Master in Artificial Intelligence

Advanced Human Language Technologies
Outline

1 Trees and Grammars

2 Constituency Parsing
   - CKY Algorithm
   - Earley Algorithm
A Syntactic Tree

S

NP

PRP

They

VBD

solved

VP

NP

DT

the

NN

problem

PP

IN

with

NNS

statistics
They solved the problem with statistics.
They solved the problem with statistics.
They solved the problem with statistics.
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.
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
- $\Sigma$ is a set of terminal symbols
- $R$ is a set of rules of the form $X \rightarrow Y_1Y_2 \ldots Y_n$ where $n \geq 0$, $X \in N$, $Y_i \in N \cup \Sigma$
Context Free Grammars, Example

\[ N = \{S, \text{VP}, \text{NP}, \text{PP}, \text{DT}, \text{Vi}, \text{Vt}, \text{NN}, \text{IN}\} \]

\[ S = \{S\} \]

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

\[ R = \begin{cases} 
S \rightarrow \text{NP} \ \text{VP} & \text{Vi} \rightarrow \text{sleeps} \\
\text{NP} \rightarrow \text{DT} \ \text{NN} & \text{Vt} \rightarrow \text{saw} \\
\text{NP} \rightarrow \text{NP} \ \text{PP} & \text{NN} \rightarrow \text{man} \\
\text{PP} \rightarrow \text{IN} \ \text{NP} & \text{NN} \rightarrow \text{woman} \\
\text{VP} \rightarrow \text{Vi} & \text{NN} \rightarrow \text{telescope} \\
\text{VP} \rightarrow \text{Vt} \ \text{NP} & \text{DT} \rightarrow \text{the} \\
\text{VP} \rightarrow \text{VP} \ \text{PP} & \text{IN} \rightarrow \text{with} \\
\text{IN} \rightarrow \text{in} 
\end{cases} \]

\[ ^1 S=\text{sentence, VP=verb phrase, NP=noun phrase, PP=prepositional phrase, DT=determiner, Vi=intransitive verb, Vt=transitive verb, NN=noun, IN=preposition} \]
Properties of CFGs

- A CFG defines a set of possible \textit{derivations} (i.e. unique trees)
- A sequence of terminals \( s \in \Sigma^* \) is \textit{generated} by the CFG (or \textit{recognized} by it, or \textit{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 (\textit{ambiguity}).
Ambiguity

Mary saw the man in the mountain with a telescope.

- Mary used a telescope to see a man who was in the mountain.
- Mary saw a man who was in the mountain and carried a telescope.
- Mary was in the mountain and used a telescope to see a man.
- Mary was in the mountain that has a telescope and saw a man.
- Mary saw a man who was in the mountain that has a telescope.
- Mary was in the mountain and saw a man carrying a telescope.
Ambiguity

- She announced a program aimed to make trucks and vans safer
- She used trucks and vans to announce a program aimed to promote safety
- She announced a program aimed to make trucks safer. She also announced vans
- She used trucks to announce a program aimed to promote safety. She also announced vans
- She announced a program. She did so in order to promote safety in trucks and vans
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
- $\Sigma$ 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 $n \geq 0$, $X \in N$, $Y_i \in N \cup \Sigma$
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
- $\Sigma$ is a set of terminal symbols
- $R$ is a set of rules of the form $X \rightarrow Y_1Y_2\ldots Y_n$ where $n \geq 0$, $X \in N$, $Y_i \in N \cup \Sigma$
A probabilistic 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
- $\Sigma$ 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 $n \geq 0$, $X \in N$, $Y_i \in N \cup \Sigma$
A probabilistic 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
- $\Sigma$ is a set of terminal symbols
- $R$ is a set of rules of the form $X \rightarrow Y_1Y_2\ldots Y_n$ 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 $X \rightarrow \alpha \in R$ such that, for any $X \in N$,

$$\sum (X \rightarrow \alpha) \in R q(X \rightarrow \alpha) = 1$$
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
- \( \Sigma \) 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 \( n \geq 0, X \in N, Y_i \in N \cup \Sigma \)
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
- $\Sigma$ is a set of terminal symbols
- $R$ is a set of rules of the form $X \rightarrow Y_1Y_2 \ldots Y_n$ 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 $X \rightarrow \alpha \in R$ such that, for any $X \in N$,

$$\sum_{(X \rightarrow \alpha) \in R} q(X \rightarrow \alpha) = 1$$
Context Free Grammars, Example

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

\[ ^1 \text{S=\textbf{sentence}, VP=\textbf{verb phrase}, NP=\textbf{noun phrase}, PP=\textbf{prepositional phrase}, DT=\textbf{determiner},} \]
\[ \text{Vi=\textbf{intransitive verb}, Vt=\textbf{transitive verb}, NN=\textbf{noun}, IN=\textbf{preposition}} \]
Probabilistic Context Free Grammars, Example

\[ N = \{S, VP, NP, PP, DT, Vi, Vt, NN, IN\} \]

\[ S = \{S\} \]

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

\[ R = \begin{cases} 
S \rightarrow NP \ VP & 1.0 \\
NP \rightarrow DT \ NN & 0.4 \\
NP \rightarrow NP \ PP & 0.6 \\
PP \rightarrow IN \ NP & 1.0 \\
VP \rightarrow Vi & 0.5 \\
VP \rightarrow Vt \ NP & 0.4 \\
VP \rightarrow VP \ PP & 0.1 \\
\end{cases} \]

\[ \begin{aligned} 
Vi \rightarrow \text{sleeps} & \quad 1.0 \\
Vt \rightarrow \text{saw} & \quad 1.0 \\
NN \rightarrow \text{man} & \quad 0.7 \\
NN \rightarrow \text{woman} & \quad 0.2 \\
nN \rightarrow \text{telescope} & \quad 0.1 \\
DT \rightarrow \text{the} & \quad 1.0 \\
IN \rightarrow \text{with} & \quad 0.5 \\
IN \rightarrow \text{in} & \quad 0.5 \\
\end{aligned} \]

\[ ^1 \text{S=sentence, VP=verb phrase, NP=noun phrase, PP=prepositional phrase, DT=determiner,} \]

\[ \text{Vi=intransitive verb, Vt=transitive verb, NN=noun, IN=preposition} \]
Properties of PCFGs

- The probability of a parse tree \( t \in T_G \) 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:
  \[
  \arg \max_{t \in T(s)} p(t)
  \]
Learning Treebank Grammars

- Read the grammar rules from a treebank

<table>
<thead>
<tr>
<th>Rule</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP VP .</td>
<td>1</td>
</tr>
<tr>
<td>NP → PRP</td>
<td>0.5</td>
</tr>
<tr>
<td>NP → DT NN</td>
<td>0.5</td>
</tr>
<tr>
<td>VP → VBD NP</td>
<td>1</td>
</tr>
<tr>
<td>PRP → She</td>
<td>1</td>
</tr>
</tbody>
</table>

- Set rule weights by maximum likelihood

\[ q(\alpha \rightarrow \beta) = \frac{\text{Count}(\alpha \rightarrow \beta)}{\text{Count}(\alpha)} \]

- Smoothing issues apply

- Having the appropriate CFG is critical to success
Outline

1. Trees and Grammars

2. Constituency Parsing
   - CKY Algorithm
   - Earley Algorithm
Goal of a parser:

- Find all possible trees
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
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 dynamic programming table (aka chart) to store intermediate results.
Outline

1. Trees and Grammars

2. Constituency Parsing
   - CKY Algorithm
   - Earley Algorithm
CKY Algorithm

- Bottom-up
- Requires a grammar in Chomsky Normal Form (CNF).
- Dynamic 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 $i \ldots j$:
  \[ \pi(i, j, X) \]
- 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

- \[ \forall i = 1 \ldots n, \forall j = (i + 1) \ldots n, \forall X \in N \]
  \[ \pi(i, j, X) = \max_{X \rightarrow YZ \in R} \frac{q(X \rightarrow YZ)}{k : i \leq k < j} \times \pi(i, k, Y) \times \pi(k + 1, j, Z) \]

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

Output:

- Return \( \pi(1, n, S) \) and recover backpath trough \( \psi(1, n, S) \)
CKY Algorithm - Example

\[ N = \{ S, \text{VP}, \text{NP}, \text{PP}, \text{DT}, \text{Vi}, \text{Vt}, \text{NN}, \text{IN}\} \]

\[ S = \{ S \} \]

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

\[ R = \begin{cases} 
\text{S} \rightarrow \text{NP} \text{ VP} & 1.0 \\
\text{NP} \rightarrow \text{DT} \text{ NN} & 0.4 \\
\text{NP} \rightarrow \text{NP} \text{ PP} & 0.6 \\
\text{PP} \rightarrow \text{IN} \text{ NP} & 1.0 \\
\text{VP} \rightarrow \text{Vi} & 0.5 \\
\text{VP} \rightarrow \text{Vt} \text{ NP} & 0.4 \\
\text{VP} \rightarrow \text{VP} \text{ PP} & 0.1 \\
\end{cases} \]

\[ \text{Vi} \rightarrow \text{sleeps} & 1.0 \\
\text{Vt} \rightarrow \text{saw} & 1.0 \\
\text{NN} \rightarrow \text{man} & 0.7 \\
\text{NN} \rightarrow \text{woman} & 0.2 \\
\text{NN} \rightarrow \text{telescope} & 0.1 \\
\text{DT} \rightarrow \text{the} & 1.0 \\
\text{IN} \rightarrow \text{with} & 0.5 \\
\text{IN} \rightarrow \text{in} & 0.5 \]

\[ 1 \text{S=sentence, VP=verb phrase, NP=noun phrase, PP=prepositional phrase, DT=determiner,} \]

\[ \text{Vi=intransitive verb, Vt=transitive verb, NN=noun, IN=preposition} \]
CKY Algorithm - Example - CNF

\[ N = \{ S, \text{VP}, \text{NP}, \text{PP}, \text{DT}, \text{Vi}, \text{Vt}, \text{NN}, \text{IN}\} \]
\[ S = \{ S \} \]
\[ \Sigma = \{ \text{sleeps}, \text{saw}, \text{man}, \text{woman}, \text{telescope}, \text{the}, \text{with}, \text{in}\} \]

\[ R = \begin{cases} 
S \rightarrow \text{NP} \text{ VP} & 1.0 \\
\text{NP} \rightarrow \text{DT} \text{ NN} & 0.4 \\
\text{NP} \rightarrow \text{NP} \text{ PP} & 0.6 \\
\text{PP} \rightarrow \text{IN} \text{ NP} & 1.0 \\
\text{VP} \rightarrow \text{Vi} & 0.5 \\
\text{VP} \rightarrow \text{Vt} \text{ NP} & 0.4 \\
\text{VP} \rightarrow \text{VP} \text{ PP} & 0.1 
\end{cases} \]

1\text{S}=\text{sentence, VP}=\text{verb phrase, NP}=\text{noun phrase, PP}=\text{prepositional phrase, DT}=\text{determiner, Vi}=\text{intransitive verb, Vt}=\text{transitive verb, NN}=\text{noun, IN}=\text{preposition}
CKY Algorithm - Example - CNF

\[ N = \{S, \text{VP}, \text{NP}, \text{PP}, \text{DT}, \text{Vi}, \text{Vt}, \text{NN}, \text{IN}\} \]

\[ S = \{S\} \]

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

\[ R = \begin{cases} 
S \rightarrow \text{NP} \text{ VP} & 0.5 \quad \text{Vi} \rightarrow \text{sleeps} & 1.0 \\
S \rightarrow \text{NP} \text{ Vi} & 0.5 \quad \text{Vt} \rightarrow \text{saw} & 1.0 \\
\text{NP} \rightarrow \text{DT} \text{ NN} & 0.4 \quad \text{NN} \rightarrow \text{man} & 0.7 \\
\text{NP} \rightarrow \text{NP} \text{ PP} & 0.6 \quad \text{NN} \rightarrow \text{woman} & 0.2 \\
\text{PP} \rightarrow \text{IN} \text{ NP} & 1.0 \quad \text{NN} \rightarrow \text{telescope} & 0.1 \\
\text{VP} \rightarrow \text{Vt} \text{ NP} & 0.4 \quad \text{DT} \rightarrow \text{the} & 1.0 \\
\text{VP} \rightarrow \text{VP} \text{ PP} & 0.1 \quad \text{IN} \rightarrow \text{with} & 0.5 \\
\text{VP} \rightarrow \text{Vi} \text{ PP} & 0.5 \quad \text{IN} \rightarrow \text{in} & 0.5 
\end{cases} \]

\[ ^1S=\text{sentence, VP=verb phrase, NP=noun phrase, PP=prepositional phrase, DT=determiner, Vi=intransitive verb, Vt=transitive verb, NN=noun, IN=preposition} \]
CKY Algorithm - Example

<table>
<thead>
<tr>
<th>DT 1.0</th>
<th>NN 0.2</th>
<th>Vt 1.0</th>
<th>DT 1.0</th>
<th>NN 0.7</th>
<th>IN 0.5</th>
<th>DT 1.0</th>
<th>NN 0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>The</td>
<td>woman</td>
<td>saw</td>
<td>the</td>
<td>man</td>
<td>with</td>
<td>the</td>
<td>telescope</td>
</tr>
<tr>
<td>11</td>
<td>22</td>
<td>33</td>
<td>44</td>
<td>55</td>
<td>66</td>
<td>77</td>
<td>88</td>
</tr>
</tbody>
</table>
### CKY Algorithm - Example

<table>
<thead>
<tr>
<th>NP → DT₁¹ NN₂²</th>
<th>0.4 * 1.0 * 0.2 = 0.08</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>23</td>
</tr>
<tr>
<td>NP → DT₄⁴ NN₅⁵</td>
<td>0.4 * 1.0 * 0.7 = 0.28</td>
</tr>
<tr>
<td>45</td>
<td>66</td>
</tr>
<tr>
<td>NP → DT₄⁴ NN₅⁵</td>
<td>0.4 * 1.0 * 0.1 = 0.04</td>
</tr>
<tr>
<td>78</td>
<td>88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DT 1.0</th>
<th>NN 0.2</th>
<th>Vt 1.0</th>
<th>DT 1.0</th>
<th>NN 0.7</th>
<th>IN 0.5</th>
<th>DT 1.0</th>
<th>NN 0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>The</td>
<td>woman</td>
<td>saw</td>
<td>the</td>
<td>man</td>
<td>with</td>
<td>the</td>
<td>telescope</td>
</tr>
<tr>
<td>11</td>
<td>22</td>
<td>33</td>
<td>44</td>
<td>55</td>
<td>66</td>
<td>77</td>
<td>88</td>
</tr>
</tbody>
</table>
### CKY Algorithm - Example

<table>
<thead>
<tr>
<th></th>
<th>11</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>33</th>
<th>34</th>
<th>44</th>
<th>45</th>
<th>46</th>
<th>55</th>
<th>56</th>
<th>57</th>
<th>66</th>
<th>67</th>
<th>77</th>
<th>88</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>DT</td>
<td>NN</td>
<td>VP</td>
<td>NP</td>
<td>NP</td>
<td>NP</td>
<td>NP</td>
<td>NP</td>
<td>NP</td>
<td>NP</td>
<td>NP</td>
<td>PP</td>
<td>NP</td>
<td>NP</td>
<td>NP</td>
<td>NP</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.2</td>
<td>1.0</td>
<td>0.4</td>
<td>1.0</td>
<td>0.7</td>
<td>0.6</td>
<td>0.8</td>
<td>0.9</td>
<td>1.0</td>
<td>1.0</td>
<td>0.5</td>
<td>0.4</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>The</td>
<td>woman</td>
<td>saw</td>
<td>DT</td>
<td>the</td>
<td>man</td>
<td>with</td>
<td>IN</td>
<td>DT</td>
<td>NN</td>
<td>NN</td>
<td>NN</td>
<td>NN</td>
<td>NN</td>
<td>NN</td>
<td>NN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11</td>
<td>22</td>
<td>33</td>
<td>44</td>
<td>55</td>
<td>66</td>
<td>77</td>
<td>88</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| CKY Algorithm - Example

- **Trees and Grammars**
- **Constituency Parsing**
- **CKY Algorithm**
CKY Algorithm - Example

Trees and Grammars
Constituency Parsing
CKY Algorithm
CKY Algorithm - Example

CKY Algorithm - Example
CKY Algorithm - Example
CKY Algorithm - Example
CKY Algorithm - Example
CKY Algorithm - Example

```
S → NP_{11} VP_{35} 0.5*0.08*0.112 =0.00448
16

14

13

NP → DT_{11} NN_{22} 0.4*1.0*0.2=0.08
12

VP → Vt_{23} NP_{45} 0.4*1.0*0.28=0.112
35

NP → DT_{44} NN_{55} 0.4*1.0*0.7=0.28
45

DT 1.0
The
11

NN 0.2
woman
22

Vt 1.0
saw
33

DT 1.0
the
44

NP → VP_{35} PP_{66} 0.1*0.112*0.02=0.000224
38

VP → Vt_{33} NP_{48} 0.4*1.0*0.00336=0.001344
48

NP → NP_{48} PP_{68} 0.6*0.28*0.02 =0.00336
48

PP → IN_{68} NP_{78} 1.0*0.5*0.04=0.02
68

NP → DT_{44} NN_{55} 0.4*1.0*0.1=0.04
78

NN 0.1
telescope
85
```
Outline

1. Trees and Grammars

2. Constituency Parsing
   - CKY Algorithm
   - Earley Algorithm
Earley Algorithm

- Top-down
- Can deal with any CFG (even left-recursive)
- Dynamic 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:
  \[ [i, j, X \rightarrow \alpha \bullet \beta] \]
- Backpath to recover which rules produced the complete tree:
  \[ \psi(i, j, X) \]

The goal is:
- Find if it is possible to reach \([1, n, S \rightarrow \alpha\bullet]\)
- Recover \(\psi(0, n, S')\) if it is
- Probabilistic versions exist, though not as straightforward as in CKY
Earley Algorithm

Parsing state examples:

\[ [0, 0, S \rightarrow \bullet \ NP \ VP] \quad \text{A NP is expected at the beginning of the sentence} \]

\[ [1, 2, NP \rightarrow DT \bullet NN] \quad \text{A NP has been partially matched (DT was found between positions 1 and 2)} \]

\[ [0, 3, VP \rightarrow V NP \bullet] \quad \text{A VP has been completed between positions 0 and 3} \]
Earley Algorithm

def Earley(words,grammar):
    chart = [ [ ] for i in range(len(words)+1) ]
    chart[0].append([0,0,\gamma \rightarrow \bullet 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 \rightarrow \alpha \bullet B \beta]):
    if B in words[j].PoS(): chart[j+1].append([j,j+1,B \rightarrow \text{word}[j] \bullet])

def Predict([i,j,A \rightarrow \alpha \bullet B \beta]):
    for B \rightarrow \gamma in grammar: chart[j].append([j,j,B \rightarrow \bullet \gamma])

def Complete([k,j,B \rightarrow \gamma \bullet]):
    for [i,k,A \rightarrow \alpha \bullet B \beta] in chart[k]: chart[j].append([i,j,A \rightarrow \alpha B \bullet \beta])
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

The woman saw the man with the telescope
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

\[ \gamma \rightarrow \bullet S \]

\[ \begin{array}{c}
\text{chart[0]}
\end{array} \]

\[ \begin{array}{c}
\text{chart[1]}
\end{array} \]

\[ \begin{array}{c}
\text{chart[2]}
\end{array} \]

\[ \begin{array}{c}
\text{chart[3]}
\end{array} \]

\[ \begin{array}{c}
\text{chart[4]}
\end{array} \]

\[ \begin{array}{c}
\text{chart[5]}
\end{array} \]

\[ \begin{array}{c}
\text{chart[6]}
\end{array} \]

\[ \begin{array}{c}
\text{chart[7]}
\end{array} \]

\[ \begin{array}{c}
\text{chart[8]}
\end{array} \]

0 1 2 3 4 5 6 7 8

| the | woman | saw | the | man | with | the | telescope |
Earley Algorithm

- Trees and Grammars
- Constituency Parsing
  - Earley Algorithm

### Earley Algorithm

#### Chart Representation

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>woman</td>
<td>saw</td>
<td>the</td>
<td>man</td>
<td>with</td>
<td>the</td>
<td>telescope</td>
<td></td>
</tr>
</tbody>
</table>
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

chart[0]
[0,0]
\gamma \rightarrow \bullet S
S \rightarrow \bullet NP VP

chart[1]

chart[2]

chart[3]

chart[4]

chart[5]

chart[6]

chart[7]

chart[8]

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

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

The sentences "the woman saw the man with the telescope" are parsed using the Earley Algorithm. The chart below shows the parsing process:

- **Chart 0**: Start state
- **Chart 1**: Predict: DT → the
- **Chart 2**: Predict: NP → DT NN
- **Chart 3**: Predict: VP → Vt NP
- **Chart 4**: Predict: NP → DT NN
- **Chart 5**: Predict: PP → IN NP
- **Chart 6**: Predict: NP → DT NN
- **Chart 7**: Complete scan: S → NP VP
- **Chart 8**: Complete scan: VP → VP PP

The grammar rules used are:

- S → •S
- S → •NP VP
- NP → •DT NN
- NP → •NP PP

The parsing process follows the Earley Algorithm's three steps: Prediction, Completion, and Scan.
Earley Algorithm
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

\[
\begin{align*}
\gamma & \rightarrow \bullet S \\
S & \rightarrow \bullet NP \ VP \\
NP & \rightarrow \bullet DT \ NN \\
NP & \rightarrow \bullet NP \ PP \\
\end{align*}
\]

\[
\begin{align*}
\text{chart[0]} & \rightarrow \bullet \text{the} \\
\text{chart[1]} & \rightarrow \bullet \text{woman} \\
\text{chart[2]} & \rightarrow \bullet \text{saw} \\
\text{chart[3]} & \rightarrow \bullet \text{the} \\
\text{chart[4]} & \rightarrow \bullet \text{man} \\
\text{chart[5]} & \rightarrow \bullet \text{with} \\
\text{chart[6]} & \rightarrow \bullet \text{the} \\
\text{chart[7]} & \rightarrow \bullet \text{telescope} \\
\end{align*}
\]

\[
\begin{align*}
\text{0} & \quad \text{1} & \quad \text{2} & \quad \text{3} & \quad \text{4} & \quad \text{5} & \quad \text{6} & \quad \text{7} & \quad \text{8} \\
\text{the} & \quad \text{woman} & \quad \text{saw} & \quad \text{the} & \quad \text{man} & \quad \text{with} & \quad \text{the} & \quad \text{telescope} \\
\end{align*}
\]
**Earley Algorithm**

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

Trees and Grammars
Constituency Parsing
Earley Algorithm
Earley Algorithm
Earley Algorithm

Trees and Grammars
Constituency Parsing

Earley Algorithm

Chart

0 1 2 3 4 5 6 7 8

the woman saw the man with the telescope

\[
\begin{align*}
&\gamma \rightarrow S \\
&S \rightarrow NP \ VP \\
&NP \rightarrow DT \ NN \\
&NP \rightarrow NP \ PP
\end{align*}
\]
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

\[
S \rightarrow \text{NP} \ \text{VP}
\]

\[
\gamma \rightarrow S
\]

\[
\text{NP} \rightarrow \text{DT} \ \text{NN}
\]

\[
\text{NP} \rightarrow \text{NP} \ \text{PP}
\]

\[
\text{VP} \rightarrow \text{NP} \ \text{VP}
\]

\[
\text{VP} \rightarrow \text{VP} \ \text{PP}
\]

\[
\text{VP} \rightarrow \text{Vt} \ \text{NP}
\]

\[
\text{PP} \rightarrow \text{IN} \ \text{NP}
\]

\[
\text{NP} \rightarrow \text{DT} \ \text{NN}
\]

\[
\text{NP} \rightarrow \text{NP} \ \text{PP}
\]

\[
\text{NP} \rightarrow \text{DT} \ \text{NN}
\]

\[
\text{NP} \rightarrow \text{NP} \ \text{PP}
\]

\[
\text{PP} \rightarrow \text{IN} \ \text{NP}
\]

\[
\text{NP} \rightarrow \text{DT} \ \text{NN}
\]

\[
\text{NP} \rightarrow \text{NP} \ \text{PP}
\]

\[
\text{PP} \rightarrow \text{IN} \ \text{NP}
\]

\[
\text{NP} \rightarrow \text{DT} \ \text{NN}
\]

\[
\text{NP} \rightarrow \text{NP} \ \text{PP}
\]

\[
\text{PP} \rightarrow \text{IN} \ \text{NP}
\]

\[
\text{NP} \rightarrow \text{DT} \ \text{NN}
\]

\[
\text{NP} \rightarrow \text{NP} \ \text{PP}
\]

\[
\text{PP} \rightarrow \text{IN} \ \text{NP}
\]

The woman saw the man with the telescope.
Trees and Grammars
Constituency Parsing
Earley Algorithm

**Earley Algorithm**

```
the woman saw the man with the telescope
```

**Chart Representation**

```
<table>
<thead>
<tr>
<th>chart[0]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0,0]</td>
</tr>
<tr>
<td>γ → •S</td>
</tr>
<tr>
<td>S → •NP VP</td>
</tr>
<tr>
<td>NP → •DT NN</td>
</tr>
<tr>
<td>NP → •NP PP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chart[1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0,1]</td>
</tr>
<tr>
<td>DT → the •</td>
</tr>
<tr>
<td>NP → DT • NN</td>
</tr>
</tbody>
</table>

<p>| scan |</p>
<table>
<thead>
<tr>
<th>chart[2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,0]</td>
</tr>
<tr>
<td>DT → the •</td>
</tr>
<tr>
<td>NP → DT • NN</td>
</tr>
<tr>
<td>chart[3]</td>
</tr>
<tr>
<td>[2,0]</td>
</tr>
<tr>
<td>VP → Vt NP •</td>
</tr>
<tr>
<td>chart[4]</td>
</tr>
<tr>
<td>[3,0]</td>
</tr>
<tr>
<td>NP → DT • NN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chart[5]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[4,0]</td>
</tr>
<tr>
<td>NP → DT • NN</td>
</tr>
<tr>
<td>chart[6]</td>
</tr>
<tr>
<td>[5,0]</td>
</tr>
<tr>
<td>PP → IN NP •</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chart[7]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[6,0]</td>
</tr>
<tr>
<td>NP → DT • NN</td>
</tr>
<tr>
<td>chart[8]</td>
</tr>
<tr>
<td>[7,0]</td>
</tr>
<tr>
<td>PP → IN NP •</td>
</tr>
</tbody>
</table>
```

---

**Trees and Grammars**

**Constituency Parsing**

**Earley Algorithm**
The Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

\[ S \rightarrow NP \ VP \]
\[ NP \rightarrow DT \ NN \]
\[ NP \rightarrow NP \ PP \]
\[ DP \rightarrow DT \ NN \]
\[ DP \rightarrow DP \ PP \]
\[ VP \rightarrow Vt \ NP \]
\[ VP \rightarrow VP \ PP \]
\[ PP \rightarrow IN \ NP \]

the woman saw the man with the telescope
Advanced Human Language Technologies

Trees and Grammars

Constituency Parsing

Earley Algorithm

Earley Algorithm

Chart[0]

\[
\begin{align*}
\gamma &\rightarrow S \\
S &\rightarrow NP \ VP \\
NP &\rightarrow DT \ NN \\
NP &\rightarrow NP \ PP \\
\end{align*}
\]

Chart[1]

\[
\begin{align*}
[0,1] \ DT &\rightarrow \text{the} \\
NP &\rightarrow DT \ NN \\
[1,2] \ NN &\rightarrow \text{woman} \\
\end{align*}
\]

Chart[2]

\[
\begin{align*}
NP &\rightarrow DT \ NN \\
\end{align*}
\]

Chart[3]

\[
\begin{align*}
\text{complete} \\
\end{align*}
\]

Chart[4]

\[
\begin{align*}
\text{chart[3]} \\
\end{align*}
\]

Chart[5]

\[
\begin{align*}
\text{complete} \\
\end{align*}
\]

Chart[6]

\[
\begin{align*}
\text{chart[5]} \\
\end{align*}
\]

Chart[7]

\[
\begin{align*}
\text{chart[6]} \\
\end{align*}
\]

Chart[8]

\[
\begin{align*}
\text{chart[7]} \\
\end{align*}
\]

The woman saw the man with the telescope.
Earley Algorithm

```
[chart[0]]
\gamma \rightarrow S
S \rightarrow NP \ VP
NP \rightarrow DT \ NN
NP \rightarrow NP PP

[chart[1]]
NP \rightarrow DT \ NN

[chart[2]]
NP \rightarrow DT \ NN

[chart[3]]
NP \rightarrow DT \ NN

[chart[4]]
NP \rightarrow DT \ NN

[chart[5]]
NP \rightarrow DT \ NN

[chart[6]]
NP \rightarrow DT \ NN

[chart[7]]
NP \rightarrow DT \ NN

[chart[8]]
NP \rightarrow DT \ NN
```

0 1 2 3 4 5 6 7 8

| the | woman | saw | the | man | with | the | telescope |
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm
Earley Algorithm

```
0 1 2 3 4 5 6 7 8

chart[0]
- γ → S
- S → NP VP
- NP → DT NN
- NP → NP PP

chart[1]
- DT → the
- NP → DT NN

chart[2]
- NP → DT NN
- S → NP VP
- NP → NP PP

chart[3]

chart[4]

chart[5]

chart[6]

chart[7]

chart[8]
```

```
| the | woman | saw | the | man | with | the | telescope |
```
### Earley Algorithm

**Chart 0:**
- $\gamma \rightarrow S$
- $S \rightarrow NP \cdot VP$
- $NP \rightarrow DT \cdot NN$
- $NP \rightarrow NP \cdot PP$

**Chart 1:**
- $\rightarrow \cdot S$
- $S \rightarrow \cdot NP \cdot VP$
- $NP \rightarrow \cdot DT \cdot NN$
- $NP \rightarrow \cdot NP \cdot PP$

**Chart 2:**
- $NP \rightarrow DT \cdot NN$
- $S \rightarrow NP \cdot VP$
- $NP \rightarrow NP \cdot PP$

**Chart 3:**
- $VP \rightarrow \cdot Vi$
- $VP \rightarrow \cdot Vt \cdot NP$
- $VP \rightarrow \cdot VP \cdot PP$

**Chart 4:**
- $\rightarrow \cdot S$
- $S \rightarrow NP \cdot VP$
- $NP \rightarrow DT \cdot NN$
- $NP \rightarrow NP \cdot PP$

**Chart 5:**
- $NP \rightarrow DT \cdot NN$
- $S \rightarrow NP \cdot VP$
- $NP \rightarrow NP \cdot PP$

**Chart 6:**
- $VP \rightarrow \cdot Vi$
- $VP \rightarrow \cdot Vt \cdot NP$
- $VP \rightarrow \cdot VP \cdot PP$

**Chart 7:**
- $\rightarrow \cdot S$
- $S \rightarrow NP \cdot VP$
- $NP \rightarrow DT \cdot NN$
- $NP \rightarrow NP \cdot PP$

**Chart 8:**
- $NP \rightarrow DT \cdot NN$
- $S \rightarrow NP \cdot VP$
- $NP \rightarrow NP \cdot PP$

---

**Sentence:**

```
the woman saw the man with the telescope
```
Earley Algorithm

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

```
[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

[1,2] NN → woman•
```

```
[2,2] VP →• Vi
VP →• Vt NP
VP →• VP PP
```

```
chart[0]
chart[1]
chart[2]
chart[3]
chart[4]
chart[5]
chart[6]
chart[7]
chart[8]
```
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[0,0]</td>
<td>[0,1]</td>
<td>[0,2]</td>
<td>[1,2]</td>
<td>[2,2]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>γ → S</td>
<td>DT → the</td>
<td>NP → DT NN</td>
<td>NP → DT NN</td>
<td>VP → Vi</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S → NP VP</td>
<td>NN → woman</td>
<td>NP → NP • VP</td>
<td>NP → NP • PP</td>
<td>VP → Vt NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP → NP PP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0 1 2 3 4 5 6 7 8

the woman saw the man with the telescope
Earley Algorithm

The sentence to be parsed is: "the woman saw the man with the telescope".

The Earley algorithm chart is shown with the following grammar rules:

\[
\begin{align*}
\gamma & \rightarrow S \\
S & \rightarrow NP \ VP \\
NP & \rightarrow DT \ NN \\
NP & \rightarrow NP \ PP \\
VP & \rightarrow Vt \ NP \\
VP & \rightarrow VP \ PP \\
PP & \rightarrow IN \ NP
\end{align*}
\]

The chart is divided into 9 sections, labeled chart[0] to chart[8]. The chart shows the Earley algorithm's predictive and scanning phases as it processes the sentence, with transitions and states marked in each cell.
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

The woman saw the man with the telescope.
Earley Algorithm

The woman saw the man with the telescope.
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

the woman saw the man with the telescope
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

the woman saw the man with the telescope
Earley Algorithm

```
γ → •
S → •S
NP → •NP VP
NP → •NP PP

NP → DT NN •
S → NP • VP
NP → NP • VP

VP → •Vi
VP → •Vt NP
VP → •VP PP
PP → •IN NP

chart[0]

chart[1]

chart[2]

chart[3]

chart[4]

chart[5]

chart[6]

chart[7]

chart[8]

0 1 2 3 4 5 6 7 8

| the | woman | saw | the | man | with | the | telescope |
```
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

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

chart[0]
[S → • S]
[S → • NP VP]
[NP → • DT NN]
[NP → • NP PP]

chart[1]
[DT → • the]
[NP → • DT • NN]

chart[2]
[NP → • DT NN]
[S → • NP • VP]
[NP → • NP • PP]

chart[3]

chart[4]

chart[5]

chart[6]

chart[7]

chart[8]

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

The diagram shows the Earley Algorithm chart for the sentence: "the woman saw the man with the telescope." The chart is a visual representation of the parsing process, with each cell in the chart representing a possible parse or a prediction that can be made at that point in the sentence. The chart is divided into sections, with each section illustrating different parts of the sentence and the corresponding rules applied in Earley's Earley Algorithm.
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm
the woman saw the man with the telescope
The Earley Algorithm is a parsing algorithm that can parse sentences into constituent structures. It works by building a chart that tracks the possible parses of the sentence.

Here is an example of how the Earley Algorithm works on the sentence: "the woman saw the man with the telescope." The chart shows the possible parses at each step:

- **chart[0]**: The sentence is empty, so the only rule available is the start rule $\gamma \rightarrow S$.
- **chart[1]**: The first word is "the", which can be parsed as a DT (determiner). The next words are "woman", which can be parsed as a NN (noun).
- **chart[2]**: The next word is "saw", which can be parsed as a Vt (transitive verb). The next words are "the", which can be parsed as a DT. The next words are "man", which can be parsed as a NN.
- **chart[3]**: The next word is "with", which can be parsed as an IN (preposition).
- **chart[4]**: The next word is "the", which can be parsed as a DT. The next words are "telescope", which can be parsed as a NN.

The Earley Algorithm uses a three-state method: complete, predict, and scan. Each state represents a different phase of the parsing process.
The Earley Algorithm is a parsing algorithm that can be used for both context-free grammars and context-free languages. It is a bottom-up parser that can be used to parse sentences into a parse tree. The Earley Algorithm works by predicting and scanning the input sentence to fill in the chart with possible parses. The chart is a table that shows all possible parses of the input sentence. The algorithm uses three types of predictions: complete, partial, and impossible. Complete predictions are those that can be completely filled in, partial predictions are those that can only be partially filled in, and impossible predictions are those that cannot be filled in. The algorithm works by moving from left to right in the input sentence and from top to bottom in the chart. For each input symbol, the algorithm predicts all possible new symbols that can be added to the chart and then scans the chart to see if any new predictions can be made. The algorithm stops when all predictions have been made or when a complete parse is found. The Earley Algorithm is a powerful and efficient parsing algorithm that can be used for a wide variety of natural language processing tasks.
Advanced Human Language Technologies

Trees and Grammars

Constituency Parsing

Earley Algorithm

Earley Algorithm

complete predict scan

chart[8]

chart[7]

chart[6]

chart[5]

chart[4]

chart[3]

chart[2]

chart[1]

chart[0]

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>woman</td>
<td>saw</td>
<td>the</td>
<td>man</td>
<td>with</td>
<td>the</td>
<td>telescope</td>
<td></td>
</tr>
</tbody>
</table>
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

the woman saw the man with the telescope
the woman saw the man with the telescope
### Earley Algorithm

A chart representing the Earley Algorithm for parsing the sentence "the woman saw the man with the telescope." The chart is structured with rows and columns, where each cell contains a production rule from the grammar and a scheduler state. The chart is organized as follows:

- **chart[0]**
  - Production rules: `S → NP VP`, `NP → DT NN`, `NP → NP PP`.
  - Scheduler states: `0 1 2 3 4 5 6 7 8`.

- **chart[1]**
  - Production rules: `DT → the`, `NP → DT NN`, `NN → woman`.

- **chart[2]**
  - Production rules: `NP → DT NN`, `S → NP VP`, `NP → NP PP`.

- **chart[3]**
  - Production rules: `VP → Vi`, `VP → Vt NP`, `PP → IN NP`.

- **chart[4]**
  - Production rules: `S → NP VP`, `NP → DT NN`, `NP → NP PP`.

- **chart[5]**
  - Production rules: `VP → Vt NP`, `NP → DT NP`, `NP → NP PP`.

- **chart[6]**
  - Production rules: `NP → DT NN`, `VP → Vt NP`, `PP → IN NP`.

- **chart[7]**
  - Production rules: `NP → DT NP`, `NP → NP PP`.

- **chart[8]**
  - Production rules: `NP → DT NN`, `VP → Vt NP`, `PP → IN NP`.

The chart shows the progressive parsing process, with each cell indicating the current state of the parser, the non-terminal symbols (like NP, VP, S), and the terminals (like the, woman, saw, man, with, telescope). The algorithm predicts a complete parse as it iterates through the chart, identifying constituents and grammatical structures.
Earley Algorithm

```
[0.0]
γ → S
S → NP VP
NP → DT NN
NP → NP PP

[0.1]
DT → the
NP → DT NN

[1.1]
NN → woman
VP → Vt NP

[2.1]
Vt → saw
VP → VP PP

[3.1]
NP → DT NN
NP → NP PP

[4.1]
IN → with
PP → IN NP

[5.1]
DT → the
NP → DT NN

[6.1]
PP → IN NP

[7.1]
NP → DT NN
NP → NP PP

[8.1]
PP → IN NP

[0.2]
NP → DT NN
S → NP VP
NP → NP PP

[1.2]
NN → woman
VP → Vt NP

[2.2]
Vt → saw
VP → VP PP

[3.2]
NP → DT NN
NP → NP PP

[4.2]
IN → with
PP → IN NP

[5.2]
NP → DT NN
NP → NP PP

[6.2]
PP → IN NP

[7.2]
NP → DT NN
NP → NP PP

[8.2]
PP → IN NP

chart[0]
chart[1]
chart[2]
chart[3]
chart[4]
chart[5]
chart[6]
chart[7]
chart[8]
```

```
| the | woman | saw | the | man | with | the | telescope |
```
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

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

Trees and Grammars
Constituency Parsing
Earley Algorithm
Earley Algorithm

Complete scan

chart[0]

chart[1]

chart[2]

chart[3]

chart[4]

chart[5]

chart[6]

chart[7]

chart[8]

\( \gamma \rightarrow \) •

NP → DT NN
S → NP • VP
NP → NP • PP

[0, 1]

DT → the
NP → DT • NN

[1, 2]

NN → woman
Vt → saw
VP → Vt • NP

[2, 3]

Vt → Vi
VP → Vt • NP

[3, 3]

NP → DT NN
NP → NP • PP
PP → IN NP

[3, 4]

DT → the
NN → man

[4, 5]

NP → DT NN
NP → NP • PP
PP → IN NP

[5, 0]

S → NP • VP

[5, 5]

NP → DT NN
NP → NP • PP
PP → IN NP

[6, 0]

NP → DT NN
NP → NP • PP
PP → IN NP

[6, 6]

NP → DT NN
NP → NP • PP
PP → IN NP

[7, 0]

NP → DT NN
NP → NP • PP
PP → IN NP

[7, 7]

NP → DT NN
NP → NP • PP
PP → IN NP

[8, 0]

NP → DT NN
NP → NP • PP
PP → IN NP

[8, 8]

NP → DT NN
NP → NP • PP
PP → IN NP

0 1 2 3 4 5 6 7 8

| the | woman | saw | the | man | with | the | telescope |
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm
the woman saw the man with the telescope
Earley Algorithm

---

The Earley Algorithm is a parsing algorithm for context-free grammars. It is a dynamic programming algorithm that constructs a complete parse tree of a sentence. The algorithm is complete in the sense that it always finds a parse tree if one exists, and it is predictive in the sense that it can handle ambiguous grammars.

---

**Chart 0:**
- $\gamma \rightarrow S$
- $S \rightarrow NP \ VP$
- $NP \rightarrow DT \ NN$
- $NP \rightarrow NP \ PP$

**Chart 1:**
- $\gamma \rightarrow S$
- $S \rightarrow NP \ VP$
- $NP \rightarrow DT \ NN$
- $VP \rightarrow Vi$
- $VP \rightarrow VT \ NP$
- $VP \rightarrow VP \ PP$
- $PP \rightarrow IN \ NP$

**Chart 2:**
- $NP \rightarrow DT \ NN$
- $NP \rightarrow NP \ PP$
- $VP \rightarrow Vi$
- $VP \rightarrow VT \ NP$
- $VP \rightarrow VP \ PP$
- $PP \rightarrow IN \ NP$

**Chart 3:**
- $NP \rightarrow DT \ NN$
- $NP \rightarrow NP \ PP$
- $VP \rightarrow Vi$
- $VP \rightarrow VT \ NP$
- $VP \rightarrow VP \ PP$
- $PP \rightarrow IN \ NP$

**Chart 4:**
- $S \rightarrow NP \ VP$
- $NP \rightarrow DT \ NN$
- $NP \rightarrow NP \ PP$
- $VP \rightarrow Vi$
- $VP \rightarrow VT \ NP$
- $VP \rightarrow VP \ PP$
- $PP \rightarrow IN \ NP$

**Chart 5:**
- $S \rightarrow NP \ VP$
- $NP \rightarrow DT \ NN$
- $NP \rightarrow NP \ PP$
- $VP \rightarrow Vi$
- $VP \rightarrow VT \ NP$
- $VP \rightarrow VP \ PP$
- $PP \rightarrow IN \ NP$

**Chart 6:**
- $S \rightarrow NP \ VP$
- $NP \rightarrow DT \ NN$
- $NP \rightarrow NP \ PP$
- $VP \rightarrow Vi$
- $VP \rightarrow VT \ NP$
- $VP \rightarrow VP \ PP$
- $PP \rightarrow IN \ NP$

**Chart 7:**
- $S \rightarrow NP \ VP$
- $NP \rightarrow DT \ NN$
- $NP \rightarrow NP \ PP$
- $VP \rightarrow Vi$
- $VP \rightarrow VT \ NP$
- $VP \rightarrow VP \ PP$
- $PP \rightarrow IN \ NP$

**Chart 8:**
- $S \rightarrow NP \ VP$
- $NP \rightarrow DT \ NN$
- $NP \rightarrow NP \ PP$
- $VP \rightarrow Vi$
- $VP \rightarrow VT \ NP$
- $VP \rightarrow VP \ PP$
- $PP \rightarrow IN \ NP$

---

The sentence being parsed is: "the woman saw the man with the telescope."
The Earley Algorithm
### Earley Algorithm

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0-0</td>
<td>0-1</td>
<td>0-2</td>
<td>0-3</td>
<td>0-4</td>
<td>0-5</td>
<td>0-6</td>
<td>0-7</td>
<td>0-8</td>
</tr>
<tr>
<td>$\gamma \rightarrow S$</td>
<td>NP $\rightarrow$ DT NN*</td>
<td>S $\rightarrow$ NP $\rightarrow$ VP*</td>
<td>NP $\rightarrow$ NP $\rightarrow$ PP*</td>
<td>VP $\rightarrow$ Vi</td>
<td>VP $\rightarrow$ Vt NP</td>
<td>VP $\rightarrow$ VP $\rightarrow$ PP</td>
<td>NP $\rightarrow$ DT NN*</td>
<td>PP $\rightarrow$ IN NP</td>
</tr>
<tr>
<td>S $\rightarrow$ NP $\rightarrow$ VP*</td>
<td>NP $\rightarrow$ DT NN*</td>
<td>S $\rightarrow$ NP $\rightarrow$ VP*</td>
<td>NP $\rightarrow$ NP $\rightarrow$ PP*</td>
<td>VP $\rightarrow$ Vi</td>
<td>VP $\rightarrow$ Vt NP</td>
<td>VP $\rightarrow$ VP $\rightarrow$ PP</td>
<td>NP $\rightarrow$ DT NN*</td>
<td>PP $\rightarrow$ IN NP</td>
</tr>
<tr>
<td>NP $\rightarrow$ DT NN*</td>
<td>S $\rightarrow$ NP $\rightarrow$ VP*</td>
<td>NP $\rightarrow$ NP $\rightarrow$ PP*</td>
<td>NP $\rightarrow$ NP $\rightarrow$ PP*</td>
<td>VP $\rightarrow$ Vi</td>
<td>VP $\rightarrow$ Vt NP</td>
<td>VP $\rightarrow$ VP $\rightarrow$ PP</td>
<td>NP $\rightarrow$ DT NN*</td>
<td>PP $\rightarrow$ IN NP</td>
</tr>
<tr>
<td>NP $\rightarrow$ DT NN*</td>
<td>NP $\rightarrow$ NP $\rightarrow$ PP*</td>
<td>NP $\rightarrow$ NP $\rightarrow$ PP*</td>
<td>VP $\rightarrow$ Vi</td>
<td>VP $\rightarrow$ Vt NP</td>
<td>VP $\rightarrow$ VP $\rightarrow$ PP</td>
<td>NP $\rightarrow$ DT NN*</td>
<td>PP $\rightarrow$ IN NP</td>
<td></td>
</tr>
</tbody>
</table>

**Sentence:**

the woman saw the man with the telescope
Earley Algorithm

Charts:

- **chart[0]**
  - $\gamma \rightarrow S$
  - $S \rightarrow NP \cdot VP$
  - $NP \rightarrow DT \cdot NN$
  - $NP \rightarrow NP \cdot PP$

- **chart[1]**
  - $DT \rightarrow the$
  - $NP \rightarrow DT \cdot NN$
  - $NN \rightarrow woman$
  - $NP \rightarrow DT \cdot NN$

- **chart[2]**
  - $NP \rightarrow DT \cdot NN$
  - $S \rightarrow NP \cdot VP$
  - $NP \rightarrow NP \cdot PP$

- **chart[3]**
  - $VP \rightarrow Vi$
  - $VP \rightarrow Vt \cdot NP$
  - $NP \rightarrow DT \cdot NN$
  - $VP \rightarrow VP \cdot PP$
  - $PP \rightarrow IN \cdot NP$

- **chart[4]**
  - $VP \rightarrow Vt \cdot NP$
  - $VP \rightarrow VP \cdot PP$
  - $NP \rightarrow DT \cdot NN$
  - $VP \rightarrow VP \cdot PP$
  - $PP \rightarrow IN \cdot NP$

- **chart[5]**
  - $S \rightarrow NP \cdot VP$
  - $\gamma \rightarrow S$
  - $NP \rightarrow DT \cdot NN$
  - $NP \rightarrow NP \cdot PP$

- **chart[6]**
  - $VP \rightarrow Vt \cdot NP$
  - $VP \rightarrow VP \cdot PP$
  - $NP \rightarrow DT \cdot NN$
  - $VP \rightarrow VP \cdot PP$
  - $PP \rightarrow IN \cdot NP$

- **chart[7]**
  - $VP \rightarrow Vt \cdot NP$
  - $VP \rightarrow VP \cdot PP$
  - $NP \rightarrow DT \cdot NN$
  - $VP \rightarrow VP \cdot PP$
  - $PP \rightarrow IN \cdot NP$

- **chart[8]**
  - $VP \rightarrow Vt \cdot NP$
  - $VP \rightarrow VP \cdot PP$
  - $NP \rightarrow DT \cdot NN$
  - $VP \rightarrow VP \cdot PP$
  - $PP \rightarrow IN \cdot NP$

Words:

- the
- woman
- saw
- the
- man
- with
- the
- telescope
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

chart[0]

chart[1]

chart[2]

chart[3]

chart[4]

chart[5]

chart[6]

chart[7]

chart[8]

0 1 2 3 4 5 6 7 8

the woman saw the man with the telescope
Advanced Human Language Technologies

Trees and Grammars
Constituency Parsing

Earley Algorithm

chart[0]

0 1 2 3 4 5 6 7 8

the woman saw the man with the telescope

\[
\begin{align*}
\gamma &\rightarrow \bullet S \\
S &\rightarrow \bullet NP \ VP \\
NP &\rightarrow \bullet DT \ NN \\
NP &\rightarrow \bullet NP \ PP \\
NP &\rightarrow \bullet DT \ NN \\
NP &\rightarrow \bullet NP \ PP \\
PP &\rightarrow \bullet IN NP \\
\end{align*}
\]

\[
\begin{align*}
[0.0] &\\
[0.1] & DT \rightarrow \bullet the \\
& NP \rightarrow DT \bullet NN \\
[0.2] & NP \rightarrow DT NN \\
& S \rightarrow NP \ VP \\
& NP \rightarrow NP \bullet PP \\
[1.2] & NN \rightarrow woman \\
& Vt \rightarrow saw \\
& VP \rightarrow Vt \bullet NP \\
[2.3] & DT \rightarrow \bullet the \\
& NP \rightarrow DT \bullet NN \\
[3.4] & DT \rightarrow \bullet the \\
& NP \rightarrow DT \bullet NN \\
[4.5] & NN \rightarrow man \\
& IN \rightarrow with \\
[5.6] & IN \rightarrow with \\
\end{align*}
\]

\[
\begin{align*}
[0.5] & S \rightarrow NP \ VP \\
& \gamma \rightarrow S \\
[2.5] & VP \rightarrow Vt \ VP \\
& VP \rightarrow VP \bullet PP \\
[3.5] & NP \rightarrow DT NN \\
& NP \rightarrow NP \bullet PP \\
\end{align*}
\]
Advanced Human Language Technologies

Trees and Grammars

Constituency Parsing

Earley Algorithm

The Earley Algorithm is a parsing algorithm for context-free grammars. It can be used to parse sentences and build a parse tree. The algorithm is based on Earley's chart, which is a table that represents the possible parses of a sentence. Each column in the chart corresponds to a position in the sentence, and each row represents a possible parse of the sentence up to that position.

The Earley algorithm works by scanning the sentence from left to right, and for each position in the sentence, it searches for possible parses that can start at that position. The algorithm then updates the chart to reflect the new parses found.

Here is an example of an Earley chart for the sentence "the woman saw the man with the telescope":

- **chart[0]**: Initial state with the start symbol `S`.
- **chart[1]**: Continue parsing the sentence.
- **chart[2]**: Parse the NP `NN`.
- **chart[3]**: Parse the VP `Vt`.
- **chart[4]**: Parse the NP `NN`.
- **chart[5]**: Parse the NP `NN`.
- **chart[6]**: Parse the PP `IN`.
- **chart[7]**: Parse the PP `IN`.
- **chart[8]**: Parse the PP `IN`.

The chart shows the possible parses of the sentence at each position, and the final parse tree is built by connecting the parses in the chart.

The Earley algorithm is a complete and predictive parser, meaning that it can parse any context-free sentence and build a parse tree for it. It is also a left-to-right parser, meaning that it only processes the sentence from left to right.

The Earley algorithm is widely used in natural language processing, particularly in parsing and understanding human language.
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

[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

[1,0]  NP → DT NN
[1,1]  NN → woman
      VP → Vt NP
      VP → VP • PP
[1,2]  VP → Vt NP
[1,3]  VP → VP • PP

[2,0]  VP → Vt NP
      VP → VP • PP
[2,1]  VP → Vt NP
      VP → VP • PP
[2,2]  VP → Vt NP

[3,0]  VP → Vt NP
      VP → VP • PP
      PP → • IN NP
[3,1]  VP → Vt NP
      VP → VP • PP
      PP → • IN NP
[3,2]  VP → Vt NP
      VP → VP • PP
      PP → • IN NP
[3,3]  VP → Vt NP
[3,4]  DT → the
      NP → DT NN
[3,5]  NP → DT NN
      NP → NP PP

[4,0]  DT → the
      NP → DT NN
[4,1]  NN → man
      VP → Vt NP
[4,2]  VP → Vt NP
[4,3]  VP → Vt NP
[4,4]  VP → Vt NP

[5,0]  DT → the
[5,1]  IN → with
[5,2]  PP → • IN NP
[5,3]  PP → • IN NP
[5,4]  PP → • IN NP
[5,5]  PP → • IN NP
[5,6]  IN → with

[6,0]  DT → the
[6,1]  NP → DT NN
[6,2]  NP → DT NN
[6,3]  NP → DT NN
[6,4]  NP → DT NN

[7,0]  DT → the
[7,1]  NP → DT NN
[7,2]  NP → DT NN
[7,3]  NP → DT NN

[8,0]  DT → the
[8,1]  NP → DT NN
[8,2]  NP → DT NN
[8,3]  NP → DT NN

The woman saw the man with the telescope.
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

`the woman saw the man with the telescope`
Earley Algorithm
Earley Algorithm

0 1 2 3 4 5 6 7 8

| the | woman | saw | the | man | with | the | telescope |
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

\[
\begin{align*}
\gamma &\rightarrow \epsilon \\
S &\rightarrow \text{NP VP} \\
\gamma &\rightarrow S
\end{align*}
\]

\[
\begin{align*}
\text{NP} &\rightarrow \text{DT NN} \\
\text{S} &\rightarrow \text{NP VP} \\
\text{NP} &\rightarrow \text{NP PP}
\end{align*}
\]

\[
\begin{align*}
\text{VP} &\rightarrow \text{Vt NP} \\
\text{VP} &\rightarrow \text{VP PP}
\end{align*}
\]

\[
\begin{align*}
\text{PP} &\rightarrow \text{IN NP}
\end{align*}
\]

\[
\begin{align*}
\text{DT} &\rightarrow \text{the} \\
\text{NN} &\rightarrow \text{woman} \\
\text{Vt} &\rightarrow \text{saw} \\
\text{DT} &\rightarrow \text{the} \\
\text{NN} &\rightarrow \text{man}
\end{align*}
\]

\[
\begin{align*}
\text{IN} &\rightarrow \text{with} \\
\text{DT} &\rightarrow \text{the}
\end{align*}
\]

\[
\begin{align*}
\text{the} &\quad \text{woman} &\quad \text{saw} &\quad \text{the} &\quad \text{man} &\quad \text{with} &\quad \text{the} &\quad \text{telescope}
\end{align*}
\]
Earley Algorithm
Earley Algorithm
### Earley Algorithm

---

#### Trees and Grammars
- Constituency Parsing
- Earley Algorithm

---

#### Earley Algorithm

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 1</td>
<td>0, 2</td>
<td>0, 3</td>
<td>0, 4</td>
<td>0, 5</td>
<td>0, 6</td>
<td>0, 7</td>
<td>0, 8</td>
<td></td>
</tr>
<tr>
<td>DT → the</td>
<td>-</td>
<td>NP → DT NN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP → DT NN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Grammar Rules
- **S → NP VP**: Complete<br>
- **γ → S**: Predict<br>
- **NP → DT NN**: Complete<br>
- **VP → Vt NP**: Complete<br>
- **PP → IN NP**: Complete

---

#### Input Sentence
- the woman saw the man with the telescope
Earley Algorithm

0 1 2 3 4 5 6 7 8

the woman saw the man with the telescope
### Earley Algorithm

**Trees and Grammars**

**Constituency Parsing**

**Earley Algorithm**

---

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[0,0]</td>
<td>[0,1]</td>
<td>[0,2]</td>
<td>[0,3]</td>
<td>[1,0]</td>
<td>[2,0]</td>
<td>[3,0]</td>
<td>[4,0]</td>
<td>[5,0]</td>
</tr>
<tr>
<td>γ → •S</td>
<td>DT → •the</td>
<td>NP → •DT •NN</td>
<td>NP → •NP •VP</td>
<td>VP → •Vi</td>
<td>VP → •Vt •NP</td>
<td>VP → •VP •PP</td>
<td>PP → •IN •NP</td>
<td>PP → •IN •NP</td>
</tr>
<tr>
<td>S → •NP •VP</td>
<td>NP → •NP •VP</td>
<td>VP → •Vt •NP</td>
<td>VP → •VP •PP</td>
<td>PP → •IN •NP</td>
<td>PP → •IN •NP</td>
<td>PP → •IN •NP</td>
<td>PP → •IN •NP</td>
<td>PP → •IN •NP</td>
</tr>
<tr>
<td>[1,2]</td>
<td>[2,2]</td>
<td>[3,3]</td>
<td>[4,3]</td>
<td>[5,5]</td>
<td>[6,5]</td>
<td>[7,5]</td>
<td>[8,5]</td>
<td>[8,8]</td>
</tr>
<tr>
<td>NN → woman</td>
<td>Vt → •saw</td>
<td>DT → •the</td>
<td>NN → •man</td>
<td>IN → •with</td>
<td>DT → •the</td>
<td>NN → •telescope</td>
<td>NN → •telescope</td>
<td>NN → •telescope</td>
</tr>
<tr>
<td>[1,3]</td>
<td>[2,3]</td>
<td>[3,4]</td>
<td>[4,4]</td>
<td>[5,6]</td>
<td>[6,6]</td>
<td>[7,7]</td>
<td>[8,8]</td>
<td>[8,8]</td>
</tr>
<tr>
<td>VP → •NP</td>
<td>VP → •NP</td>
<td>VT → •NP</td>
<td>VP → •NP</td>
<td>VP → •NP</td>
<td>VP → •NP</td>
<td>VP → •NP</td>
<td>VP → •NP</td>
<td>VP → •NP</td>
</tr>
<tr>
<td>[2,4]</td>
<td>[3,5]</td>
<td>[4,5]</td>
<td>[5,7]</td>
<td>[6,7]</td>
<td>[7,8]</td>
<td>[8,8]</td>
<td>[8,8]</td>
<td>[8,8]</td>
</tr>
</tbody>
</table>

---

**Example Sentence:**

"the woman saw the man with the telescope"
Advanced Human Language Technologies

Trees and Grammars

Constituency Parsing

Earley Algorithm

Earley Algorithm

Trees and Grammars

Constituency Parsing

Earley Algorithm

Chart 0:  
- \( \gamma \rightarrow S \)
- \( S \rightarrow NP \ VP \gamma \)
- \( \gamma \rightarrow S \)

Chart 1:  
- \( DT \rightarrow the \)
- \( NN \rightarrow woman \)
- \( Vt \rightarrow saw \)
- \( VP \rightarrow Vt \ NP \)

Chart 2:  
- \( NP \rightarrow DT \ NN \)
- \( S \rightarrow NP \ VP \)
- \( NP \rightarrow NP \ PP \)
- \( VP \rightarrow Vt \ NP \)
- \( VP \rightarrow Vt \ NP \)
- \( PP \rightarrow IN \ NP \)

Chart 3:  
- \( PP \rightarrow IN \ NP \)
- \( VP \rightarrow Vt \ NP \)
- \( NP \rightarrow DT \ NN \)
- \( NP \rightarrow NP \ PP \)

Chart 4:  
- \( NP \rightarrow NP \ PP \)
- \( VP \rightarrow Vt \ NP \)
- \( S \rightarrow NP \ VP \)
- \( \gamma \rightarrow S \)

Chart 5:  
- \( S \rightarrow NP \ VP \)
- \( \gamma \rightarrow S \)

Chart 6:  
- \( VP \rightarrow Vt \ NP \)
- \( VP \rightarrow Vt \ NP \)
- \( PP \rightarrow IN \ NP \)

Chart 7:  
- \( PP \rightarrow IN \ NP \)
- \( VP \rightarrow Vt \ NP \)
- \( S \rightarrow NP \ VP \)
- \( \gamma \rightarrow S \)

Chart 8:  
- \( NP \rightarrow DT \ NN \)

The woman saw the man with the telescope.
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

the woman saw the man with the telescope
Earley Algorithm
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm
The Earley Algorithm
Earley Algorithm

0 1 2 3 4 5 6 7 8

chart[0]

chart[1]

chart[2]

chart[3]

chart[4]

chart[5]

chart[6]

chart[7]

chart[8]

the woman saw the man with the telescope
The Earley Algorithm
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

chart[0] 0 1 2 3 4 5 6 7 8
[0,0] γ → S
S → NP VP
NP → DT NN
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 → VP PP
PP → IN NP
[5,5] PP → IN NP
NP → DT NN
NP → NP PP
[6,6] PP → IN NP
NP → DT NN
NP → NP PP
[8,8] PP → IN NP
NP → DT NN
NP → NP PP

chart[1] 0 1 2 3 4 5 6 7 8
[0,1] DT → the
NP → DT NN
[2,2] VP → Vi
VP → Vt NP
VP → VP PP
PP → IN NP

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

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

chart[4] 0 1 2 3 4 5 6 7 8

chart[5] 0 1 2 3 4 5 6 7 8
[0,5] S → NP VP
γ → S
[2,5] VP → Vt NP
VP → VP PP

chart[6] 0 1 2 3 4 5 6 7 8
[0,8] S → NP VP

chart[7] 0 1 2 3 4 5 6 7 8
[2,8] VP → VP PP
VP → VP PP

chart[8] 0 1 2 3 4 5 6 7 8
[3,8] NP → NP PP
[5,8] PP → IN NP

chart[0,0] γ → S
S → NP VP
NP → DT NN
NP → NP PP

chart[1,0] DT → the
NP → DT NN

chart[2,0] NP → DT NN
S → NP VP
NP → NP PP

chart[3,0] NP → DT NN
NP → NP PP

chart[4,0] S → NP VP
γ → S

chart[5,0] S → NP VP
γ → S

chart[6,0] VP → Vt NP
VP → VP PP

chart[7,0] VP → Vt NP
VP → VP PP

chart[8,0] S → NP VP

the woman saw the man with the telescope
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

0 1 2 3 4 5 6 7 8

<table>
<thead>
<tr>
<th>chart[0]</th>
</tr>
</thead>
<tbody>
<tr>
<td>γ → •S</td>
</tr>
<tr>
<td>S → •NP VP</td>
</tr>
<tr>
<td>NP → •DT NN</td>
</tr>
<tr>
<td>NP → •NP PP</td>
</tr>
<tr>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chart[1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT → the</td>
</tr>
<tr>
<td>NP → DT • NN</td>
</tr>
<tr>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chart[2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP → DT NN •</td>
</tr>
<tr>
<td>S → NP • VP</td>
</tr>
<tr>
<td>NP → NP • PP</td>
</tr>
<tr>
<td>0.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chart[3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP → DT NN •</td>
</tr>
<tr>
<td>VP → Vt • NP</td>
</tr>
<tr>
<td>VP → VP • PP</td>
</tr>
<tr>
<td>0.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chart[4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP • VP</td>
</tr>
<tr>
<td>γ → S •</td>
</tr>
<tr>
<td>0.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chart[5]</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP → Vt • NP</td>
</tr>
<tr>
<td>VP → VP • PP</td>
</tr>
<tr>
<td>2.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chart[6]</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP → Vt • NP</td>
</tr>
<tr>
<td>VP → VP • PP</td>
</tr>
<tr>
<td>3.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chart[7]</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP → Vt • NP</td>
</tr>
<tr>
<td>VP → VP • PP</td>
</tr>
<tr>
<td>4.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chart[8]</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP → Vt • NP</td>
</tr>
<tr>
<td>VP → VP • PP</td>
</tr>
<tr>
<td>5.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>complete</td>
</tr>
<tr>
<td>complete</td>
</tr>
<tr>
<td>complete</td>
</tr>
<tr>
<td>complete</td>
</tr>
<tr>
<td>complete</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0 1 2 3 4 5 6 7 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>the woman saw the man with the telescope</td>
</tr>
</tbody>
</table>
### Earley Algorithm

#### Trees and Grammars

**Constituency Parsing**

**Earley Algorithm**

---

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[0.0]</td>
<td>[0.1]</td>
<td>[0.2]</td>
<td>[0.3]</td>
<td>[0.4]</td>
<td>[0.5]</td>
<td>[0.6]</td>
<td>[0.7]</td>
</tr>
<tr>
<td>DT → the</td>
<td>DT → the</td>
<td>NP → DT</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>S → NP VP</td>
<td>VP → VP PP</td>
<td>NP → NP PP</td>
</tr>
<tr>
<td>NP → DT</td>
<td>NN → woman</td>
<td>VP → VP</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>VP → VP</td>
<td>NP → NP PP</td>
<td>NP → NP PP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VP → NP</td>
<td>VN → DT</td>
<td>VP → NP</td>
<td>VN → DT</td>
<td>VN → NP PP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[1.0]</td>
<td>[1.1]</td>
<td>[1.2]</td>
<td>[1.3]</td>
<td>[1.4]</td>
<td>[1.5]</td>
<td>[1.6]</td>
<td>[1.7]</td>
</tr>
<tr>
<td>DT → the</td>
<td>DT → the</td>
<td>NN → woman</td>
<td>NP → DT</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
</tr>
<tr>
<td>NP → DT</td>
<td>NN → woman</td>
<td>VP → VP</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>VP → VP</td>
<td>NP → NP PP</td>
<td>NP → NP PP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VP → NP</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>VP → VP</td>
<td>NP → NP PP</td>
<td>NP → NP PP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[2.0]</td>
<td>[2.1]</td>
<td>[2.2]</td>
<td>[2.3]</td>
<td>[2.4]</td>
<td>[2.5]</td>
<td>[2.6]</td>
<td>[2.7]</td>
</tr>
<tr>
<td>DT → the</td>
<td>DT → the</td>
<td>NN → woman</td>
<td>NP → DT</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
</tr>
<tr>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[3.0]</td>
<td>[3.1]</td>
<td>[3.2]</td>
<td>[3.3]</td>
<td>[3.4]</td>
<td>[3.5]</td>
<td>[3.6]</td>
<td>[3.7]</td>
</tr>
<tr>
<td>DT → the</td>
<td>DT → the</td>
<td>NN → woman</td>
<td>NP → DT</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
</tr>
<tr>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[4.0]</td>
<td>[4.1]</td>
<td>[4.2]</td>
<td>[4.3]</td>
<td>[4.4]</td>
<td>[4.5]</td>
<td>[4.6]</td>
<td>[4.7]</td>
</tr>
<tr>
<td>DT → the</td>
<td>DT → the</td>
<td>NN → woman</td>
<td>NP → DT</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
</tr>
<tr>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[5.0]</td>
<td>[5.1]</td>
<td>[5.2]</td>
<td>[5.3]</td>
<td>[5.4]</td>
<td>[5.5]</td>
<td>[5.6]</td>
<td>[5.7]</td>
</tr>
<tr>
<td>DT → the</td>
<td>DT → the</td>
<td>NN → woman</td>
<td>NP → DT</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
</tr>
<tr>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[6.0]</td>
<td>[6.1]</td>
<td>[6.2]</td>
<td>[6.3]</td>
<td>[6.4]</td>
<td>[6.5]</td>
<td>[6.6]</td>
<td>[6.7]</td>
</tr>
<tr>
<td>DT → the</td>
<td>DT → the</td>
<td>NN → woman</td>
<td>NP → DT</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
</tr>
<tr>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[7.0]</td>
<td>[7.1]</td>
<td>[7.2]</td>
<td>[7.3]</td>
<td>[7.4]</td>
<td>[7.5]</td>
<td>[7.6]</td>
<td>[7.7]</td>
</tr>
<tr>
<td>DT → the</td>
<td>DT → the</td>
<td>NN → woman</td>
<td>NP → DT</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
</tr>
<tr>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[8.0]</td>
<td>[8.1]</td>
<td>[8.2]</td>
<td>[8.3]</td>
<td>[8.4]</td>
<td>[8.5]</td>
<td>[8.6]</td>
<td>[8.7]</td>
</tr>
<tr>
<td>DT → the</td>
<td>DT → the</td>
<td>NN → woman</td>
<td>NP → DT</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
<td>VP → Vi</td>
</tr>
<tr>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
<td>NP → DT</td>
<td>NN → man</td>
</tr>
</tbody>
</table>

---

**The Sentence:**

the woman saw the man with the telescope
### Earley Algorithm

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8</td>
<td>DT → the</td>
<td>NP → DT • NN</td>
<td>S → NP • VP</td>
<td>VP → Vt • NP</td>
<td>DT → the</td>
<td>NP → DT • NN</td>
<td>S → NP • VP</td>
<td>VP → Vt • NP</td>
</tr>
<tr>
<td>γ → •S</td>
<td>NN → woman</td>
<td>VP → •Vi</td>
<td>VP → •Vt • NP</td>
<td>DT → the</td>
<td>NP → DT • NN</td>
<td>S → NP • VP</td>
<td>VP → Vt • NP</td>
<td></td>
</tr>
<tr>
<td>S → •NP VP</td>
<td>NP → •DT NN</td>
<td>VP → •Vt • NP</td>
<td>VP → •VP • PP</td>
<td>PP → •IN • NP</td>
<td>NP → •DT NN</td>
<td>VP → •VP • PP</td>
<td>NP → •IN • NP</td>
<td></td>
</tr>
<tr>
<td>NP → •NP PP</td>
<td>NP → •DT NN</td>
<td>VP → •Vt • NP</td>
<td>VP → •VP • PP</td>
<td>PP → •IN • NP</td>
<td>NP → •DT NN</td>
<td>VP → •VP • PP</td>
<td>NP → •IN • NP</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0 1 2 3 4 5 6 7 8</th>
<th>the</th>
<th>woman</th>
<th>saw</th>
<th>the</th>
<th>man</th>
<th>with</th>
<th>the</th>
<th>telescope</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>γ → •S</td>
<td>S → •NP VP</td>
<td>VP → •Vt • NP</td>
<td>VP → •VP • PP</td>
<td>PP → •IN • NP</td>
<td>NP → •DT NN</td>
<td>VP → •VP • PP</td>
<td>NP → •IN • NP</td>
<td>NP → •DT NN</td>
</tr>
</tbody>
</table>
The Earley Algorithm is a complete, predictive, and deterministic parsing algorithm that can be applied to a wide range of context-free grammars. It is particularly useful for parsing natural language sentences. The algorithm works by constructing a table (or chart) that represents the possible parses of a sentence at each position. Each entry in the chart represents a combination of non-terminals, with their predecessors at the current stage.

For example, consider the sentence: "The woman saw the man with the telescope." The Earley Algorithm begins with an initial parse stack that contains the start symbol, S. As the algorithm processes each word in the sentence, it updates the chart to reflect the possible parses at that point. Each cell in the chart stores a parse tree, which is a node with a non-terminal symbol representing a phrase structure in the sentence.

The chart is updated by considering all possible transitions from non-terminals to terminals or other non-terminals at each stage. The algorithm can predict the next possible parse by examining the chart entries from previous stages. The Earley Algorithm is guaranteed to find all valid parses of the sentence and is particularly adept at handling languages with ambiguous parses.
Earley Algorithm

The diagram illustrates the Earley algorithm for parsing the sentence "The woman saw the man with the telescope." The chart is divided into eight states, each representing a different part of the parse tree. Each cell in the chart contains rules for expanding the parse tree, such as "NP → DT NN" or "VP → Vt NP." The algorithm progresses through these states, starting from the first state and moving to subsequent states as the parser processes the input sentence.

The sentence is broken down into its constituent parts:
- **The**: DT (determiner) → the
- **woman**: NN (noun) → woman
- **saw**: Vt (verb) → saw
- **the**: DT (determiner) → the
- **man**: NN (noun) → man
- **with**: IN (preposition) → with
- **the**: DT (determiner) → the
- **telescope**: NN (noun) → telescope

These parts are connected to form the complete sentence in the final state of the chart.
Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

0 1 2 3 4 5 6 7 8

| the | woman | saw | the | man | with | the | telescope |
### Earley Algorithm

Here is a representation of the Earley algorithm for parsing the sentence: **the woman saw the man with the telescope**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>chart[0]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>chart[1]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>chart[2]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>chart[3]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>chart[4]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>chart[5]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>chart[6]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>chart[7]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>chart[8]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Earley algorithm constructs a parse tree by inserting shifts and reduces where possible. The chart representation shows the states and transitions at each step of the parsing process. Each cell in the chart corresponds to a non-terminal symbol or a transition rule.

### Grammar Rules
- **S**: \( S \rightarrow NP \ VP \)
- **NP**: \( NP \rightarrow DT \ NN \) \( \gamma \rightarrow S \) \( NP \rightarrow NP \ PP \) \( NP \rightarrow DT \ NN \) \( NP \rightarrow NP \ PP \)
- **VP**: \( VP \rightarrow Vt \ NP \) \( VP \rightarrow VP \ PP \) \( VP \rightarrow DT \ NN \) \( VP \rightarrow NP \ PP \)
- **PP**: \( PP \rightarrow IN \ NP \) \( PP \rightarrow IN \ NP \) \( PP \rightarrow IN \ NP \) \( PP \rightarrow IN \ NP \)
- **words**:
  - **the**: DT, NN
  - **woman**: NN
  - **saw**: Vt
  - **man**: NN
  - **with**: IN
  - **telescope**: NN

The algorithm proceeds through the chart, inserting shifts and transitions, until a complete parse tree is constructed, which correctly represents the syntactic structure of the input sentence.
Earley Algorithm

The Earley algorithm is a dynamic programming algorithm for parsing sentences into constituent structures. It is characterized by its ability to handle context-free languages and produce a parse tree for each sentence.

The Earley algorithm works by constructing a chart, where each cell corresponds to a possible parse of the input sentence. The chart is filled row by row, with each row corresponding to a different position in the input sentence. Each cell contains a set of productions that are possible for the given position.

The chart is divided into eight states:
- **S** (start symbol)
- **NP** (noun phrase)
- **VP** (verb phrase)
- **PP** (prepositional phrase)
- **DT** (determiner)
- **NN** (noun)
- **Vt** (verb, transitive)
- **Vf** (verb, intransitive)

### Productions

- **S → NP VP**
- **NP → DT NN**
- **NP → NP PP**
- **VP → Vt NP**
- **VP → VP PP**
- **PP → IN NP**
- **DT → the**
- **NN → woman**
- **NN → man**
- **Vt → saw**
- **Vt → with**
- **IN → with**
- **IN → the**
- **DT → the**
- **NN → telescope**

The Earley algorithm proceeds by predicting the next possible symbols and scanning the chart to update the states.

**Example:**

The sentence "the woman saw the man with the telescope" can be parsed using the Earley algorithm, resulting in the following chart:

```
<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td><strong>the</strong></td>
<td><strong>woman</strong></td>
<td><strong>saw</strong></td>
<td><strong>the</strong></td>
<td><strong>man</strong></td>
<td><strong>with</strong></td>
<td><strong>the</strong></td>
<td><strong>telescope</strong></td>
<td></td>
</tr>
<tr>
<td><strong>0.0</strong></td>
<td><strong>0.1</strong></td>
<td><strong>0.2</strong></td>
<td><strong>0.3</strong></td>
<td><strong>0.4</strong></td>
<td><strong>0.5</strong></td>
<td><strong>0.6</strong></td>
<td><strong>0.7</strong></td>
<td><strong>0.8</strong></td>
</tr>
<tr>
<td>γ → •S</td>
<td>γ → •NP</td>
<td>γ → •NP</td>
<td>γ → •DT</td>
<td>γ → •IN</td>
<td>γ → •DT</td>
<td>γ → •IN</td>
<td>γ → •IN</td>
<td>γ → •IN</td>
</tr>
<tr>
<td>S → NP</td>
<td>VP → Vt</td>
<td>NP → DT</td>
<td>VP → VP</td>
<td>PP → IN</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>PP → IN</td>
</tr>
<tr>
<td>NP → DT</td>
<td>VP → VP</td>
<td>NP → DT</td>
<td>PP → IN</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>PP → IN</td>
<td>NP → DT</td>
<td>PP → IN</td>
</tr>
<tr>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
</tr>
<tr>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
<td>NP → DT</td>
</tr>
</tbody>
</table>
```

The chart is filled with the possible parses at each step, with the final state indicating the complete parse of the sentence.
Earley Algorithm
The Earley Algorithm

Trees and Grammars
Constituency Parsing
Earley Algorithm

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

chart[1]
- $\gamma \rightarrow \bullet S$
- $S \rightarrow \bullet NP \ VP$
- $NP \rightarrow \bullet DT \ NN$
- $NP \rightarrow \bullet NP \ PP$

chart[2]
- $NP \rightarrow \bullet DT \ NN$
- $S \rightarrow \bullet NP \ VP$
- $NP \rightarrow \bullet NP \ PP$

chart[3]
- $VP \rightarrow \bullet Vt \ NP$
- $VP \rightarrow \bullet VP \ PP$
- $PP \rightarrow \bullet IN \ NP$

chart[4]
- $VP \rightarrow \bullet Vt \ NP$
- $VP \rightarrow \bullet VP \ PP$
- $PP \rightarrow \bullet IN \ NP$
- $NP \rightarrow \bullet DT \ NN$
- $NP \rightarrow \bullet NP \ PP$

chart[5]
- $VP \rightarrow \bullet Vt \ NP$
- $VP \rightarrow \bullet VP \ PP$
- $PP \rightarrow \bullet IN \ NP$

chart[6]
- $NP \rightarrow \bullet NP \ PP$
- $NP \rightarrow \bullet DT \ NN$

chart[7]
- $NP \rightarrow \bullet NP \ PP$

chart[8]
- $S \rightarrow \bullet NP \ VP$
- $\gamma \rightarrow \bullet S$

The sentence: "the woman saw the man with the telescope"
Earley Algorithm

Trees and Grammars

Constituency Parsing

Earley Algorithm

```
0 1 2 3 4 5 6 7 8

chart[0]
γ → S
S → NP VP
NP → DT NN
NP → DT NN
NP → DT PP

chart[1]
DT → the
NP → DT NN
NP → DT NN
NP → DT PP

chart[2]
NP → DT NN
S → NP VP
NP → NP PP
NP → NP PP

chart[3]
NP → DT NN
NP → DT NP
NP → DT NP

chart[4]
VP → Vt NP
VP → NP PP
VP → NP PP

chart[5]
S → NP VP
γ → S

chart[6]
VP → Vt NP
VP → NP PP
VP → NP PP

chart[7]
NP → NP PP
NP → NP PP
NP → NP PP

chart[8]
S → NP VP
γ → S
```

```
| the | woman | saw | the | man | with | the | telescope |
```
CKY vs Earley

**CKY**
- Bottom-up
- Requires CNF
- Can compute all trees
- $O(n^3)$
- Straightforward probabilistic version

**Earley**
- Top-down
- Any CFG can be used, no need for CNF
- Can compute all trees
- $O(n^3)$
- Not so straightforward probabilistic version
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

Trees and Grammars
Constituency Parsing
Earley Algorithm
Natural Language is not Context-Free

May contain non-projective structures: