**Constructing the Predictive Parser Table**

A predictive parse table has a row for each non-terminal X, and a column for each input token t. Entries $table[X,t]$ contain productions:

- for each $X \rightarrow \gamma$
  - for each $t$ in $FIRST(\gamma)$
    - $table[X,t] = X \rightarrow \gamma$
  - if $\gamma$ is nullable
    - for each $t$ in $FOLLOW(X)$
      - $table[X,t] = X \rightarrow \gamma$

**Compute the predictive parse table for**

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>$X \rightarrow a$</td>
<td>$X \rightarrow Y$</td>
<td>$X \rightarrow Y$</td>
</tr>
<tr>
<td>$Y$</td>
<td>$Y \rightarrow \epsilon$</td>
<td>$Y \rightarrow \epsilon$</td>
<td>$Y \rightarrow \epsilon$</td>
</tr>
<tr>
<td>$Z \rightarrow d$</td>
<td>$Z \rightarrow XYZ$</td>
<td>$Z \rightarrow XYZ$</td>
<td>$Z \rightarrow XYZ$</td>
</tr>
</tbody>
</table>

**Multiple entries in the Predictive parse table: Ambiguity**

An ambiguous grammar will lead to multiple entries in the parse table.

Our grammar IS ambiguous, e.g. $Z \rightarrow d$

but also $Z \rightarrow XYZ \rightarrow YZ \rightarrow d$

For grammars with no multiple entries in the table, we can use the table to produce one parse tree for each valid sentence. We call these grammars $LL(1)$: Left to right parse, Left-most derivation, 1 symbol lookahead.

A recursive descent parser examines input left to right. The order it expands non-terminals is leftmost first, and it looks ahead 1 token.

**One more time**

Balanced parentheses grammar 1:

$S \rightarrow ( \ S \ ) \mid SS \mid \epsilon$

1. Augment the grammar with EOF/$$
2. Construct Nullable, First and Follow
3. Build the predictive parse table, what happens?

**One more time, but this time with feeling **

Balanced parentheses grammar 2:

$S \rightarrow ( \ S \ )S \mid \epsilon$

1. Augment the grammar with EOF/$$
2. Construct Nullable, First and Follow
3. Build the predictive parse table
4. Using the predictive parse table, construct the parse tree for

$( ( ) )$ $\rightarrow$ $\epsilon$
and

$( ) ( )$ $\rightarrow$ $\epsilon$