<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td>Idea</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>Algorithm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>Tabulation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>4</strong></td>
<td>Parsing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>5</strong></td>
<td>Lookaheads</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Goal: overcome problems with pure TD/BU approaches. Earley’s algorithm can be seen as a

- **bottom-up parser with top-down control**, i.e., a bottom-up parsing that does only reductions that can be top-down predicted from S, or a

- **top-down parser with bottom-up recognition**
At each time of parsing, one production $A \rightarrow X_1 \ldots X_k$ is considered such that

- some part $X_1 \ldots X_i$ has already been bottom-up recognized (completed)
- while some part $X_{i+1} \ldots X_k$ has been top-down predicted.

As in the left-corner chart parser, this situation can be characterized by a dotted production (sometimes called Earley item)
$A \rightarrow X_1 \ldots X_i \bullet X_{i+1} \ldots X_k$.

Dotted productions are called active items. Productions of the form $A \rightarrow \alpha \bullet$ are called completed items.
The Earley parser simulates a top-down left-to-right depth-first traversal of the parse tree while moving the dot such that for each node

- first, the dot is to its left (the node is predicted),
- then the dot traverses the tree below,
- then the dot is to its right (the subtree below the node is completed)

Each state of the parser can be characterized by a set of dotted productions $A \rightarrow X_1 \ldots X_i \bullet X_{i+1} \ldots X_k$. For each of these, one needs to keep track of the input part spanned by the completed part of the rhs, i.e., by $X_1 \ldots X_i$. 
The items describing partial results of the parser contain a dotted production and the start and end index of the completed part of the rhs:

Item form: \([A \rightarrow \alpha \bullet \beta, i, j]\) with \(A \rightarrow \alpha \beta \in P, 0 \leq i \leq j \leq n\).

Parsing starts with predicting all \(S\)-productions:

Axioms: \([S \rightarrow \bullet \alpha, 0, 0]\) \(S \rightarrow \alpha \in P\)
Algorithm (2)

If the dot of an item is followed by a non-terminal symbol $B$, a new $B$-production can be predicted. The completed part of the new item (still empty) starts at the index where the completed part of the first item ends.

Predict: $\frac{[A \rightarrow \alpha \cdot B\beta, i, j]}{[B \rightarrow \cdot \gamma, j, j]} B \rightarrow \gamma \in P$

If the dot of an item is followed by a terminal symbol $a$ that is the next input symbol, then the dot can be moved over this terminal (the terminal is scanned). The end position of the completed part is incremented.

Scan: $\frac{[A \rightarrow \alpha \cdot a\beta, i, j]}{[A \rightarrow \alpha a \cdot \beta, i, j + 1]} w_{j+1} = a$
Algorithm (3)

If the dot of an item is followed by a non-terminal symbol $B$ and if there is a second item with a dotted $B$-production and a fully completed rhs and if, furthermore, the completed part of the second item starts at the position where the completed part of the first ends, then the dot in the first can be moved over the $B$ while changing the end index to the end index of the completed $B$-production.

Complete: \[
[A \rightarrow \alpha \bullet B\beta, i, j], [B \rightarrow \gamma\bullet, j, k]
\]
\[
[A \rightarrow \alpha B \bullet \beta, i, k]
\]

The parser is successful if a completed $S$-production spanning the entire input can be deduced:

Goal items: $[S \rightarrow \alpha\bullet, 0, n]$ for some $S \rightarrow \alpha \in P$. 
Note that

- this algorithm can deal with $\epsilon$-productions;
- loops and left-recursions are no problem since an active item is generated only once;
- the algorithm works for any type of CFG.
Example

Productions: \( S \rightarrow aB \mid bA, A \rightarrow aS \mid bAA \mid a, B \rightarrow bS \mid aBB \mid b. \)

\( w = abab. \)

Set of deduced items:
1. \([S \rightarrow \bullet aB, 0, 0]\) axiom
2. \([S \rightarrow \bullet bA, 0, 0]\) axiom
3. \([S \rightarrow a \bullet B, 0, 1]\) scan with 1.
4. \([B \rightarrow \bullet bS, 1, 1]\) predict with 3.
5. \([B \rightarrow \bullet b, 1, 1]\) predict with 3.
6. \([B \rightarrow \bullet aBB, 1, 1]\) predict with 3.
7. \([B \rightarrow b \bullet S, 1, 2]\) scan with 4.
8. \([B \rightarrow b \bullet, 1, 2]\) scan with 5.
9. \([S \rightarrow aB\bullet, 0, 2]\) complete with 3. and 8.
Algorithm (6)

Example continued

10. \([S \rightarrow \bullet aB, 2, 2]\) predict with 7.
11. \([S \rightarrow \bullet bA, 2, 2]\) predict with 7.
12. \([S \rightarrow a \bullet B, 2, 3]\) scan with 10.
13. \([B \rightarrow \bullet bS, 3, 3]\) predict with 12.
14. \([B \rightarrow \bullet b, 3, 3]\) predict with 12.
15. \([B \rightarrow \bullet aBB, 3, 3]\) predict with 12.
16. \([B \rightarrow b \bullet S, 3, 4]\) scan with 13.
17. \([B \rightarrow b\bullet, 3, 4]\) scan with 14.
18. \([S \rightarrow \bullet aB, 4, 4]\) predict with 16.
19. \([S \rightarrow \bullet bA, 4, 4]\) predict with 16.
20. \([S \rightarrow aB\bullet, 2, 4]\) complete with 12. and 17.
21. \([B \rightarrow bS\bullet, 1, 4]\) complete with 7. and 20.
22. \([S \rightarrow aB\bullet, 0, 4]\) complete with 3. and 21.
Soundness and completeness:
The following holds:

\[[A \rightarrow \alpha \bullet \beta, i, j]\]

iff

\[S \Rightarrow w_1 \ldots w_i A \gamma \Rightarrow w_1 \ldots w_i \alpha \beta \gamma \Rightarrow w_1 \ldots w_i w_{i+1} \ldots w_j \beta \gamma\]

for some \(\gamma \in (N \cup T^*)\).

The algorithm is in particular \textbf{prefix-valid}: if there is an item with end position \(j\), then there is a word in the language with prefix \(w_1 \ldots w_j\).
In addition, one can use passive items $[A, i, j]$ with $A \in N$, $0 \leq i \leq j \leq n$. Then, we need an additional convert rule, that converts a completed active item into a passive one:

\[
[ B \to \gamma \bullet, j, k ] \quad \xrightarrow{\text{Convert:}} \quad [ B, j, k ]
\]

The goal item is then $[ S, 0, n ]$. 
The \textit{Complete} rule can use passive items now:

\[
\begin{align*}
\text{Complete:} & \quad [A \rightarrow \alpha \bullet B\beta, i, j], [B, j, k] \\
& \quad \Rightarrow [A \rightarrow \alpha B \bullet \beta, i, k]
\end{align*}
\]

The advantage is that we obtain a higher degree of factorization: A $B$-subtree might have different analyses. \textit{Complete} can use this $B$ category now independent from the concrete analyses, i.e., there is only one single application of \textit{Complete} for all of them.
We can tabulate the dotted productions depending on the indices of the covered input.

I.e., we adopt a \((n + 1) \times (n + 1)\)-chart \(C\) with 
\(A \rightarrow \alpha \mathbin{\bullet} \beta \in C_{i,j}\) iff 
\([A \rightarrow \alpha \mathbin{\bullet} \beta, i, j]\).

The chart is initialized with 
\(C_{0,0} := \{ S \rightarrow \bullet \alpha \mid S \rightarrow \alpha \in P\} \) and 
\(C_{i,j} = \emptyset\) for all \(i, j \in [0..n]\) with \(i \neq 0\) or \(j \neq 0\). 
It can then be filled in the following way:
Let us consider the version without passive items.

The chart is filled row by row: for every end-of-span index $k$:

- we first compute all applications of $\text{predict}$ and $\text{complete}$ that yield new items with end-of-span index $k$;
- then, we compute all applications of $\text{scan}$ which gives items with end-of-span index $k + 1$. 
for all $k \in [0..n]$: 
    do until chart does not change any more:
        for all $j \in [0..k]$ and all $p \in C_{j,k}$:
            if $p = A \rightarrow \alpha \cdot B\beta$
                then add $B \rightarrow \cdot \gamma$ to $C_{k,k}$ for all $B \rightarrow \gamma \in P$
            else if $p = B \rightarrow \gamma\cdot$
                then for all $i \in [0..j]$: 
                    if there is a $A \rightarrow \alpha \cdot B\beta \in C_{i,j}$
                        then add $A \rightarrow \alpha B \cdot \beta$ to $C_{i,k}$
        for all $j \in [0..k]$ and for all $p \in C_{j,k}$:
            if $p = A \rightarrow \alpha \cdot w_{k+1}\beta$
                then add $A \rightarrow \alpha w_{k+1} \cdot \beta$ to $C_{j,k+1}$
Note that *predict* and *complete* do not increment the end of the spanned input, i.e., they add only elements to the fields $C_{...k}$ (the $k$-th row of the chart).

*Scan* however adds elements to the $C_{...k+1}$ (the $k + 1$-th row).

This is why first, all possible *predict* and *complete* operations are performed to generate new chart entries in the $k$-th row. Then, *scan* is applied and one can move on to the next row $k + 1$.

Since *predict* and *complete* are applied as often as possible, $\epsilon$-productions and left recursion are no problem for this algorithm.
Implementation:
Besides the chart, for every $k$, we keep an agenda $A_k$ of those items from the chart that still need to be processed.
Initially, for $k = 0$, this agenda contains all $S \rightarrow \bullet \alpha$, the other agendas are empty.
We process the items in the agendas from $k = 0$ to $k = n$. For each $k$, we stop once the $k$-agenda is empty.
Items $x$ of the form $A \rightarrow \alpha \cdot B\beta$ trigger a *predict* operation. The newly created items, if they are not yet in the chart, are added to chart and $k$-agenda.

In addition, if $\epsilon$-productions are allowed, $x$ also triggers a *complete* where the chart is searched for a completed $B$-item ranging from $k$ to $k$. The new items (if not in the chart yet) are added to the $k$-agenda and the chart. $x$ is removed from the $k$-agenda.
Items $x$ of the form $B \rightarrow \gamma \bullet$ trigger a complete operation where the chart is searched for corresponding items $A \rightarrow \alpha \bullet B\beta$. The newly created items are added to the chart and the $k$-agenda (if they are not yet in the chart), $x$ is removed from the $k$-agenda.

Items $x$ of the form $A \rightarrow \alpha \bullet a\beta$ trigger a scan operation. The newly created items (if not yet in the chart) are added to the chart and the $k+1$-agenda, $x$ is removed from the $k$-agenda.
Example without $\epsilon$-productions

Productions $S \rightarrow ASB \mid c, A \rightarrow a, B \rightarrow b$, input $w = acb$

$A_0 = \{ [S \rightarrow \bullet ASB, 0, 0], [S \rightarrow \bullet c, 0, 0] \}$

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>$S \rightarrow \bullet ASB$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S \rightarrow \bullet c$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$S \rightarrow \bullet ASB$ triggers a predict:
Example continued

\[ A_0 = \{ [S \rightarrow \bullet c, 0, 0], [A \rightarrow \bullet a, 0, 0] \} \]

<table>
<thead>
<tr>
<th>()</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(S \rightarrow \bullet ASB)</td>
<td>(S \rightarrow \bullet c)</td>
<td>(S \rightarrow \bullet c)</td>
<td>(A \rightarrow \bullet a)</td>
</tr>
</tbody>
</table>

\(S \rightarrow \bullet c\) triggers a scan that fails, \(A \rightarrow \bullet a\) triggers a successful scan:
Example continued

\[ A_0 = \{\}, A_1 = \{[A \rightarrow a\bullet, 0, 1]\} \]

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A → a\bullet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>S → •ASB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>S → •c</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A → •a</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A → a\bullet triggers a complete:
**Tabulation (12)**

**Example continued**

\[ A_0 = \{\}, \ A_1 = \{[S \rightarrow A \cdot SB, 0, 1]\}\]

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ S \rightarrow A \cdot SB \text{ triggers a predict:} \]
Example continued

\[ A_0 = \{\}, \ A_1 = \{[S \rightarrow \bullet ASB, 1, 1], [S \rightarrow \bullet c, 1, 1]\} \]

<table>
<thead>
<tr>
<th>2</th>
<th>( S \rightarrow A \bullet SB )</th>
<th>( S \rightarrow \bullet ASB )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( A \rightarrow a\bullet )</td>
<td>( S \rightarrow \bullet c )</td>
</tr>
<tr>
<td>0</td>
<td>( S \rightarrow \bullet ASB )</td>
<td>( S \rightarrow \bullet c )</td>
</tr>
</tbody>
</table>

\( S \rightarrow \bullet ASB \) triggers a predict:
Tabulation (14)

Example continued

\[ A_0 = \{ \}, A_1 = \{ [S \rightarrow \bullet c, 1, 1], [A \rightarrow \bullet a, 1, 1] \} \]

<table>
<thead>
<tr>
<th>2</th>
<th>( S \rightarrow A \bullet SB )</th>
<th>( S \rightarrow \bullet ASB )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( A \rightarrow a \bullet )</td>
<td>( S \rightarrow \bullet c )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( A \rightarrow \bullet a )</td>
</tr>
<tr>
<td>0</td>
<td>( S \rightarrow \bullet ASB )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( S \rightarrow \bullet c )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( A \rightarrow \bullet a )</td>
<td></td>
</tr>
</tbody>
</table>

\[
S \rightarrow \bullet c \text{ triggers a scan:}
\]

\[
\begin{array}{c|c|c}
0 & 1 & 2 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
3 & & \\
\end{array}
\]
Example continued

\[ A_0 = \{\}, A_1 = \{[A \rightarrow \bullet a, 1, 1]\}, A_2 = \{[S \rightarrow c\bullet, 1, 2]\} \]

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow \bullet ASB )</td>
<td>( S \rightarrow c \bullet )</td>
<td>( S \rightarrow \bullet ASB )</td>
<td>( S \rightarrow c \bullet )</td>
</tr>
<tr>
<td>( S \rightarrow \bullet c )</td>
<td>( A \rightarrow a \bullet )</td>
<td>( A \rightarrow a \bullet )</td>
<td>( A \rightarrow a \bullet )</td>
</tr>
<tr>
<td>( S \rightarrow \bullet c )</td>
<td>( A \rightarrow a )</td>
<td>( A \rightarrow a )</td>
<td>( A \rightarrow a )</td>
</tr>
<tr>
<td>( A \rightarrow a )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( A \rightarrow a \) triggers a scan that fails, then \( S \rightarrow c\bullet \) triggers a complete:
Tabulation (16)

Example continued

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow AS \bullet B$</td>
<td>$S \rightarrow c\bullet$</td>
<td>$S \rightarrow ASB$</td>
<td>$S \rightarrow c$</td>
</tr>
<tr>
<td>$S \rightarrow ASB$</td>
<td>$S \rightarrow ASB$</td>
<td>$S \rightarrow ASB$</td>
<td>$S \rightarrow ASB$</td>
</tr>
<tr>
<td>$S \rightarrow c$</td>
<td>$S \rightarrow c$</td>
<td>$S \rightarrow c$</td>
<td>$S \rightarrow c$</td>
</tr>
<tr>
<td>$A \rightarrow a\bullet$</td>
<td>$A \rightarrow a\bullet$</td>
<td>$A \rightarrow a\bullet$</td>
<td>$A \rightarrow a\bullet$</td>
</tr>
<tr>
<td>$A \rightarrow a$</td>
<td>$A \rightarrow a$</td>
<td>$A \rightarrow a$</td>
<td>$A \rightarrow a$</td>
</tr>
</tbody>
</table>

$S \rightarrow AS \bullet B$ triggers the prediction of $[B \rightarrow \bullet b, 2, 2]$, which triggers a successful scan:
Example continued

\[ A_0 = \{ \}, A_1 = \{ \}, A_2 = \{ \}, A_3 = \{ [B \rightarrow b\bullet, 2, 3] \} \]

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

\[ B \rightarrow b\bullet \text{ triggers a complete which leads to a goal item:} \]
### Example continued

\[
A_0 = \{\}, A_1 = \{\}, A_2 = \{\}, A_3 = \{
\]

<table>
<thead>
<tr>
<th></th>
<th>( S \rightarrow ASB \bullet )</th>
<th>( S \rightarrow c \bullet )</th>
<th>( B \rightarrow b \bullet )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>( S \rightarrow AS \bullet B )</td>
<td>( S \rightarrow c \bullet )</td>
<td>( B \rightarrow b \bullet )</td>
</tr>
<tr>
<td>2</td>
<td>( S \rightarrow A \bullet SB )</td>
<td>( S \rightarrow \bullet ASB )</td>
<td>( B \rightarrow b \bullet )</td>
</tr>
<tr>
<td>1</td>
<td>( S \rightarrow A \bullet SB )</td>
<td>( S \rightarrow \bullet ASB )</td>
<td>( B \rightarrow b \bullet )</td>
</tr>
<tr>
<td></td>
<td>( A \rightarrow a \bullet )</td>
<td>( S \rightarrow \bullet c )</td>
<td>( A \rightarrow a \bullet )</td>
</tr>
<tr>
<td>0</td>
<td>( S \rightarrow \bullet ASB )</td>
<td>( S \rightarrow \bullet c )</td>
<td>( A \rightarrow a \bullet )</td>
</tr>
<tr>
<td></td>
<td>( S \rightarrow \bullet c )</td>
<td>( S \rightarrow \bullet c )</td>
<td>( A \rightarrow a \bullet )</td>
</tr>
<tr>
<td></td>
<td>( A \rightarrow \bullet a )</td>
<td>( S \rightarrow \bullet c )</td>
<td>( A \rightarrow a \bullet )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Example with $\epsilon$-productions

Productions $S \rightarrow ABB$, $A \rightarrow a$, $B \rightarrow \epsilon$, input $w = a$

<table>
<thead>
<tr>
<th>1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A $\rightarrow$ a$\bullet$</td>
<td></td>
</tr>
<tr>
<td>S $\rightarrow$ A$\bullet$ BB</td>
<td>B $\rightarrow$ $\bullet$</td>
</tr>
<tr>
<td>S $\rightarrow$ AB$\bullet$ B</td>
<td></td>
</tr>
<tr>
<td>S $\rightarrow$ ABB$\bullet$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S $\rightarrow$ $\bullet$ABB</td>
<td></td>
</tr>
<tr>
<td>A $\rightarrow$ $\bullet$a</td>
<td></td>
</tr>
</tbody>
</table>

| 0 | 1 |
### Overview: TD/BU and mixed approaches:

<table>
<thead>
<tr>
<th>Method</th>
<th>Approach</th>
<th>Item form</th>
<th>Chart Parsing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-down</td>
<td>TD</td>
<td>$[\Gamma, i]$</td>
<td>no</td>
</tr>
<tr>
<td>Bottom-up</td>
<td>BU</td>
<td>$[\Gamma, i]$</td>
<td>no</td>
</tr>
<tr>
<td>CYK</td>
<td>BU</td>
<td>$[A, i, l]$</td>
<td>yes</td>
</tr>
<tr>
<td>Left corner</td>
<td>mixed</td>
<td>$[\Gamma_{compl}, \Gamma_{td}, \Gamma_{lhs}]$</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$[A \rightarrow \alpha \bullet \beta, i, l]$</td>
<td>yes</td>
</tr>
<tr>
<td>Earley</td>
<td>mixed</td>
<td>$[A \rightarrow \alpha \bullet \beta, i, j]$</td>
<td>yes</td>
</tr>
</tbody>
</table>
So far, we have a recognizer.

- One way to extend it to a parser is to read off the parse tree in a top-down way from the chart.
- Alternatively, in every completion step, we can record in the chart the way the new item can be obtained by adding pointers to its pair of antecedents. Then, for constructing the parse tree, we only need to follow the pointers.
First possibility (initial call parse-tree(S, 0, n)):

parse-tree(X, i, j)
  trees := ∅;
  if X = w_j and j = i + 1 then trees := {w_j}
  else for all X → X_1...X_r • ∈ C_{i,j}
    for all i_1,...,i_r, i ≤ i_1 ≤ ... ≤ i_{r-1} ≤ i_r = j
      and all t_1,...,t_r with
        t_1 ∈ parse-tree(X_1, i, i_1) and
        t_l ∈ parse-tree (X_l, i_{l-1}, i_l) for 1 < l ≤ r:
      trees := trees ∪ {X(t_1,...,t_r)};
  output trees
Second possibility:

- We equip items with an additional set of pairs of pointers to other items in the item set.

- Whenever an item \([A \rightarrow \alpha A \bullet \beta, i, k]\) is obtained in a complete operation from \([A \rightarrow \alpha \bullet A\beta, i, j]\) and \([A \rightarrow \gamma \bullet, j, k]\), we add a pair of pointers to the two antecedent items to the pointer set of the consequent item.

- Obviously, items might have more than one element in their set if the grammar is ambiguous.
Example: Backpointers

Productions $S \to AB, A \to Ac \mid a, B \to cB \mid b$, input $w = acb$.

Item set (with list of pointer pairs):

1. $[S \to \bullet AB, 0, 0]$
2. $[A \to \bullet Ac, 0, 0]$
3. $[A \to \bullet a, 0, 0]$
4. $[A \to a\bullet, 0, 1]$
5. $[A \to A \bullet c, 0, 1], \{\langle 2, 4 \rangle\}$
6. $[S \to A \bullet B, 0, 1], \{\langle 1, 4 \rangle\}$
7. $[B \to \bullet cB, 1, 1]$
8. $[B \to \bullet b, 1, 1]$
9. $[A \to Ac\bullet, 0, 2]$
10. $[B \to c \bullet B, 1, 2]$
11. $[A \to A \bullet c, 0, 2], \{\langle 2, 9 \rangle\}$
12. $[S \to A \bullet B, 0, 2], \{\langle 1, 9 \rangle\}$
13. $[B \to \bullet cB, 2, 2]$
14. $[B \to \bullet b, 2, 2]$
15. $[B \to b\bullet, 2, 3]$
16. $[B \to cB\bullet, 1, 3], \{\langle 10, 15 \rangle\}$
17. $[S \to AB\bullet, 0, 3], \{\langle 6, 16 \rangle, \langle 12, 15 \rangle\}$
Lookaheads

Two kinds of lookaheads: a prediction lookahead and a reduction lookahead:

Predict with lookahead:

\[
\frac{[A \rightarrow \alpha \bullet B\beta, i, j]}{[B \rightarrow \bullet \gamma, j, j]} \quad B \rightarrow \gamma \in P, \ w_{j+1} \in \text{First}(\gamma) \text{ or } \epsilon \in \text{First}(\gamma)
\]

Complete with lookahead:

\[
\frac{[A \rightarrow \alpha \bullet B\beta, i, j], [B \rightarrow \gamma \bullet, j, k]}{[A \rightarrow \alpha B \bullet \beta, i, k]} \quad w_{k+1} \in \text{First}(\beta)
\text{ or } \epsilon \in \text{First}(\beta) \text{ and } w_{k+1} \in \text{Follow}(A)
\]

Instead of precomputing the Follow sets, one can compute the actual follows while predicting.
Conclusion

- Earley is a top-down restricted bottom-up parser.
- The three operations to compute new partial parsing results are predict, scan and complete.
- Earley is a chart parser.
- Earley can parse all CFGs.