CS480
Translators

Top-down Parsing
Chap. 4
Milestone 3

• What is the purpose of this milestone?
• What does this mean?
  – The parser produces a list encoding the input.
• What is accepted/not accepted by the grammar thus far? Why/Why not?
  ()
  (())
  assign b 2
  (1 2 3)
  (1) (2)
  (1) ()
Example Input/Output

```
Testing stutest0.in:
-----------------------------------------------------
Result:
-----------------------------------------------------
Testing stutest1a.in:
()
-----------------------------------------------------
Result:
stutest1a.in:1,1: syntax error near '(': missing ')' at end of file
-----------------------------------------------------
Testing stutest1b.in:
)
-----------------------------------------------------
Result:
stutest1b.in:1,1: syntax error near '): extra ')' 
```
Example Input/Output

Testing stutest6d.in:
(((1 2))

Result:
(
  (1
   2)
)

Testing stutest6e.in:
(1(2))

Result:
(
  1
   (2
     2)
)
Defining an LL Grammar

- Need two definitions:
- **First** and **Follow**

![Diagram](image)

**Figure 4.15**: Terminal $c$ is in $\text{FIRST}(A)$ and $a$ is in $\text{FOLLOW}(A)$
First(\(\alpha\))

- If \(\alpha\) is any string in grammar, First(\(\alpha\)) is set of terminals that begin strings derived from \(\alpha\).
- If \(\alpha \Rightarrow \epsilon\), then \(\epsilon\) is in First(\(\alpha\)).
- What does it mean if \(A \rightarrow \alpha | \beta\), and First(\(\alpha\)) and First(\(\beta\)) are disjoint?
Follow(A)

• If $A$ is a nonterminal in grammar, Follow($A$) is set of terminals that can appear immediately to the right of $A$.

• If $A$ can be the rightmost symbol, then $\$ \text{ is in Follow}(A)$.

• What is $\$?$
Compute First(X) for all symbols

- If X is a terminal, then First(X) = \{X\}
- If X is a nonterminal and X \rightarrow Y_1Y_2...Y_k is a production, then place a in First(X) if for some i, a is in First(Y_i) and \epsilon is in all First(Y_1), ..., First(Y_{i-1}). If \epsilon is in First(Y_j) for all j = 1, 2, ..., k, then add \epsilon First(X).
- If X \rightarrow \epsilon, then \epsilon is in First(X)
Compute Follow(A) for nonterminals

• Place $ in Follow(S), where S is start symbol and $ is the input endmarker.
• If there is a production $A \rightarrow \alpha B\beta$, then everything in First($\beta$), except $\epsilon$, is in Follow($B$).
• If there is a production $A \rightarrow \alpha B$ or $A \rightarrow \alpha B\beta$, where $\epsilon$ is in First($\beta$), then everything in Follow($A$) is Follow($B$).
Example First and Follow

\[ E \rightarrow T \ E' \]
\[ E' \rightarrow + \ T \ E' \mid \varepsilon \]
\[ T \rightarrow F \ T' \]
\[ T' \rightarrow * \ F \ T' \mid \varepsilon \]
\[ F \rightarrow ( \ E ) \mid \text{id} \]

- First(E), First(E’), First(T), First(T’), First(F)?
- Follow(E), Follow(E’), Follow(T), Follow(T’), Follow(F)?
Predictive Parsing Table

- For each production A->α in the grammar:
  - For each terminal a in First(α), add A->α to M[A, a]
  - If ε is in First(α), then for each terminal b in Follow(A), add A->α to M[A, b]. If $ is in Follow(A), add A->α to M[A, $] as well

<table>
<thead>
<tr>
<th>NON-TERMINAL</th>
<th>id</th>
<th>+</th>
<th>*</th>
<th>(</th>
<th>)</th>
<th>$</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>$E \rightarrow TE'$</td>
<td>$E' \rightarrow +TE'$</td>
<td>$E' \rightarrow \epsilon$</td>
<td>$E \rightarrow TE'$</td>
<td>$E' \rightarrow \epsilon$</td>
<td>$E' \rightarrow \epsilon$</td>
</tr>
<tr>
<td>E'</td>
<td>$T \rightarrow FT'$</td>
<td>$T' \rightarrow \epsilon$</td>
<td>$T' \rightarrow *FT'$</td>
<td>$T \rightarrow FT'$</td>
<td>$T' \rightarrow \epsilon$</td>
<td>$T' \rightarrow \epsilon$</td>
</tr>
<tr>
<td>T</td>
<td>$F \rightarrow \text{id}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T'</td>
<td>$F \rightarrow \text{id}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Quiz #6

• For each grammar below, calculate First and Follow sets for each nonterminal and construct a parsing table.

(a) \( S \rightarrow 0 S' \)
\( S' \rightarrow S \ 1 \mid 1 \)

(b) \( S \rightarrow ( \ S \ ) \ S \mid \epsilon \)

• What do we need to do to our grammar to use top-down parsing? Is it LL(1)?