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 $\rightarrow$ ε is a production, then add ε to FIRST(X).

3. If X is nonterminal and X $\rightarrow Y_1 Y_2 \ldots Y_k$. is a production, then place *a* in FIRST(X) if for some *i*, *a* is in FIRST(Y_i), and ε is in all of FIRST(Y_1), ... , FIRST(Y_{i-1}); that is, $Y_1, \ldots, Y_{i-1}$ ε. If ε is in FIRST(Y_j) for all *j* = 1, 2, ... , *k*, then add ε to FIRST(X). For example, everything in FIRST(Y_1) is surely in FIRST(X). If Y_1 does not derive ε, then we add nothing more to FIRST(X), but if Y_1 ε, then we add FIRST(Y_2) and so on.

Now, we can compute FIRST for any string $X_1X_2 \ldots X_n$ as follows. Add to FIRST($X_1X_2 \ldots X_n$) all the non-ε symbols of FIRST(X_1). Also add the non-ε symbols of FIRST(X_2) if ε is in FIRST(X_1), the non-ε symbols of FIRST(X_3) if ε is in both FIRST(X_1) and FIRST(X_2), and so on. Finally, add ε to FIRST($X_1X_2 \ldots X_n$) if, for all *i*, FIRST(X_i) 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 αA $a\beta$ 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 $\alpha B\beta$, 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 $\alpha B\beta$ 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:

E → T E'
E'→ + T E' | ε
T → F T'
T' → * F T' | ε
F → (E) | **id**

Then:

FIRST(E) = FIRST(T) = FIRST(F) = {(, **id**}
FIRST(E') = {+, ε}
FIRST(T') = {*, ε}

FOLLOW(E) = FOLLOW(E') = {) , $}
FOLLOW(T) = FOLLOW(T') = {+,), $}
FOLLOW(F) = {+, *,), $}