Table Driven Predictive Parsing

<table>
<thead>
<tr>
<th>Non Terminal</th>
<th>Input Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>+</td>
</tr>
<tr>
<td>E</td>
<td>E-&gt;TP</td>
</tr>
<tr>
<td>P</td>
<td>P-&gt;+TP</td>
</tr>
<tr>
<td>T</td>
<td>T-&gt;FU</td>
</tr>
<tr>
<td>U</td>
<td>U-&gt;ε</td>
</tr>
<tr>
<td>F</td>
<td>F-&gt;id</td>
</tr>
</tbody>
</table>

Use parse table to derive: \( a^*(a+a)^a $\) ($ indicates end)

Notice we have only ONE choice for each cell in the table.

Rules for First Sets

- If X is a terminal then First(X) is just X
- If there is a Production \( X \rightarrow \varepsilon \) then add \( \varepsilon \) to First(X)
- If there is a Production \( X \rightarrow Y_1Y_2..Y_k \) then add First(Y1Y2..Yk) to first(X)
- First(Y1Y2..Yk) is either
  - First(Y1) (if First(Y1) doesn't contain \( \varepsilon \))
  - OR (if First(Y1) does contain \( \varepsilon \)) then First(Y1Y2..Yk) is everything in First(Y1) (except for \( \varepsilon \)) as well as everything in First(Y2..Yk)
  - If First(Y1) First(Y2)..<First(Yk) all contain \( \varepsilon \) then add \( \varepsilon \) to First(Y1Y2..Yk) as well.
Rules for Follow Sets

• First put $ (the end of input marker) in Follow(S) (S is the start symbol)
• If there is a production $A \rightarrow aBb$, (where a and b can be whole strings) then everything in FIRST(b) except for $\varepsilon$ is placed in FOLLOW(B).
• If there is a production $A \rightarrow aB$, then put everything in FOLLOW(A) in FOLLOW(B)
• If there is a production $A \rightarrow aBb$, where FIRST(b) contains $\varepsilon$, then put everything in FOLLOW(A) in FOLLOW(B)

Parse Table construction

M[A,a] refers to table entry at row non-terminal A and column terminal a.

For each production $A \rightarrow \alpha$ do
1. For each terminal a in first(\alpha) Add $A \rightarrow \alpha$ to M[A,a]
2. if $\varepsilon$ in first(\alpha) add $A \rightarrow \alpha$ to M[A,b] for each terminal b in follow(A)
3. if $\varepsilon$ in first(\alpha) and $\$$ in follow(A), Add $A \rightarrow \alpha$ to M[A,$]$
4. Set each undefined entry of M to error
Example: Create first and follow sets for the following grammar

1. \( E \rightarrow TE' \)
2. \( E' \rightarrow +E \)
3. \( E' \rightarrow -E \)
4. \( E' \rightarrow \varepsilon \)
5. \( T \rightarrow FT' \)
6. \( T' \rightarrow *T \)
7. \( T' \rightarrow /T \)
8. \( T' \rightarrow \varepsilon \)
9. \( F \rightarrow \text{num} \)
10. \( F \rightarrow \text{id} \)

<table>
<thead>
<tr>
<th></th>
<th>first</th>
<th>follow</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E )</td>
<td>( E )</td>
<td>( )</td>
</tr>
<tr>
<td>( E' )</td>
<td>( E' )</td>
<td>( )</td>
</tr>
<tr>
<td>( T )</td>
<td>( F )</td>
<td>( )</td>
</tr>
<tr>
<td>( T' )</td>
<td>( F )</td>
<td>( )</td>
</tr>
</tbody>
</table>

Create parse table – by inserting correct production to use in each cell. Nothing in a cell indicates an error.

<table>
<thead>
<tr>
<th></th>
<th>id</th>
<th>num</th>
<th>+</th>
<th>-</th>
<th>*</th>
<th>/</th>
<th>$</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
</tr>
<tr>
<td>( E' )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
</tr>
<tr>
<td>( T )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
</tr>
<tr>
<td>( T' )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
</tr>
<tr>
<td>( F )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
<td>( )</td>
</tr>
</tbody>
</table>
Two grammar problems – for top down parsing

1. Eliminating left recursion
   (without changing associativity)

   \[ E \rightarrow E + a \mid a \]

   I can’t tell which rule to use \( E \rightarrow E + a \) or \( E \rightarrow a \) as both generate same first symbol

2. removing two rules with same prefix

   \[ E \rightarrow aB \mid aC \]

   I can’t tell which rule to use as both generate same symbol

Removing Left Recursion

Before
- \( A \rightarrow A \ x \)
- \( A \rightarrow y \)

After
- \( A \rightarrow yA' \)
- \( A' \rightarrow x \ A' \)
- \( A' \rightarrow \epsilon \)