

• HOME • UP •

Top/Top-down parsing/FOLLOW(x)

8.3. FOLLOW(N)

Definition of FOLLOW(N)

Let G be a grammar with start nonterminal S .

A *sentential form* of G is a string α where $S \Rightarrow^* \alpha$. That is, it is a string that can be derived from the start nonterminal using the productions of G .

Let N be a nonterminal of G . FOLLOW(N) is a set of tokens, possibly additionally including special symbol $\$$ indicating end-of-input.

Definition. FOLLOW(N) contains all tokens t so that there exists a sentential form that contains substring Nt .

That is, t is in FOLLOW(N) provided t can follow N in a derivation.

Additionally, FOLLOW(N) contains $\$$ if N there exists a sentential form that ends on N .

Example

Let's use the left-factored expression grammar, with start nonterminal E .

$$E \rightarrow TR$$

$$R \rightarrow \epsilon$$

$$R \rightarrow +E$$

$$T \rightarrow FS$$

$$S \rightarrow \epsilon$$

$$S \rightarrow *T$$

$$F \rightarrow n$$

$$F \rightarrow (E)$$

$$\begin{aligned} E &\rightarrow TR \\ &\rightarrow T\epsilon \\ &\rightarrow T+E \\ &\rightarrow T+^* \\ &\rightarrow FS+^* \\ &\rightarrow F+^* \\ &\rightarrow F*T+^* \\ &\rightarrow F*^*+^* \\ &\rightarrow (E)*^*+^* \end{aligned}$$

Here are the FOLLOW sets.

N	FOLLOW(N)
E	$\{), \$\}$
R	$\{), \$\}$
T	$\{+,), \$\}$

S	$\{+,), \$\}$
F	$\{+, *,), \$\}$

1. FOLLOW(E) must contain $\$$ because E is the start symbol, so E occurs at the end of sentential form E .
2. FOLLOW(E) also contains $)$ because the derivation

$$E \Rightarrow T R \Rightarrow F S R \Rightarrow (E) S R$$

yields a sentential form where E is followed by $)$.

3. You can see from the derivation above that FOLLOW(R) contains $\$$ (the sentential form $T R$ shows that).
4. FOLLOW(R) contains $)$ because the derivation above can be continued as follows.

$$(E) S R \Rightarrow (T R) S R$$

5. Derivation

$$E \Rightarrow T R \Rightarrow T$$

shows that FOLLOW(T) contains $\$$ and derivation

$$E \Rightarrow T R \Rightarrow T + E$$

shows that FOLLOW(T) contains $+$.

You should be able to derive the other FOLLOW sets. Note that FOLLOW(T) does not contain $*$. Nonterminal S is the only one that derives a $*$. But S can only be derived with F in front of it, and F cannot be erased.

An algorithm to compute the FOLLOW sets

Start with FOLLOW(N) = $\{\}$ for every nonterminal N . Then perform the following steps until none of the FOLLOW sets can be enlarged any more.

1. Add $\$$ to FOLLOW(S), where S is the start nonterminal.
2. If there is a production $A \rightarrow \alpha B \beta$, then add every token that is in FIRST(β) to FOLLOW(B). (Do not add ϵ to FOLLOW(B).
3. If there is a production $A \rightarrow \alpha B$, then add all members of FOLLOW(A) to FOLLOW(B).

(If t can follow A , then there must be a sentential form $\beta A t \gamma$. Using production $A \rightarrow \alpha B$ gives sentential form $\beta \alpha B t \gamma$, where B is followed by t .)

4. If there is a production $A \rightarrow \alpha B \beta$ where FIRST(β) contains ϵ , then add all members of FOLLOW(A) to FOLLOW(B).

(Reasoning is like rule 3. Just erase β .)

Example computation of FOLLOW sets

Let's use the algorithm to compute FOLLOW sets for the left-factored expression grammar. Like the algorithm for FIRST sets, this algorithm goes in phases, where the FOLLOW sets are enlarged until they cannot be enlarged any more.

Starting with empty sets and using rule (1) gives

N	E	T	F	R	S
FOLLOW(N)	{ $\$$ }	{ }	{ }	{ }	{ }

Now let's employ rule (2).

1. Production $F \rightarrow (E)$ tells us to add $)$ to FOLLOW(E), by rule (2).
2. Since FIRST(R) contains $+$, production $E \rightarrow T R$ tells us to add $+$ to FOLLOW(T).
3. Since FIRST(S) contains $*$, production $T \rightarrow F S$ tells us to add $*$ to FOLLOW(F).

N	E	T	F	R	S
FOLLOW(N)	{ $\}$, $\$$ }	{ $+$ }	{ $*$ }	{ }	{ }

Production $E \rightarrow T R$ tells us, by rules (3) and (4), to add all members of FOLLOW(E) to FOLLOW(R) and FOLLOW(T) (since FIRST(R) contains ϵ).

N	E	T	F	R	S
FOLLOW(N)	{ $\}$, $\$$ }	{ $+$, $\}$, $\$$ }	{ $*$ }	{ $\}$, $\$$ }	{ }

Now production $T \rightarrow F S$ tells us to add all members of FOLLOW(T) to FOLLOW(S) and FOLLOW(F) (since S can be erased).

N	E	T	F	R	S
FOLLOW(N)	{ $\}$, $\$$ }	{ $+$, $\}$, $\$$ }	{ $+$, $*$, $\}$, $\$$ }	{ $\}$, $\$$ }	{ $+$, $\}$, $\$$ }

Using the FOLLOW sets to build the parsing table

The FOLLOW sets cause additional members to be added to the top-down parsing table.

For each production $N \rightarrow \alpha$,
 if FIRST(α) contains ϵ ,
 for every b in FOLLOW(N)
 add production $N \rightarrow \alpha$ to $D(N, b)$.

The reasoning for that goes like this. Since production $N \rightarrow \alpha$ can erase N , the lookahead might be a token that comes after N . So use $N \rightarrow \alpha$ when the lookahead is a token that can follow N .



• [HOME](#) • [UP](#) •



[Top/Top-down parsing/FOLLOW\(x\)](#)