FIRST & FOLLOW

The construction of a predictive parser is aided by two functions associated with a grammar G. These functions, FIRST and FOLLOW, allow us to fill in the entries of a predictive parsing table for G, whenever possible. Sets of tokens yielded by the FOLLOW function can also be used as synchronizing tokens during panic-mode error recovery.

FIRST(α)

If α is any string of grammar symbols, let FIRST(α) be the set of terminals that begin the strings derived from α. If α ⇒ ε then ε is also in FIRST(α).

To compute FIRST(X) for all grammar symbols X, apply the following rules until no more terminals or ε can be added to any FIRST set:

1. If X is terminal, then FIRST(X) is {X}.
2. If X → ε is a production, then add ε to FIRST(X).
3. If X is nonterminal and X → Y₁ Y₂ ... Yₖ is a production, then place a in FIRST(X) if for some i, a is in FIRST(Yᵢ), and ε is in all of FIRST(Y₁), ..., FIRST(Yᵢ₋₁); that is, Y₁, ..., Yᵢ₋₁ ⇒ ε. If ε is in FIRST(Yⱼ) for all j = 1, 2, ..., k, then add ε to FIRST(X). For example, everything in FIRST(Y₁) is surely in FIRST(X). If Y₁ does not derive ε, then we add nothing more to FIRST(X), but if Y₁ ⇒ ε, then we add FIRST(Y₂) and so on.

Now, we can compute FIRST for any string X₁X₂ ... Xₙ as follows. Add to FIRST(X₁X₂ ... Xₙ) all the non-ε symbols of FIRST(X₁). Also add the non-ε symbols of FIRST(X₂) if ε is in FIRST(X₁), the non-ε symbols of FIRST(X₃) if ε is in both FIRST(X₁) and FIRST(X₂), and so on. Finally, add ε to FIRST(X₁X₂ ... Xₙ) if, for all i, FIRST(Xᵢ) contains ε.

FOLLOW(A)

Define FOLLOW(A), for nonterminal A, to be the set of terminals a that can appear immediately to the right of A in some sentential form, that is, the set of terminals a such that there exists a derivation of the form S⇒αAαβ for some α and β. Note that there may, at some time during the derivation, have been symbols between A and a, but if so, they derived ε and disappeared. If A can be the rightmost symbol in some sentential form, then $, representing the input right endmarker, is in FOLLOW(A).

To compute FOLLOW(A) for all nonterminals A, apply the following rules until nothing can be added to any FOLLOW set:

1. Place $ in FOLLOW(S), where S is the start symbol and $ is the input right endmarker.
2. If there is a production A ⇒ αBβ, then everything in FIRST(β), except for ε, is placed in FOLLOW(B).
3. If there is a production A ⇒ αB, or a production A ⇒ αBβ where FIRST(β) contains ε (i.e., β ⇒ ε), then everything in FOLLOW(A) is in FOLLOW(B).
EXAMPLE

Consider the expression grammar (4.11), repeated below:

\[
\begin{align*}
E & \rightarrow T \ E' \\
E' & \rightarrow + \ T \ E' \mid \epsilon \\
T & \rightarrow F \ T' \\
T' & \rightarrow * \ F \ T' \mid \epsilon \\
F & \rightarrow ( \ E ) \mid \text{id} \\
\end{align*}
\]

Then:

\[
\begin{align*}
\text{FIRST}(E) & = \text{FIRST}(T) = \text{FIRST}(F) = \{( , \text{id}\} \\
\text{FIRST}(E') & = \{+, \epsilon\} \\
\text{FIRST}(T') & = \{*, \epsilon\} \\
\text{FOLLOW}(E) & = \text{FOLLOW}(E') = \{) , \$\} \\
\text{FOLLOW}(T) & = \text{FOLLOW}(T') = \{+, ) , \$\} \\
\text{FOLLOW}(F) & = \{+, *, ) , \$\}
\end{align*}
\]