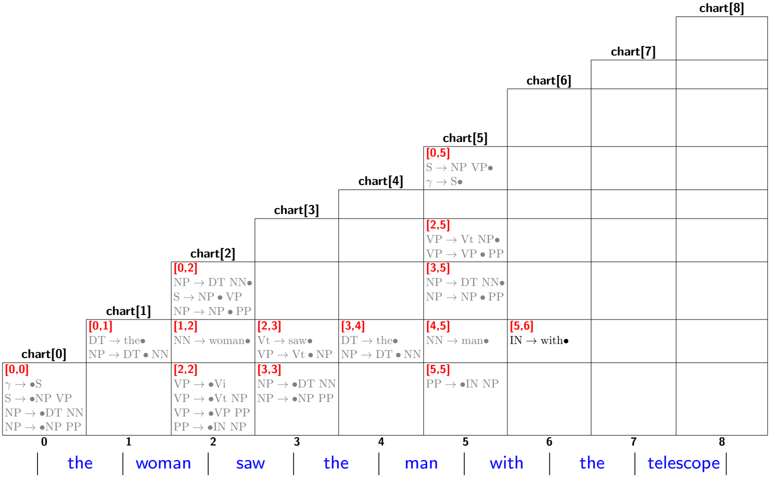
Trees and Grammars

Constituency Parsing

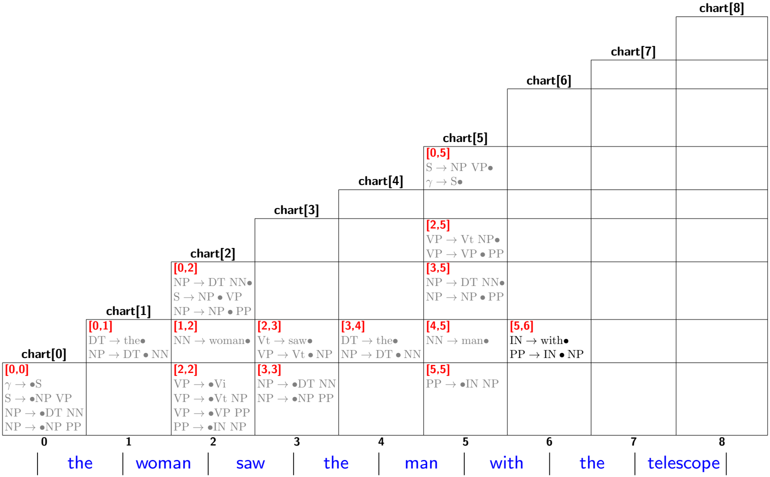
Earley Algorithm

Dependency Parsing

Earley Algorithm



Earley Algorithm



Earley Algorithm

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
					[0,5] S → NP VP• γ → S•			
					[2,5] VP → Vt NP• VP → VP • PP			
		[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[3,5] NP → DT NN• NP → NP • PP			
	[0,1] DT → the• NP → DT • NN	[1,2] NN → woman•	[2,3] Vt → saw• VP → Vt • NP	[3,4] DT → the• NP → DT • NN	[4,5] NN → man•	[5,6] IN → with• PP → IN • NP		
[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[3,3] NP → •DT NN NP → •NP PP		[5,5] PP → •IN NP			

0 | the | 1 | woman | 2 | saw | 3 | the | 4 | man | 5 | with | 6 | the | 7 | telescope | 8

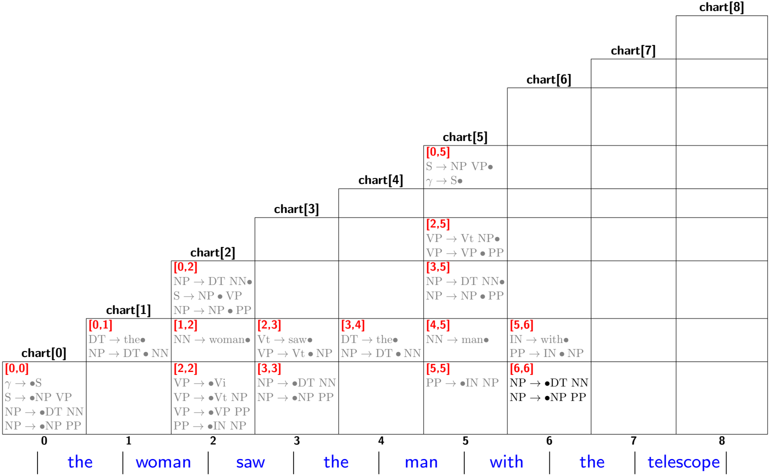
Trees and Grammars
Constituency Parsing
Earley Algorithm
Dependency Parsing

Earley Algorithm

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
					[0,5] S → NP VP• γ → S•			
					[2,5] VP → Vt NP• VP → VP • PP			
		[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[3,5] NP → DT NN• NP → NP • PP			
	[0,1] DT → the• NP → DT • NN	[1,2] NN → woman•	[2,3] Vt → saw• VP → Vt • NP	[3,4] DT → the• NP → DT • NN	[4,5] NN → man•	[5,6] IN → with• PP → IN • NP		
[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[3,3] NP → •DT NN NP → •NP PP		[5,5] PP → •IN NP	[6,6] NP → •DT NN NP → •NP PP		
0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm



Earley Algorithm

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
					[0,5] S → NP VP• γ → S•			
					[2,5] VP → Vt NP• VP → VP • PP			
		[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[3,5] NP → DT NN• NP → NP • PP			
	[0,1] DT → the• NP → DT • NN	[1,2] NN → woman•	[2,3] Vt → saw• VP → Vt • NP	[3,4] DT → the• NP → DT • NN	[4,5] NN → man•	[5,6] IN → with• PP → IN • NP	[6,7] DT → the•	
[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[3,3] NP → •DT NN NP → •NP PP		[5,5] PP → •IN NP	[6,6] NP → •DT NN NP → •NP PP		
0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm

chart[0]
- [0,0]: $\gamma \rightarrow \bullet$S; S $\rightarrow \bullet$NP VP; NP $\rightarrow \bullet$DT NN; NP $\rightarrow \bullet$NP PP

chart[1]
- [0,1]: DT $\rightarrow$ the$\bullet$; NP $\rightarrow$ DT $\bullet$ NN

chart[2]
- [0,2]: NP $\rightarrow$ DT NN$\bullet$; S $\rightarrow$ NP $\bullet$ VP; NP $\rightarrow$ NP $\bullet$ PP
- [1,2]: NN $\rightarrow$ woman$\bullet$
- [2,2]: VP $\rightarrow \bullet$Vi; VP $\rightarrow \bullet$Vt NP; VP $\rightarrow \bullet$VP PP; PP $\rightarrow \bullet$IN NP

chart[3]
- [2,3]: Vt $\rightarrow$ saw$\bullet$; VP $\rightarrow$ Vt $\bullet$ NP
- [3,3]: NP $\rightarrow \bullet$DT NN; NP $\rightarrow \bullet$NP PP

chart[4]
- [3,4]: DT $\rightarrow$ the$\bullet$; NP $\rightarrow$ DT $\bullet$ NN

chart[5]
- [0,5]: S $\rightarrow$ NP VP$\bullet$; $\gamma \rightarrow$ S$\bullet$
- [2,5]: VP $\rightarrow$ Vt NP$\bullet$; VP $\rightarrow$ VP $\bullet$ PP
- [3,5]: NP $\rightarrow$ DT NN$\bullet$; NP $\rightarrow$ NP $\bullet$ PP
- [4,5]: NN $\rightarrow$ man$\bullet$
- [5,5]: PP $\rightarrow \bullet$IN NP

chart[6]
- [5,6]: IN $\rightarrow$ with$\bullet$; PP $\rightarrow$ IN $\bullet$ NP
- [6,6]: NP $\rightarrow \bullet$DT NN; NP $\rightarrow \bullet$NP PP

chart[7]
- [6,7]: DT $\rightarrow$ the$\bullet$

chart[8]

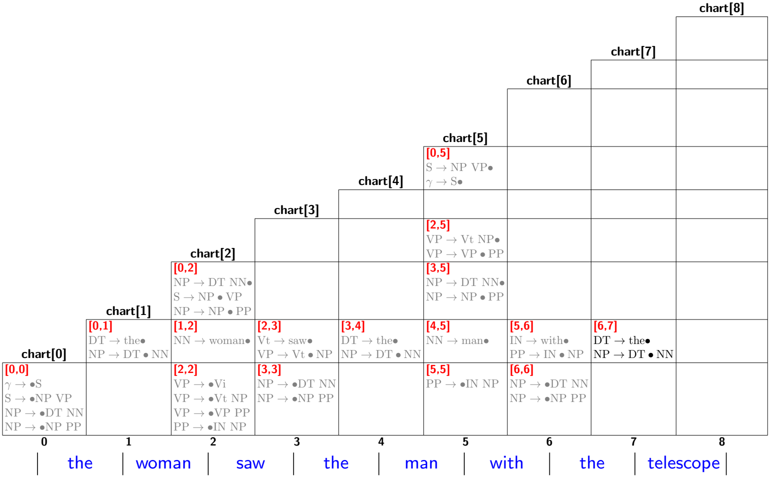
0		1		2		3		4		5		6		7		8
	the		woman		saw		the		man		with		the		telescope	

Earley Algorithm

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
					[0,5] S → NP VP• γ → S•			
					[2,5] VP → Vt NP• VP → VP • PP			
		[0,2] NP → DT NN• S → NP • VP NP → NP • PP			**[3,5]** NP → DT NN• NP → NP • PP			
	[0,1] DT → the• NP → DT • NN	**[1,2]** NN → woman•	**[2,3]** Vt → saw• VP → Vt • NP	**[3,4]** DT → the• NP → DT • NN	**[4,5]** NN → man•	**[5,6]** IN → with• PP → IN • NP	**[6,7]** DT → the•	
[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		**[2,2]** VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	**[3,3]** NP → •DT NN NP → •NP PP		**[5,5]** PP → •IN NP	**[6,6]** NP → •DT NN NP → •NP PP		
0	1	2	3	4	5	6	7	8

the	woman	saw	the	man	with	the	telescope

Earley Algorithm



Earley Algorithm

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
					[0,5] S → NP VP• γ → S•			
					[2,5] VP → Vt NP• VP → VP • PP			
		[0,2] NP → DT NN• S → NP • VP NP → NP • PP			**[3,5]** NP → DT NN• NP → NP • PP			
	[0,1] DT → the• NP → DT • NN	**[1,2]** NN → woman•	**[2,3]** Vt → saw• VP → Vt • NP	**[3,4]** DT → the• NP → DT • NN	**[4,5]** NN → man•	**[5,6]** IN → with• PP → IN • NP	**[6,7]** DT → the• NP → DT • NN	
[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		**[2,2]** VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	**[3,3]** NP → •DT NN NP → •NP PP		**[5,5]** PP → •IN NP	**[6,6]** NP → •DT NN NP → •NP PP		
0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

Trees and Grammars
Constituency Parsing
Earley Algorithm
Dependency Parsing

Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
	[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP	[0,1] DT → the• NP → DT • NN	[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[0,5] S → NP VP• γ → S•			
			[1,2] NN → woman•						
			[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[2,3] Vt → saw• VP → Vt • NP		[2,5] VP → Vt NP• VP → VP • PP			
				[3,3] NP → •DT NN NP → •NP PP	[3,4] DT → the• NP → DT • NN	[3,5] NP → DT NN• NP → NP • PP			
						[4,5] NN → man•			
						[5,5] PP → •IN NP	[5,6] IN → with• PP → IN • NP		
							[6,6] NP → •DT NN NP → •NP PP	[6,7] DT → the• NP → DT • NN	
									[7,8] NN → telescope•

0 the 1 woman 2 saw 3 the 4 man 5 with 6 the 7 telescope 8

Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
						[0,5] S → NP VP• γ → S•			
						[2,5] VP → Vt NP• VP → VP • PP			
			[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[3,5] NP → DT NN• NP → NP • PP			
		[0,1] DT → the• NP → DT • NN	[1,2] NN → woman•	[2,3] Vt → saw• VP → Vt • NP	[3,4] DT → the• NP → DT • NN	[4,5] NN → man•	[5,6] IN → with• PP → IN • NP	[6,7] DT → the• NP → DT • NN	[7,8] NN → telescope•
	[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[3,3] NP → •DT NN NP → •NP PP		[5,5] PP → •IN NP	[6,6] NP → •DT NN NP → •NP PP		
	0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
					[0,5] S → NP VP• γ → S•			
					[2,5] VP → Vt NP• VP → VP • PP			
		[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[3,5] NP → DT NN• NP → NP • PP			[6,8] NP → DT NN•
	[0,1] DT → the• NP → DT • NN	[1,2] NN → woman•	[2,3] Vt → saw• VP → Vt • NP	[3,4] DT → the• NP → DT • NN	[4,5] NN → man•	[5,6] IN → with• PP → IN • NP	[6,7] DT → the• NP → DT • NN	[7,8] NN → telescope•
[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[3,3] NP → •DT NN NP → •NP PP		[5,5] PP → •IN NP	[6,6] NP → •DT NN NP → •NP PP		
0	1	2	3	4	5	6	7	8

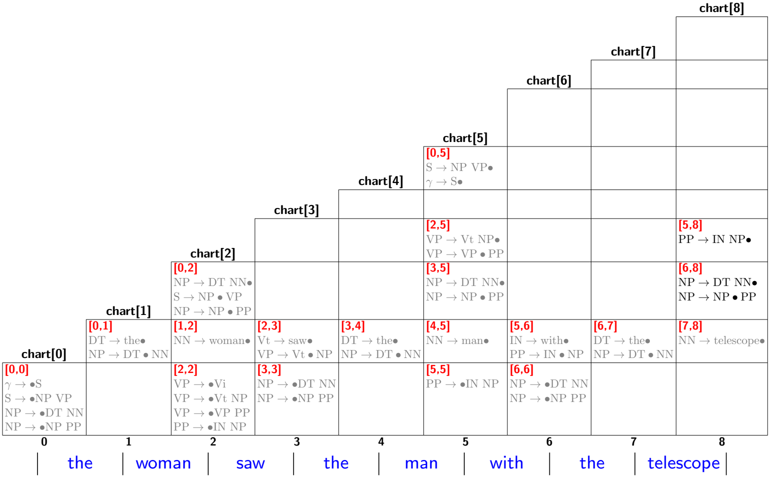
| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
						[0,5] S → NP VP• γ → S•			
						[2,5] VP → Vt NP• VP → VP • PP			
			[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[3,5] NP → DT NN• NP → NP • PP			[6,8] NP → DT NN•
		[0,1] DT → the• NP → DT • NN	[1,2] NN → woman•	[2,3] Vt → saw• VP → Vt • NP	[3,4] DT → the• NP → DT • NN	[4,5] NN → man•	[5,6] IN → with• PP → IN • NP	[6,7] DT → the• NP → DT • NN	[7,8] NN → telescope•
	[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[3,3] NP → •DT NN NP → •NP PP		[5,5] PP → •IN NP	[6,6] NP → •DT NN NP → •NP PP		
	0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm

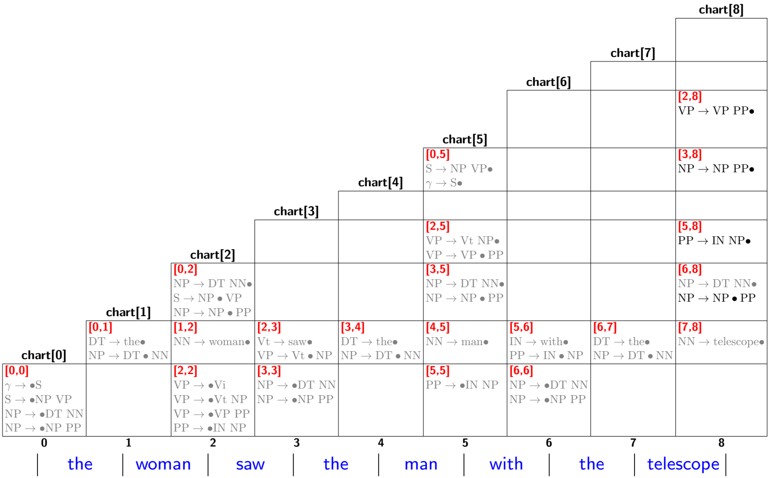


Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
						[0,5] $S \to NP\ VP\bullet$ $\gamma \to S\bullet$			
						[2,5] $VP \to Vt\ NP\bullet$ $VP \to VP \bullet PP$			[5,8] $PP \to IN\ NP\bullet$
			[0,2] $NP \to DT\ NN\bullet$ $S \to NP \bullet VP$ $NP \to NP \bullet PP$			[3,5] $NP \to DT\ NN\bullet$ $NP \to NP \bullet PP$			[6,8] $NP \to DT\ NN\bullet$ $NP \to NP \bullet PP$
		[0,1] $DT \to the\bullet$ $NP \to DT \bullet NN$	[1,2] $NN \to woman\bullet$	[2,3] $Vt \to saw\bullet$ $VP \to Vt \bullet NP$	[3,4] $DT \to the\bullet$ $NP \to DT \bullet NN$	[4,5] $NN \to man\bullet$	[5,6] $IN \to with\bullet$ $PP \to IN \bullet NP$	[6,7] $DT \to the\bullet$ $NP \to DT \bullet NN$	[7,8] $NN \to telescope\bullet$
	[0,0] $\gamma \to \bullet S$ $S \to \bullet NP\ VP$ $NP \to \bullet DT\ NN$ $NP \to \bullet NP\ PP$		[2,2] $VP \to \bullet Vi$ $VP \to \bullet Vt\ NP$ $VP \to \bullet VP\ PP$ $PP \to \bullet IN\ NP$	[3,3] $NP \to \bullet DT\ NN$ $NP \to \bullet NP\ PP$		[5,5] $PP \to \bullet IN\ NP$	[6,6] $NP \to \bullet DT\ NN$ $NP \to \bullet NP\ PP$		
	0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm



Earley Algorithm

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
								[2,8] VP → VP PP•
					[0,5] S → NP VP• γ → S•			[3,8] NP → NP PP•
					[2,5] VP → Vt NP• VP → VP • PP			[5,8] PP → IN NP•
		[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[3,5] NP → DT NN• NP → NP • PP			[6,8] NP → DT NN• NP → NP • PP
	[0,1] DT → the• NP → DT • NN	[1,2] NN → woman•	[2,3] Vt → saw• VP → Vt • NP	[3,4] DT → the• NP → DT • NN	[4,5] NN → man•	[5,6] IN → with• PP → IN • NP	[6,7] DT → the• NP → DT • NN	[7,8] NN → telescope•
[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[3,3] NP → •DT NN NP → •NP PP		[5,5] PP → •IN NP	[6,6] NP → •DT NN NP → •NP PP		
0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
								[0,8] S $\rightarrow$ NP VP$\bullet$
								[2,8] VP $\rightarrow$ VP PP$\bullet$ VP $\rightarrow$ VP $\bullet$ PP
					[0,5] S $\rightarrow$ NP VP$\bullet$ γ $\rightarrow$ S$\bullet$			[3,8] NP $\rightarrow$ NP PP$\bullet$
					[2,5] VP $\rightarrow$ Vt NP$\bullet$ VP $\rightarrow$ VP $\bullet$ PP			[5,8] PP $\rightarrow$ IN NP$\bullet$
		[0,2] NP $\rightarrow$ DT NN$\bullet$ S $\rightarrow$ NP $\bullet$ VP NP $\rightarrow$ NP $\bullet$ PP			[3,5] NP $\rightarrow$ DT NN$\bullet$ NP $\rightarrow$ NP $\bullet$ PP			[6,8] NP $\rightarrow$ DT NN$\bullet$ NP $\rightarrow$ NP $\bullet$ PP
	[0,1] DT $\rightarrow$ the$\bullet$ NP $\rightarrow$ DT $\bullet$ NN	[1,2] NN $\rightarrow$ woman$\bullet$	[2,3] Vt $\rightarrow$ saw$\bullet$ VP $\rightarrow$ Vt $\bullet$ NP	[3,4] DT $\rightarrow$ the$\bullet$ NP $\rightarrow$ DT $\bullet$ NN	[4,5] NN $\rightarrow$ man$\bullet$	[5,6] IN $\rightarrow$ with$\bullet$ PP $\rightarrow$ IN $\bullet$ NP	[6,7] DT $\rightarrow$ the$\bullet$ NP $\rightarrow$ DT $\bullet$ NN	[7,8] NN $\rightarrow$ telescope$\bullet$
[0,0] γ $\rightarrow$ $\bullet$S S $\rightarrow$ $\bullet$NP VP NP $\rightarrow$ $\bullet$DT NN NP $\rightarrow$ $\bullet$NP PP		[2,2] VP $\rightarrow$ $\bullet$Vi VP $\rightarrow$ $\bullet$Vt NP VP $\rightarrow$ $\bullet$VP PP PP $\rightarrow$ $\bullet$IN NP	[3,3] NP $\rightarrow$ $\bullet$DT NN NP $\rightarrow$ $\bullet$NP PP		[5,5] PP $\rightarrow$ $\bullet$IN NP	[6,6] NP $\rightarrow$ $\bullet$DT NN NP $\rightarrow$ $\bullet$NP PP		
0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
								[0,8] S → NP VP•
								[2,8] VP → VP PP• VP → VP • PP
					[0,5] S → NP VP• γ → S•			[3,8] NP → NP PP•
					[2,5] VP → Vt NP• VP → VP • PP			[5,8] PP → IN NP•
		[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[3,5] NP → DT NN• NP → NP • PP			[6,8] NP → DT NN• NP → NP • PP
	[0,1] DT → the• NP → DT • NN	[1,2] NN → woman•	[2,3] Vt → saw• VP → Vt • NP	[3,4] DT → the• NP → DT • NN	[4,5] NN → man•	[5,6] IN → with• PP → IN • NP	[6,7] DT → the• NP → DT • NN	[7,8] NN → telescope•
[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP		[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[3,3] NP → •DT NN NP → •NP PP		[5,5] PP → •IN NP	[6,6] NP → •DT NN NP → •NP PP		
0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
0	[0,0] $\gamma \to \bullet$S S $\to \bullet$NP VP NP $\to \bullet$DT NN NP $\to \bullet$NP PP	[0,1] DT $\to$ the$\bullet$ NP $\to$ DT $\bullet$ NN	[0,2] NP $\to$ DT NN$\bullet$ S $\to$ NP $\bullet$ VP NP $\to$ NP $\bullet$ PP			[0,5] S $\to$ NP VP$\bullet$ $\gamma \to$ S$\bullet$			[0,8] S $\to$ NP VP$\bullet$ $\gamma \to$ S$\bullet$
1			[1,2] NN $\to$ woman$\bullet$						
2			[2,2] VP $\to \bullet$Vi VP $\to \bullet$Vt NP VP $\to \bullet$VP PP PP $\to \bullet$IN NP	[2,3] Vt $\to$ saw$\bullet$ VP $\to$ Vt $\bullet$ NP		[2,5] VP $\to$ Vt NP$\bullet$ VP $\to$ VP $\bullet$ PP			[2,8] VP $\to$ VP PP$\bullet$ VP $\to$ VP $\bullet$ PP
3				[3,3] NP $\to \bullet$DT NN NP $\to \bullet$NP PP	[3,4] DT $\to$ the$\bullet$ NP $\to$ DT $\bullet$ NN	[3,5] NP $\to$ DT NN$\bullet$ NP $\to$ NP $\bullet$ PP			[3,8] NP $\to$ NP PP$\bullet$
4						[4,5] NN $\to$ man$\bullet$			
5						[5,5] PP $\to \bullet$IN NP	[5,6] IN $\to$ with$\bullet$ PP $\to$ IN $\bullet$ NP		[5,8] PP $\to$ IN NP$\bullet$
6							[6,6] NP $\to \bullet$DT NN NP $\to \bullet$NP PP	[6,7] DT $\to$ the$\bullet$ NP $\to$ DT $\bullet$ NN	[6,8] NP $\to$ DT NN$\bullet$ NP $\to$ NP $\bullet$ PP
7									[7,8] NN $\to$ telescope$\bullet$
8									

0 | the | 1 | woman | 2 | saw | 3 | the | 4 | man | 5 | with | 6 | the | 7 | telescope | 8

Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
	[0,0] $\gamma \rightarrow \bullet$S S $\rightarrow \bullet$NP VP NP $\rightarrow \bullet$DT NN NP $\rightarrow \bullet$NP PP	[0,1] DT $\rightarrow$ the$\bullet$ NP $\rightarrow$ DT $\bullet$ NN	[0,2] NP $\rightarrow$ DT NN$\bullet$ S $\rightarrow$ NP $\bullet$ VP NP $\rightarrow$ NP $\bullet$ PP			[0,5] S $\rightarrow$ NP VP$\bullet$ $\gamma \rightarrow$ S$\bullet$			[0,8] S $\rightarrow$ NP VP$\bullet$ $\gamma \rightarrow$ S$\bullet$
			[1,2] NN $\rightarrow$ woman$\bullet$						
			[2,2] VP $\rightarrow \bullet$Vi VP $\rightarrow \bullet$Vt NP VP $\rightarrow \bullet$VP PP PP $\rightarrow \bullet$IN NP	[2,3] Vt $\rightarrow$ saw$\bullet$ VP $\rightarrow$ Vt $\bullet$ NP		[2,5] VP $\rightarrow$ Vt NP$\bullet$ VP $\rightarrow$ VP $\bullet$ PP			[2,8] VP $\rightarrow$ VP PP$\bullet$ VP $\rightarrow$ VP $\bullet$ PP
				[3,3] NP $\rightarrow \bullet$DT NN NP $\rightarrow \bullet$NP PP	[3,4] DT $\rightarrow$ the$\bullet$ NP $\rightarrow$ DT $\bullet$ NN	[3,5] NP $\rightarrow$ DT NN$\bullet$ NP $\rightarrow$ NP $\bullet$ PP			[3,8] NP $\rightarrow$ NP PP$\bullet$
						[4,5] NN $\rightarrow$ man$\bullet$			
						[5,5] PP $\rightarrow \bullet$IN NP	[5,6] IN $\rightarrow$ with$\bullet$ PP $\rightarrow$ IN $\bullet$ NP		[5,8] PP $\rightarrow$ IN NP$\bullet$
							[6,6] NP $\rightarrow \bullet$DT NN NP $\rightarrow \bullet$NP PP	[6,7] DT $\rightarrow$ the$\bullet$ NP $\rightarrow$ DT $\bullet$ NN	[6,8] NP $\rightarrow$ DT NN$\bullet$ NP $\rightarrow$ NP $\bullet$ PP
									[7,8] NN $\rightarrow$ telescope$\bullet$
	0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
0	[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP	[0,1] DT → the• NP → DT • NN	[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[0,5] S → NP VP• γ → S•			[0,8] S → NP VP• γ → S•
1			[1,2] NN → woman•						
2			[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[2,3] Vt → saw• VP → Vt • NP		[2,5] VP → Vt NP• VP → VP • PP			[2,8] VP → VP PP• VP → VP • PP
3				[3,3] NP → •DT NN NP → •NP PP	[3,4] DT → the• NP → DT • NN	[3,5] NP → DT NN• NP → NP • PP			[3,8] NP → NP PP•
4						[4,5] NN → man•			
5						[5,5] PP → •IN NP	[5,6] IN → with• PP → IN • NP		[5,8] PP → IN NP•
6							[6,6] NP → •DT NN NP → •NP PP	[6,7] DT → the• NP → DT • NN	[6,8] NP → DT NN• NP → NP • PP
7									[7,8] NN → telescope•

0 1 2 3 4 5 6 7 8

| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
0	[0,0] $\gamma \to \bullet$S S $\to \bullet$NP VP NP $\to \bullet$DT NN NP $\to \bullet$NP PP	[0,1] DT $\to$ the$\bullet$ NP $\to$ DT $\bullet$ NN	[0,2] NP $\to$ DT NN$\bullet$ S $\to$ NP $\bullet$ VP NP $\to$ NP $\bullet$ PP			[0,5] S $\to$ NP VP$\bullet$ $\gamma \to$ S$\bullet$			[0,8] S $\to$ NP VP$\bullet$ $\gamma \to$ S$\bullet$
1			[1,2] NN $\to$ woman$\bullet$						
2			[2,2] VP $\to \bullet$Vi VP $\to \bullet$Vt NP VP $\to \bullet$VP PP PP $\to \bullet$IN NP	[2,3] Vt $\to$ saw$\bullet$ VP $\to$ Vt $\bullet$ NP		[2,5] VP $\to$ Vt NP$\bullet$ VP $\to$ VP $\bullet$ PP			[2,8] VP $\to$ VP PP$\bullet$ VP $\to$ VP $\bullet$ PP
3				[3,3] NP $\to \bullet$DT NN NP $\to \bullet$NP PP	[3,4] DT $\to$ the$\bullet$ NP $\to$ DT $\bullet$ NN	[3,5] NP $\to$ DT NN$\bullet$ NP $\to$ NP $\bullet$ PP			[3,8] NP $\to$ NP PP$\bullet$
4						[4,5] NN $\to$ man$\bullet$			
5						[5,5] PP $\to \bullet$IN NP	[5,6] IN $\to$ with$\bullet$ PP $\to$ IN $\bullet$ NP		[5,8] PP $\to$ IN NP$\bullet$
6							[6,6] NP $\to \bullet$DT NN NP $\to \bullet$NP PP	[6,7] DT $\to$ the$\bullet$ NP $\to$ DT $\bullet$ NN	[6,8] NP $\to$ DT NN$\bullet$ NP $\to$ NP $\bullet$ PP
7									[7,8] NN $\to$ telescope$\bullet$
8									[8,8] PP $\to \bullet$IN NP

0 | the | 1 | woman | 2 | saw | 3 | the | 4 | man | 5 | with | 6 | the | 7 | telescope | 8

Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
0	[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP	[0,1] DT → the• NP → DT • NN	[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[0,5] S → NP VP• γ → S•			[0,8] S → NP VP• γ → S•
1			[1,2] NN → woman•						
2			[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[2,3] Vt → saw• VP → Vt • NP		[2,5] VP → Vt NP• VP → VP • PP			[2,8] VP → VP PP• VP → VP • PP
3				[3,3] NP → •DT NN NP → •NP PP	[3,4] DT → the• NP → DT • NN	[3,5] NP → DT NN• NP → NP • PP			[3,8] NP → NP PP•
4						[4,5] NN → man•			
5						[5,5] PP → •IN NP	[5,6] IN → with• PP → IN • NP		[5,8] PP → IN NP•
6							[6,6] NP → •DT NN NP → •NP PP	[6,7] DT → the• NP → DT • NN	[6,8] NP → DT NN• NP → NP • PP
7									[7,8] NN → telescope•
8									[8,8] PP → •IN NP

0 | the | 1 | woman | 2 | saw | 3 | the | 4 | man | 5 | with | 6 | the | 7 | telescope | 8

Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
0	[0,0] $\gamma \to \bullet S$ $S \to \bullet NP\ VP$ $NP \to \bullet DT\ NN$ $NP \to \bullet NP\ PP$	[0,1] $DT \to the \bullet$ $NP \to DT \bullet NN$	[0,2] $NP \to DT\ NN \bullet$ $S \to NP \bullet VP$ $NP \to NP \bullet PP$			[0,5] $S \to NP\ VP \bullet$ $\gamma \to S \bullet$			[0,8] $S \to NP\ VP \bullet$ $\gamma \to S \bullet$
1			[1,2] $NN \to woman \bullet$						
2			[2,2] $VP \to \bullet Vi$ $VP \to \bullet Vt\ NP$ $VP \to \bullet VP\ PP$ $PP \to \bullet IN\ NP$	[2,3] $Vt \to saw \bullet$ $VP \to Vt \bullet NP$		[2,5] $VP \to Vt\ NP \bullet$ $VP \to VP \bullet PP$			[2,8] $VP \to VP\ PP \bullet$ $VP \to VP \bullet PP$ $VP \to Vt\ NP \bullet$
3				[3,3] $NP \to \bullet DT\ NN$ $NP \to \bullet NP\ PP$	[3,4] $DT \to the \bullet$ $NP \to DT \bullet NN$	[3,5] $NP \to DT\ NN \bullet$ $NP \to NP \bullet PP$			[3,8] $NP \to NP\ PP \bullet$ $NP \to NP \bullet PP$
4						[4,5] $NN \to man \bullet$			
5						[5,5] $PP \to \bullet IN\ NP$	[5,6] $IN \to with \bullet$ $PP \to IN \bullet NP$		[5,8] $PP \to IN\ NP \bullet$
6							[6,6] $NP \to \bullet DT\ NN$ $NP \to \bullet NP\ PP$	[6,7] $DT \to the \bullet$ $NP \to DT \bullet NN$	[6,8] $NP \to DT\ NN \bullet$ $NP \to NP \bullet PP$
7									[7,8] $NN \to telescope \bullet$
8									[8,8] $PP \to \bullet IN\ NP$

0 | the | 1 | woman | 2 | saw | 3 | the | 4 | man | 5 | with | 6 | the | 7 | telescope | 8

Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
0	[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP	[0,1] DT → the• NP → DT • NN	[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[0,5] S → NP VP• γ → S•			[0,8] S → NP VP• γ → S•
1			[1,2] NN → woman•						
2			[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[2,3] Vt → saw• VP → Vt • NP		[2,5] VP → Vt NP• VP → VP • PP			[2,8] VP → VP PP• VP → VP • PP VP → Vt NP•
3				[3,3] NP → •DT NN NP → •NP PP	[3,4] DT → the• NP → DT • NN	[3,5] NP → DT NN• NP → NP • PP			[3,8] NP → NP PP• NP → NP • PP
4						[4,5] NN → man•			
5						[5,5] PP → •IN NP	[5,6] IN → with• PP → IN • NP		[5,8] PP → IN NP•
6							[6,6] NP → •DT NN NP → •NP PP	[6,7] DT → the• NP → DT • NN	[6,8] NP → DT NN• NP → NP • PP
7									[7,8] NN → telescope•
8									[8,8] PP → •IN NP

0 the 1 woman 2 saw 3 the 4 man 5 with 6 the 7 telescope 8

Earley Algorithm

chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP	[0,1] DT → the• NP → DT • NN	[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[0,5] S → NP VP• γ → S•			[0,8] S → NP VP• γ → S•
		[1,2] NN → woman•						
		[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[2,3] Vt → saw• VP → Vt • NP		[2,5] VP → Vt NP• VP → VP • PP			[2,8] VP → VP PP• VP → VP • PP VP → Vt NP•
			[3,3] NP → •DT NN NP → •NP PP	[3,4] DT → the• NP → DT • NN	[3,5] NP → DT NN• NP → NP • PP			[3,8] NP → NP PP• NP → NP • PP
					[4,5] NN → man•			
					[5,5] PP → •IN NP	[5,6] IN → with• PP → IN • NP		[5,8] PP → IN NP•
						[6,6] NP → •DT NN NP → •NP PP	[6,7] DT → the• NP → DT • NN	[6,8] NP → DT NN• NP → NP • PP
								[7,8] NN → telescope•
								[8,8] PP → •IN NP

0 1 2 3 4 5 6 7 8

| the | woman | saw | the | man | with | the | telescope |

Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
0	[0,0] $\gamma \rightarrow \bullet S$ $S \rightarrow \bullet NP\ VP$ $NP \rightarrow \bullet DT\ NN$ $NP \rightarrow \bullet NP\ PP$	[0,1] $DT \rightarrow the \bullet$ $NP \rightarrow DT \bullet NN$	[0,2] $NP \rightarrow DT\ NN \bullet$ $S \rightarrow NP \bullet VP$ $NP \rightarrow NP \bullet PP$			[0,5] $S \rightarrow NP\ VP \bullet$ $\gamma \rightarrow S \bullet$			[0,8] $S \rightarrow NP\ VP \bullet$ $\gamma \rightarrow S \bullet$
1			[1,2] $NN \rightarrow woman \bullet$						
2			[2,2] $VP \rightarrow \bullet Vi$ $VP \rightarrow \bullet Vt\ NP$ $VP \rightarrow \bullet VP\ PP$ $PP \rightarrow \bullet IN\ NP$	[2,3] $Vt \rightarrow saw \bullet$ $VP \rightarrow Vt \bullet NP$		[2,5] $VP \rightarrow Vt\ NP \bullet$ $VP \rightarrow VP \bullet PP$			[2,8] $VP \rightarrow VP\ PP \bullet$ $VP \rightarrow VP \bullet PP$ $VP \rightarrow Vt\ NP \bullet$
3				[3,3] $NP \rightarrow \bullet DT\ NN$ $NP \rightarrow \bullet NP\ PP$	[3,4] $DT \rightarrow the \bullet$ $NP \rightarrow DT \bullet NN$	[3,5] $NP \rightarrow DT\ NN \bullet$ $NP \rightarrow NP \bullet PP$			[3,8] $NP \rightarrow NP\ PP \bullet$ $NP \rightarrow NP \bullet PP$
4						[4,5] $NN \rightarrow man \bullet$			
5						[5,5] $PP \rightarrow \bullet IN\ NP$	[5,6] $IN \rightarrow with \bullet$ $PP \rightarrow IN \bullet NP$		[5,8] $PP \rightarrow IN\ NP \bullet$
6							[6,6] $NP \rightarrow \bullet DT\ NN$ $NP \rightarrow \bullet NP\ PP$	[6,7] $DT \rightarrow the \bullet$ $NP \rightarrow DT \bullet NN$	[6,8] $NP \rightarrow DT\ NN \bullet$ $NP \rightarrow NP \bullet PP$
7									[7,8] $NN \rightarrow telescope \bullet$
8									[8,8] $PP \rightarrow \bullet IN\ NP$

0 | the | 1 | woman | 2 | saw | 3 | the | 4 | man | 5 | with | 6 | the | 7 | telescope | 8

Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
	[0,0] $\gamma \to \bullet$S S $\to \bullet$NP VP NP $\to \bullet$DT NN NP $\to \bullet$NP PP	[0,1] DT $\to$ the$\bullet$ NP $\to$ DT $\bullet$ NN	[0,2] NP $\to$ DT NN$\bullet$ S $\to$ NP $\bullet$ VP NP $\to$ NP $\bullet$ PP			[0,5] S $\to$ NP VP$\bullet$ $\gamma \to$ S$\bullet$			[0,8] S $\to$ NP VP$\bullet$ $\gamma \to$ S$\bullet$
			[1,2] NN $\to$ woman$\bullet$						
			[2,2] VP $\to \bullet$Vi VP $\to \bullet$Vt NP VP $\to \bullet$VP PP PP $\to \bullet$IN NP	[2,3] Vt $\to$ saw$\bullet$ VP $\to$ Vt $\bullet$ NP		[2,5] VP $\to$ Vt NP$\bullet$ VP $\to$ VP $\bullet$ PP			[2,8] VP $\to$ VP PP$\bullet$ VP $\to$ VP $\bullet$ PP VP $\to$ Vt NP$\bullet$
				[3,3] NP $\to \bullet$DT NN NP $\to \bullet$NP PP	[3,4] DT $\to$ the$\bullet$ NP $\to$ DT $\bullet$ NN	[3,5] NP $\to$ DT NN$\bullet$ NP $\to$ NP $\bullet$ PP			[3,8] NP $\to$ NP PP$\bullet$ NP $\to$ NP $\bullet$ PP
						[4,5] NN $\to$ man$\bullet$			
						[5,5] PP $\to \bullet$IN NP	[5,6] IN $\to$ with$\bullet$ PP $\to$ IN $\bullet$ NP		[5,8] PP $\to$ IN NP$\bullet$
							[6,6] NP $\to \bullet$DT NN NP $\to \bullet$NP PP	[6,7] DT $\to$ the$\bullet$ NP $\to$ DT $\bullet$ NN	[6,8] NP $\to$ DT NN$\bullet$ NP $\to$ NP $\bullet$ PP
									[7,8] NN $\to$ telescope$\bullet$
									[8,8] PP $\to \bullet$IN NP
	0	1	2	3	4	5	6	7	8

| the | woman | saw | the | man | with | the | telescope |

Trees and Grammars
Constituency Parsing
Earley Algorithm
Dependency Parsing

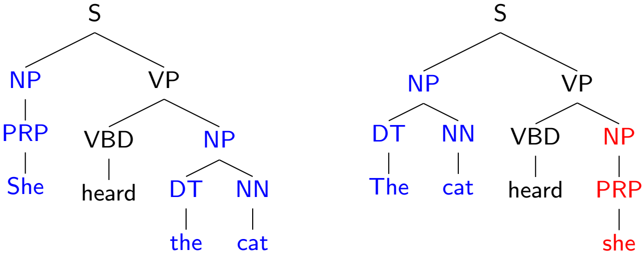
Earley Algorithm

	chart[0]	chart[1]	chart[2]	chart[3]	chart[4]	chart[5]	chart[6]	chart[7]	chart[8]
0	[0,0] γ → •S S → •NP VP NP → •DT NN NP → •NP PP	[0,1] DT → the• NP → DT • NN	[0,2] NP → DT NN• S → NP • VP NP → NP • PP			[0,5] S → NP VP• γ → S•			[0,8] S → NP VP• γ → S•
1			[1,2] NN → woman•						
2			[2,2] VP → •Vi VP → •Vt NP VP → •VP PP PP → •IN NP	[2,3] Vt → saw• VP → Vt • NP		[2,5] VP → Vt NP• VP → VP • PP			[2,8] VP → VP PP• VP → VP • PP VP → Vt NP•
3				[3,3] NP → •DT NN NP → •NP PP	[3,4] DT → the• NP → DT • NN	[3,5] NP → DT NN• NP → NP • PP			[3,8] NP → NP PP• NP → NP • PP
4						[4,5] NN → man•			
5						[5,5] PP → •IN NP	[5,6] IN → with• PP → IN • NP		[5,8] PP → IN NP•
6							[6,6] NP → •DT NN NP → •NP PP	[6,7] DT → the• NP → DT • NN	[6,8] NP → DT NN• NP → NP • PP
7									[7,8] NN → telescope•
8									[8,8] PP → •IN NP

0 | the | 1 | woman | 2 | saw | 3 | the | 4 | man | 5 | with | 6 | the | 7 | telescope | 8

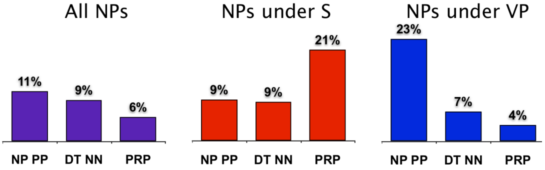
Why *context-free* ?

- Context-free means *independent of the context*, i.e, assumes that any expansion of a non-terminal is applicable, regardless of the context in which it occurs.



Natural Language is not Context-Free

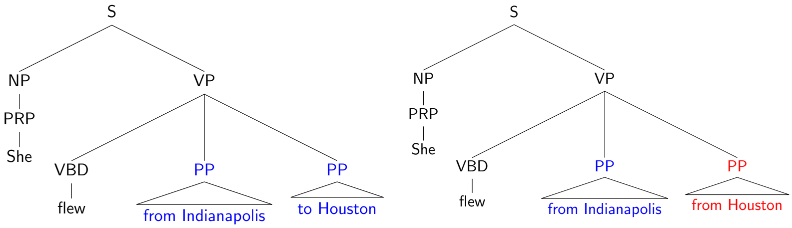
- NP expansion (for instance) is highly dependent on the parent of the NP



- Complete context independence is a too strong independence assumption for natural language.

Natural Language is not Context-Free

- The application of a rule may affect the applicability of others



Natural Language is not Context-Free

- May contain non-projective structures:

John saw the dog yesterday which was a Yorkshire Terrier

Outline

1 Trees and Grammars

2 Constituency Parsing
- CKY Algorithm
- Earley Algorithm

3 Dependency Parsing
- Dependency Trees
- Arc-factored Dependency Parsing
- Parsing Projective Structures
- Parsing non-Projective Structures
- Transition-Based parsers

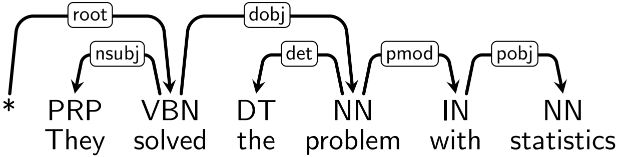
Outline

1 Trees and Grammars

2 Constituency Parsing
- CKY Algorithm
- Earley Algorithm

3 Dependency Parsing
- Dependency Trees
- Arc-factored Dependency Parsing
- Parsing Projective Structures
- Parsing non-Projective Structures
- Transition-Based parsers

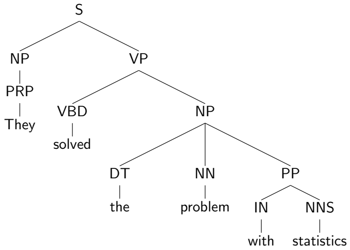
Dependency Trees



Trees and Grammars
Constituency Parsing
Dependency Parsing
Dependency Trees

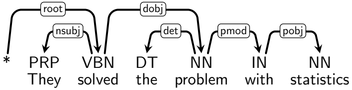
Theories of Syntactic Structure

Constituent Trees



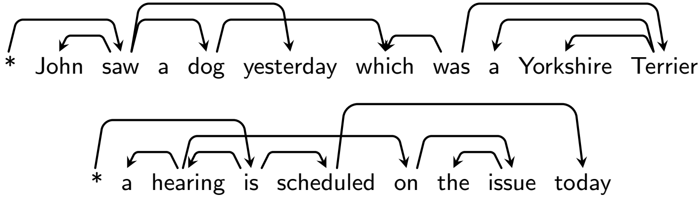
- Main element: constituents (or phrases, or bracketings)
- Constituents = abstract linguistic units
- Focus on word order
- Builds nested trees

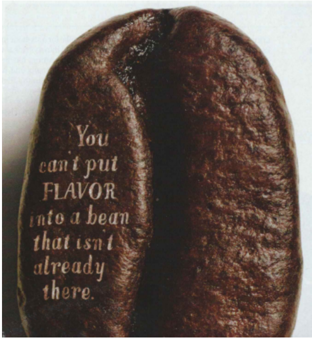
Dependency Trees



- Main element: dependency
- Focus on relations between words
- Nicely handles free word order (*fish the cat eats**) and non-projectivity (*John saw the dog yesterday which was a Yorkshire Terrier*)
- Builds dependency graphs

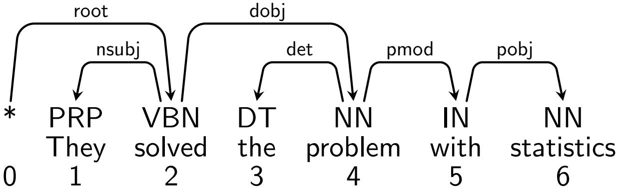
Non-projective dependency trees





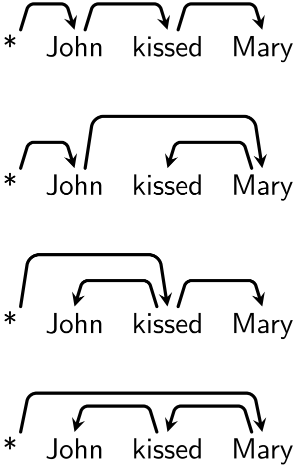
©Starbuck's 2013

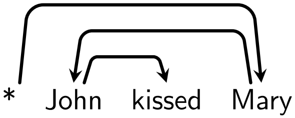
Dependency trees



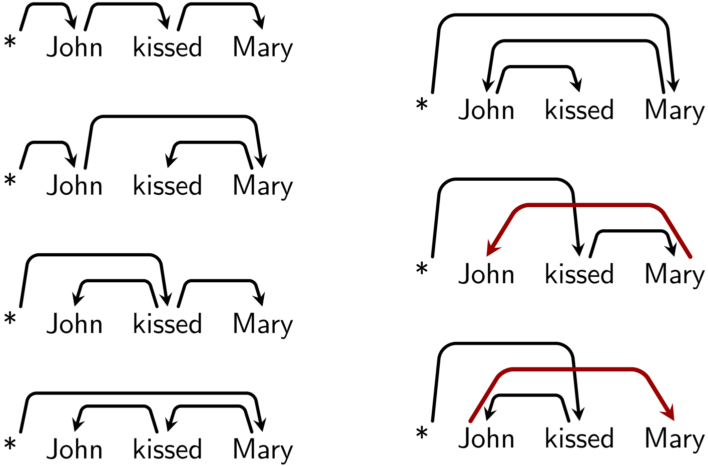
- $*$ is a special *root* symbol
- Each dependency is a tuple (h, m, l) where
  - h is the index of the head word (root is 0)
  - m is the index of the modifier word
  - l is a dependency label
  - e.g.: $(0, 2, \text{root})$, $(2, 1, \text{nsubj})$, $(2, 5, \text{dobj})$, $(4, 3, \text{det})$, $(4, 5, \text{pmod})$, $(5, 6, \text{pobj})$
- Sometimes we just consider unlabeled dependencies

Dependency trees for “John kissed Mary”

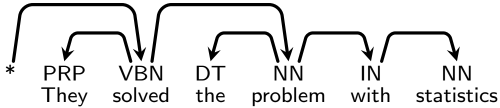




Dependency trees for "John kissed Mary"

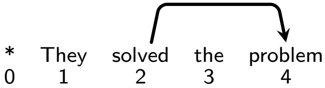


Conditions on Dependency Structures



- $\mathbf{y}$ is a dependency tree if:
  (a) Each non-root token has exactly an incoming arc (i.e. one parent)
  (b) The graph is connected
  (c) There are no cycles
  - That is, dependency arcs form a directed tree rooted at $*$
- $\mathbf{y}$ is a projective dependency tree if:
  - Is a dependency tree
  - There are no crossing dependencies
- Note that a projective tree is also in the non-projective set –must be read as non-*necessarily*-projective

Some Notation



Given a sentence with n words:

- $\mathcal{D}$ is the set of all possible dependencies that can be assigned to the sentence. Eg.

$$\begin{aligned}\mathcal{D} = \quad \{ \quad & (0,1),(0,2),(0,3),(0,4),(1,2),(1,3),(1,4) \\ & (2,1),(2,3),(2,4),(3,1),(3,2),(3,4) \\ & (4,1),(4,2),(4,3) \quad \}\end{aligned}$$

- $\mathbf{y}$ is a valid parse for s if:
  - $\mathbf{y} \subseteq \mathcal{D}$
  - $\mathbf{y}$ is a dependency tree
- $\mathcal{Y} \subseteq 2^{\mathcal{D}}$ is the set of all valid dependency trees for the sentence

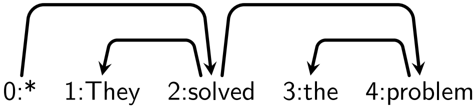
Outline

1 Trees and Grammars

2 Constituency Parsing
- CKY Algorithm
- Earley Algorithm

3 Dependency Parsing
- Dependency Trees
- Arc-factored Dependency Parsing
- Parsing Projective Structures
- Parsing non-Projective Structures
- Transition-Based parsers

Probabilistic Arc-Factored Dependency Parsing



- Assume we have $p(\text{modifier word} \mid \text{head word})$
- In a probabilistic arc-factored model:

$$
\begin{aligned}
p(\mathbf{x}, \mathbf{y}) &= p(\mathbf{x}, (*, 2), (2, 1), (2, 4), (4, 3)) \\
&= p(\mathbf{x}_2, (*, 2)\ ) \times p(\mathbf{x}, (2, 1), (2, 4), (4, 3) \mid \mathbf{x}_2, (*, 2)) \\
&= p(*) \times p(\mathbf{x}_2 \mid *) \times p(\mathbf{x}, (2, 1), (2, 4), (4, 3) \mid \mathbf{x}_2, (*, 2)) \\
&= \ldots \\
&= p(\mathbf{x}_2 \mid *) \times p(\mathbf{x}_1 \mid \mathbf{x}_2) \times p(\mathbf{x}_4 \mid \mathbf{x}_2) \times p(\mathbf{x}_3 \mid \mathbf{x}_4) \\
&= \prod_{(h,m) \in \mathbf{y}} p(\mathbf{x}_m \mid \mathbf{x}_h)
\end{aligned}
$$

- Note that we assume independence between arcs

Towards Linear Arc-Factored Dependency Parsing

- Consider an arc-factored probabilistic model

$$p(\mathbf{x}, \mathbf{y}) = \prod_{(h,m) \in \mathbf{y}} p(\mathbf{x}_m \mid \mathbf{x}_h)$$

- Prediction is:

$$\begin{aligned}
\operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} p(\mathbf{x}, \mathbf{y}) &= \operatorname*{argmax}_{\mathbf{y}} \prod_{(h,m) \in \mathbf{y}} p(\mathbf{x}_m \mid \mathbf{x}_h) \\
&= \operatorname*{argmax}_{\mathbf{y}} \exp \left\{ \sum_{(h,m) \in \mathbf{y}} \log p(\mathbf{x}_m \mid \mathbf{x}_h) \right\} \\
&= \operatorname*{argmax}_{\mathbf{y}} \sum_{(h,m) \in \mathbf{y}} \log p(\mathbf{x}_m \mid \mathbf{x}_h) \\
&= \operatorname*{argmax}_{\mathbf{y}} \sum_{(h,m) \in \mathbf{y}} \operatorname{score}(\mathbf{x}, h, m)
\end{aligned}$$

where $\operatorname{score}(\mathbf{x}, h, m) = \log p(\mathbf{x}_m \mid \mathbf{x}_h)$

A CRF for Arc-Factored Dependency Parsing

- A log-linear distribution of trees $\mathbf{y}$ given $\mathbf{x}$

$$p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) = \frac{\exp(\sum_{(h,m,l)\in\mathbf{y}} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, h, m, l))}{Z(\mathbf{x}; \mathbf{w})}$$

- $\mathbf{f}(\mathbf{x}, h, m)$ is a vector of d features of (h, m, l) assigned to x
- $\mathbf{w} \in \mathbb{R}^d$ are the parameters of the model
- $Z(\mathbf{x}; \mathbf{w}) = \sum_{\mathbf{y}\in\mathcal{Y}} \exp(\sum_{(h,m,l)\in\mathbf{y}} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, h, m, l))$
- Prediction is linear:

$$\begin{aligned}
\operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}^*} P(\mathbf{y}|\mathbf{x}; \mathbf{w}) &= \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}^*} \frac{\exp(\sum_{(h,m,l)\in\mathbf{y}} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, h, m, l))}{Z(\mathbf{x}; \mathbf{w})} \\
&= \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}^*} \sum_{(h,m,l)\in\mathbf{y}} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, h, m, l)
\end{aligned}$$

Features in Arc-Factored Dependency Parsing

Trees and Grammars
Constituency Parsing
Dependency Parsing
Arc-factored Dependency Parsing

$\mathbf{f}(\mathbf{x}, l, h, m)$: a vector of features of (h, m, l) assigned to x

- As in PoS tagging or NERC, we typically use indicator features
- Templates in (McDonald et al 2005):

word features
h-word, h-pos
h-word
h-pos
m-word, m-pos
m-word
m-pos

dependency features
h-word, h-pos, m-word, m-pos
h-pos, m-word, m-pos
h-word, m-word, m-pos
h-word, h-pos, m-pos
h-word, h-pos, m-word
h-word, m-word
h-pos, m-pos

- Example: (feature template + dependency direction)

$$\mathbf{f}_j(\mathbf{x}, h, m, l) = \begin{cases} 1 & \text{if } \mathrm{word}(h) = \textit{solve} \text{ and } \mathrm{word}(m) = \textit{problem} \\ & \quad \text{and } l = \textit{dobj} \text{ and } h < m \\ 0 & \text{otherwise} \end{cases}$$

A CRF for Arc-Factored Dependency Parsing

$$p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) = \frac{\exp(\sum_{(h,m,l) \in \mathbf{y}} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, h, m, l))}{Z(\mathbf{x}; \mathbf{w})}$$

- Parameter estimation: Learn parameters $\mathbf{w}$ given training data

$$\left\{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), \ldots, (\mathbf{x}^{(m)}, \mathbf{y}^{(m)})\right\}$$

- Decoding: predict the best dependency tree for $\mathbf{x}$

$$\operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} \mathrm{P}(\mathbf{y} | \mathbf{x}; \mathbf{w})$$

when

  - $\mathcal{Y}$ is the set of projective trees for $\mathbf{x}$
  - $\mathcal{Y}$ is the set of non-projective trees for $\mathbf{x}$

Parameter Estimation: CRFs for Parsing

...analogous to CRFs for Tagging

- Goal: Estimate $\mathbf{w}$ given a training set

$$\left\{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), \ldots, (\mathbf{x}^{(m)}, \mathbf{y}^{(m)})\right\}$$

- Define the conditional log-likelihood of the data:

$$L(\mathbf{w}) = \frac{1}{m} \sum_{k=1}^{m} \log \mathrm{P}(\mathbf{y}^{(k)} | \mathbf{x}^{(k)}; \mathbf{w})$$

  $L(\mathbf{w})$ measures how well $\mathbf{w}$ explains the data. A good value for $\mathbf{w}$ will give a high value for $\mathrm{P}(\mathbf{y}^{(k)} | \mathbf{x}^{(k)}; \mathbf{w})$ for all training examples $k = 1 \ldots m$.

- We want $\mathbf{w}$ that maximizes $L(\mathbf{w})$

Learning the Parameters of a CRF

...analogous to CRFs for Tagging

Trees and Grammars

Constituency Parsing

Dependency Parsing

Arc-factored Dependency Parsing

- Consider a regularized objective:

$$\mathbf{w}^* = \underset{\mathbf{w} \in \mathbb{R}^D}{\operatorname{argmax}} L(\mathbf{w}) - \frac{\lambda}{2}||\mathbf{w}||^2$$

where

  - The first term is the log-likelihood of the data
  - The second term is a regularization term, it penalizes solutions with large norm
  - λ is a parameter to control the trade-off between fitting the data and model complexity

Learning the Parameters of a CRF

...analogous to CRFs for Tagging

- Find

$$\mathbf{w}^* = \operatorname*{argmax}_{\mathbf{w} \in \mathbb{R}^D} L(\mathbf{w}) - \frac{\lambda}{2} ||\mathbf{w}||^2$$

- In general there is no analytical solution to this optimization
- We use iterative techniques, i.e. gradient-based optimization
  1. Initialize $\mathbf{w} = \mathbf{0}$
  2. Take derivatives of $L(\mathbf{w}) - \frac{\lambda}{2}||\mathbf{w}||^2$, compute gradient
  3. Move $\mathbf{w}$ in steps proportional to the gradient
  4. Repeat steps 2 and 3 until convergence

Computing the gradient

. . . analogous to CRFs for Tagging

$$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}_j} = \frac{1}{m}\sum_{k=1}^{m} \mathbf{f}_j(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) - \sum_{k=1}^{m} \sum_{\mathbf{y} \in \mathcal{Y}^*} \mathrm{P}(\mathbf{y}|\mathbf{x}^{(k)}; \mathbf{w})\ \mathbf{f}_j(\mathbf{x}^{(k)}, \mathbf{y})$$

where

$$\mathbf{f}(\mathbf{x}, \mathbf{y}) = \sum_{(h,m,l) \in \mathbf{y}} \mathbf{f}_j(\mathbf{x}, h, m, l)$$

- First term: observed mean feature value
- Second term: expected feature value under current $\mathbf{w}$

Computing the gradient

. . . analogous to CRFs for Tagging

- The first term is easy to compute, by counting explicitly

$$\frac{1}{m}\sum_{k=1}^{m}\sum_{(h,m,l)\in\mathbf{y}^{(k)}}\mathbf{f}_j(\mathbf{x},h,m,l)$$

- The second term is more involved,

$$\sum_{k=1}^{m}\sum_{\mathbf{y}\in\mathcal{Y}}\mathrm{P}(\mathbf{y}|\mathbf{x}^{(k)};\mathbf{w})\sum_{(h,m,l)\in\mathbf{y}}\mathbf{f}_j(\mathbf{x}^{(k)},h,m,l)$$

  because it sums over all sequences $\mathbf{y}\in\mathcal{Y}$

- There exist efficient algorithms for summing over $\mathcal{Y}$, both for projective and non-projective sets of trees

Outline

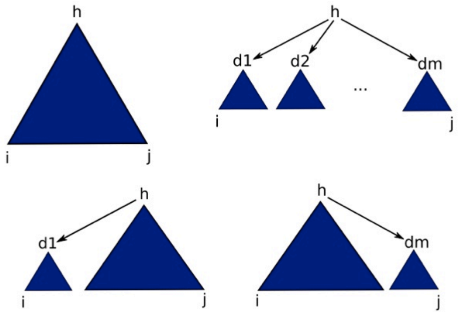
1 Trees and Grammars

2 Constituency Parsing
- CKY Algorithm
- Earley Algorithm

3 Dependency Parsing
- Dependency Trees
- Arc-factored Dependency Parsing
- Parsing Projective Structures
- Parsing non-Projective Structures
- Transition-Based parsers

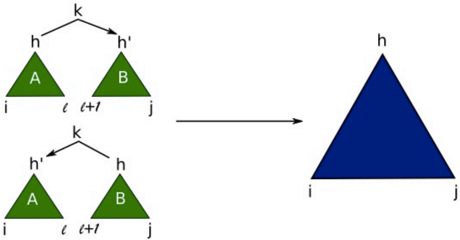
Parsing Projective Structures (I)

- Any projective tree can be written as the combination of:
  - two smaller *adjacent* projective trees and
  - a dependency connecting their roots



Parsing Projective Structures (II)

- The algorithm is a variation of CKY
- $\pi[i, j, h]$: score of dependency tree from i to j with head h



$$\pi[i,j,h] = \max_{\substack{i \le l < j \\ 1 \le k \le K}} \left\{ \begin{array}{l} \max_{l < h' \le j} \pi[i,l,h] + \pi[l+1,j,h'] + \mathbf{w} \cdot \mathbf{f}(\mathbf{x},h,h') \quad , \\ \max_{i \le h' \le l} \pi[i,l,h'] + \pi[l+1,j,h] + \mathbf{w} \cdot \mathbf{f}(\mathbf{x},h,h') \end{array} \right\}$$

- Cost: $O(Kn^5)$

Parsing Projective Structures (III)

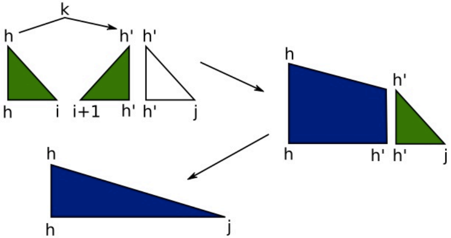
Trees and Grammars

Constituency Parsing

Dependency Parsing

Parsing Projective Structures

- (Eisner 1996), (Eisner 2000): an algorithm in $O(Kn^3)$
- Main idea: split constituents in half so that heads are at the boundary



Outline

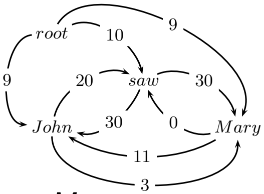
1 Trees and Grammars

2 Constituency Parsing
- CKY Algorithm
- Earley Algorithm

3 Dependency Parsing
- Dependency Trees
- Arc-factored Dependency Parsing
- Parsing Projective Structures
- Parsing non-Projective Structures
- Transition-Based parsers

Parsing Non-Projective Structures

- (McDonald et al 2005): non-projective parsing as maximum-spanning trees, using the Chu-Liu-Edmonds algorithm

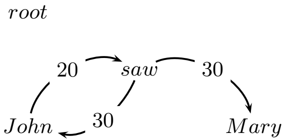


- Example for *John saw Mary*
- Build a graph:
  - Nodes are tokens (and the root token)
  - A weighted directed edge between any two vertices

$$w_{i,j} = \max_{1 \leq k \leq K} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, i, j, k)$$

Chu-Liu-Edmonds, example

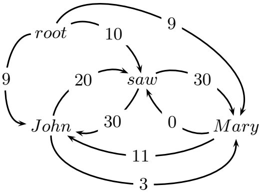
- Step 1: for each word, find highest-scoring incoming edge

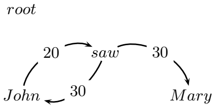


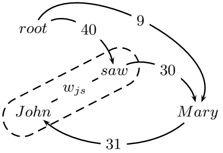
- If we get a tree, we have found the MST
- If not, there has to be a cycle

Chu-Liu-Edmonds, example

- Step 2: identify cycle and *contract* it into a new node c







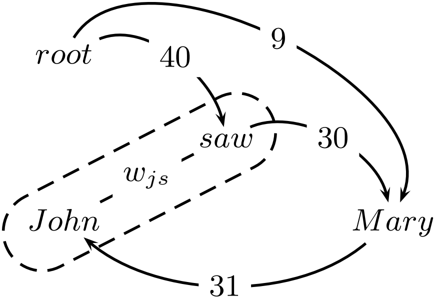
- Weight of edges between c and other nodes i:
  - $c \to i$: max weight of any node in c to i
  - $i \to c$: max weight of tree with root i that spans c

    root $\to$ *saw* $\to$ *John* : 40

    root $\to$ *John* $\to$ *saw* : 29

Chu-Liu-Edmonds

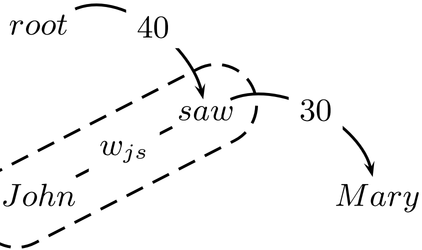
- Theorem (Leonidas 2003): the weight of the MST on the contracted graph is equal to the weight of the MST in the original graph



- Recursively call the algorithm on the new graph

Chu-Liu-Edmonds

- After one recursive call we get



- It is a tree! (if not, contract and recurse)
- The original MST can be reconstructued by undoing the contraction operations (see (McDonald et al 2005) for details)
- Cost: $O(n^3)$ (naive), $O(n^2)$ (improved)

Outline

1 Trees and Grammars

2 Constituency Parsing
- CKY Algorithm
- Earley Algorithm

3 Dependency Parsing
- Dependency Trees
- Arc-factored Dependency Parsing
- Parsing Projective Structures
- Parsing non-Projective Structures
- Transition-Based parsers

Transition-Based parsers

Trees and Grammars
Constituency Parsing
Dependency Parsing
Transition-Based parsers

- Inspired on shift-reduce parsers.
- The parser has a current state or configuration consisting of a stack (of tokens processed and tree built so far) and a buffer (tokens remaining).
- At each step, a transition is chosen to alter the configuration and move.
- Parsing stops when a final configuration is reached
- No backtracking, cost is $\mathcal{O}(n)$

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN

Trees and Grammars

Constituency Parsing

Dependency Parsing

Transition-Based parsers

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift

Trees and Grammars

Constituency Parsing

Dependency Parsing

Transition-Based parsers

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	

Trees and Grammars
Constituency Parsing
Dependency Parsing
Transition-Based parsers

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift
NP VP IN	DT NN	

Trees and Grammars
Constituency Parsing
Dependency Parsing
Transition-Based parsers

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift
NP VP IN	DT NN	shift

Trees and Grammars
Constituency Parsing
Dependency Parsing
Transition-Based parsers

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift
NP VP IN	DT NN	shift
NP VP IN DT	NN	

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift
NP VP IN	DT NN	shift
NP VP IN DT	NN	shift

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift
NP VP IN	DT NN	shift
NP VP IN DT	NN	shift
NP VP IN DT NN		

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift
NP VP IN	DT NN	shift
NP VP IN DT	NN	shift
NP VP IN DT NN		reduce NP→DT NN

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift
NP VP IN	DT NN	shift
NP VP IN DT	NN	shift
NP VP IN DT NN		reduce NP→DT NN
NP VP IN NP		

Shift-Reduce Parsing Example

Trees and Grammars
Constituency Parsing
Dependency Parsing
Transition-Based parsers

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift
NP VP IN	DT NN	shift
NP VP IN DT	NN	shift
NP VP IN DT NN		reduce NP→DT NN
NP VP IN NP		reduce PP→IN NP

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift
NP VP IN	DT NN	shift
NP VP IN DT	NN	shift
NP VP IN DT NN		reduce NP→DT NN
NP VP IN NP		reduce PP→IN NP
NP VP PP		

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift
NP VP IN	DT NN	shift
NP VP IN DT	NN	shift
NP VP IN DT NN		reduce NP→DT NN
NP VP IN NP		reduce PP→IN NP
NP VP PP		reduce VP→VP PP

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift
NP VP IN	DT NN	shift
NP VP IN DT	NN	shift
NP VP IN DT NN		reduce NP→DT NN
NP VP IN NP		reduce PP→IN NP
NP VP PP		reduce VP→VP PP
NP VP		

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift
NP VP IN	DT NN	shift
NP VP IN DT	NN	shift
NP VP IN DT NN		reduce NP→DT NN
NP VP IN NP		reduce PP→IN NP
NP VP PP		reduce VP→VP PP
NP VP		reduce S→NP VP

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift
NP VP IN	DT NN	shift
NP VP IN DT	NN	shift
NP VP IN DT NN		reduce NP→DT NN
NP VP IN NP		reduce PP→IN NP
NP VP PP		reduce VP→VP PP
NP VP		reduce S→NP VP
S		

Shift-Reduce Parsing Example

The	woman	saw	the	man	with	the	telescope
DT	NN	Vt	DT	NN	IN	DT	NN

Stack	Buffer	Transition
	DT NN Vt DT NN IN DT NN	shift
DT	NN Vt DT NN IN DT NN	shift
DT NN	Vt DT NN IN DT NN	reduce NP→DT NN
NP	Vt DT NN IN DT NN	shift
NP Vt	DT NN IN DT NN	shift
NP Vt DT	NN IN DT NN	shift
NP Vt DT NN	IN DT NN	reduce NP→DT NN
NP Vt NP	IN DT NN	*reduce VP→Vt NP
NP VP	IN DT NN	shift
NP VP IN	DT NN	shift
NP VP IN DT	NN	shift
NP VP IN DT NN		reduce NP→DT NN
NP VP IN NP		reduce PP→IN NP
NP VP PP		reduce VP→VP PP
NP VP		reduce S→NP VP
S		stop

Transition-Based parsers

- Only one tree is produced: Not suitable for ambiguous grammars (common in NLP)
- We can add probabilities to select which transition is selected at each step: Similar to CKY with PCFGs, but greedy search (may be made less greedy with e.g. beam-search)
- Or better: we can add features and use ML to take the decision.

Let's see how it is applied to dependency parsing

Arc-Standard algorithm

- A configuration (S, B, A) of the parser consists of:
  - A stack S containing seen words
  - A buffer B containing not-yet seen words
  - The dependency graph A built so far (not a tree yet)
- Initial configuration: $([\,], [0 \ldots n], [\,])$
- Final configuration: $([0], [\,], A)$
- Possible transitions:
  - shift: push next word in the buffer onto the stack
  - left-arc: add an arc from $S[0]$ to $S[1]$ and remove $S[1]$ from the stack
  - right-arc: add an arc from $S[1]$ to $S[0]$ and remove $S[0]$ from the stack

Arc-Standard Transition definitions

- shift (sh)
  $(\sigma, [i|\beta], A) \Rightarrow ([\sigma|i], \beta, A)$
- left-arc (la-L)
  $([\sigma|i|j], B, A) \Rightarrow ([\sigma|j], B, A \cup \{j, i, L\})$
- right-arc (ra-L): $([\sigma|i|j], B, A) \Rightarrow ([\sigma|i], B, A \cup \{i, j, L\})$

Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	

* the woman saw the man with glasses

Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh

* the woman saw the man with glasses

Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	

Trees and Grammars

Constituency Parsing

Dependency Parsing

Transition-Based parsers

* the woman saw the man with glasses

Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh

* the woman saw the man with glasses

Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	

* the woman saw the man with glasses

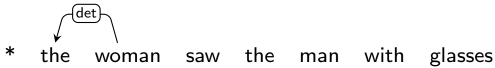
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det

* the woman saw the man with glasses

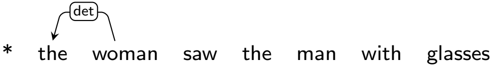
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	



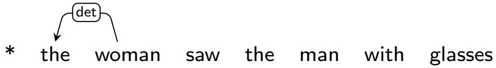
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh



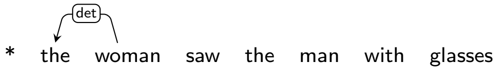
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	



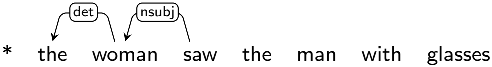
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj



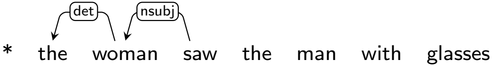
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	



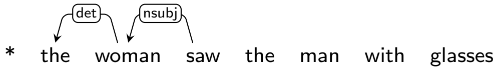
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh



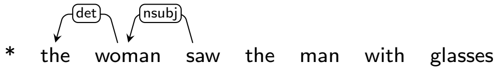
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	



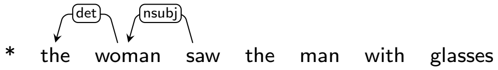
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh



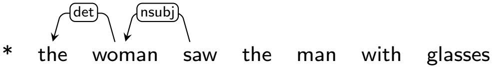
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	



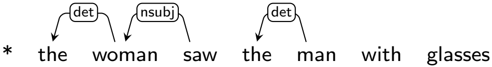
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det



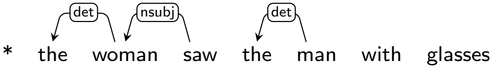
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det
* saw man	with glasses	



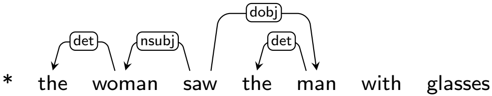
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det
* saw man	with glasses	ra-dobj



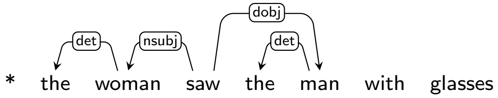
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det
* saw man	with glasses	ra-dobj
* saw	with glasses	



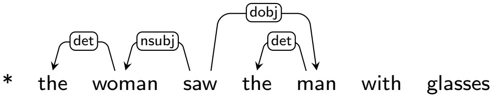
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det
* saw man	with glasses	ra-dobj
* saw	with glasses	sh



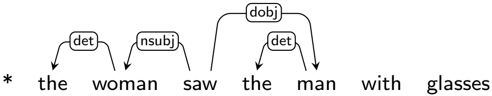
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det
* saw man	with glasses	ra-dobj
* saw	with glasses	sh
* saw with	glasses	



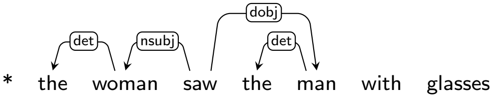
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det
* saw man	with glasses	ra-dobj
* saw	with glasses	sh
* saw with	glasses	sh



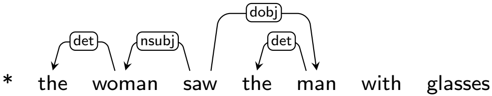
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det
* saw man	with glasses	ra-dobj
* saw	with glasses	sh
* saw with	glasses	sh
* saw with glasses		



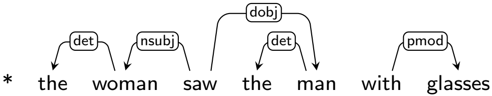
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det
* saw man	with glasses	ra-dobj
* saw	with glasses	sh
* saw with	glasses	sh
* saw with glasses		ra-pmod



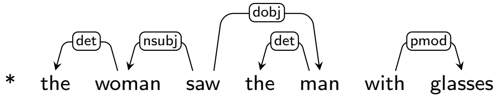
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det
* saw man	with glasses	ra-dobj
* saw	with glasses	sh
* saw with	glasses	sh
* saw with glasses		ra-pmod
* saw with		



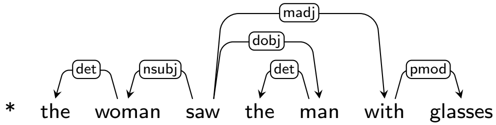
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det
* saw man	with glasses	ra-dobj
* saw	with glasses	sh
* saw with	glasses	sh
* saw with glasses		ra-pmod
* saw with		ra-madj



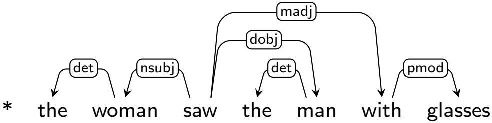
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det
* saw man	with glasses	ra-dobj
* saw	with glasses	sh
* saw with	glasses	sh
* saw with glasses		ra-pmod
* saw with		ra-madj
* saw		



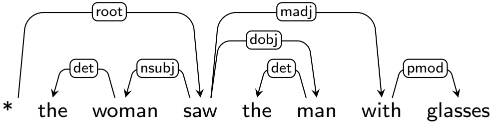
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det
* saw man	with glasses	ra-dobj
* saw	with glasses	sh
* saw with	glasses	sh
* saw with glasses		ra-pmod
* saw with		ra-madj
* saw		ra-root



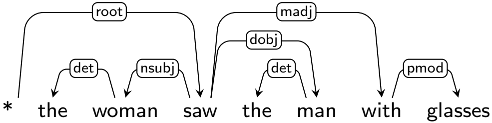
Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det
* saw man	with glasses	ra-dobj
* saw	with glasses	sh
* saw with	glasses	sh
* saw with glasses		ra-pmod
* saw with		ra-madj
* saw		ra-root
*		



Arc-Standard Example

Stack	Buffer	Transition
	* the woman saw the man with glasses	sh
* the	woman saw the man with glasses	sh
* the woman	saw the man with glasses	la-det
* woman	saw the man with glasses	sh
* woman saw	the man with glasses	la-subj
* saw	the man with glasses	sh
* saw the	man with glasses	sh
* saw the man	with glasses	la-det
* saw man	with glasses	ra-dobj
* saw	with glasses	sh
* saw with	glasses	sh
* saw with glasses		ra-pmod
* saw with		ra-madj
* saw		ra-root
*		stop



Alternative Transition Models

- Stack-stack arcs
  - Arc-standard (shift, left-arc, right-arc)
  - Non-projective (shift, swap, left-arc, right-arc)
- Stack-buffer arcs
  - Arc-eager (shift, reduce, left-arc, right-arc)
  - Arc-standard variant (shift, left-arc, right-arc)

Transition Selection

- Classifier that produces the best transition for the current configuration
- Too many possible configurations: Need to model them as feature vectors and use ML:
- Typical features:
  - word/lemma/PoS for $S[0]$, $S[1]$, $B[0]$, $B[1]$
  - morphological features (gender, number, mode, tense, etc) in $S[0]$, $B[0]$
  - number of children of $S[0]$
  - dependency labels of $S[0]$ children
  - ..etc
- We can use SVM, perceptron, MBL, DT, ... any feature-based ML classifier

Transition Selection

- Classifier that produces the best transition for the current configuration
- Too many possible configurations: Need to model them as feature vectors and use ML:
- Typical features:
  - word/lemma/PoS for $S[0]$, $S[1]$, $B[0]$, $B[1]$
  - morphological features (gender, number, mode, tense, etc) in $S[0]$, $B[0]$
  - number of children of $S[0]$
  - dependency labels of $S[0]$ children
  - ..etc
- We can use SVM, perceptron, MBL, DT, ... any feature-based ML classifier

... or we can use Deep Learning