innovate achieve lead

Predictive Parsing

BITS Pilani
Pilani Campus

Dr. Shashank Gupta
Assistant Professor
Department of Computer Science and Information Systems

Removal of Left Recursion

$$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid A\alpha_3 \mid - - - - \mid A\alpha_n \mid \beta_1 \mid \beta_2 \mid \beta_3 \mid - - - \mid \beta_m$$

$$A \rightarrow \beta_1 A' \mid \beta_2 A' \mid - - - - \mid \beta_m A'$$

$$A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid - - - \mid \alpha_n A' \mid \in$$

Example

Left Recursive Grammar

$$A \rightarrow AC \mid Aad \mid bd \mid c$$

Modified Grammar

$$A \rightarrow bd\,A' \mid c\,A'$$

$$A' \rightarrow C\,A' \mid ad\,A' \mid \in$$

Indirect Left Recursion

innovate achieve lead

Left Recursive Grammar

$$S \rightarrow Aa|b$$
$$A \rightarrow Ac|Sd|\in$$

$$S \rightarrow Aa|b$$
$$A \rightarrow Ac|Aad|bd|\in$$

Modified Grammar

$$S \rightarrow Aa|b$$
$$A \rightarrow bdA'|A'$$
$$A' \rightarrow cA'|adA'|\in$$

Left Factoring

It is the process of removing the common left factor that appears in two productions of the same non-terminal.

$$A \rightarrow \alpha\ \beta_1 | \alpha\ \beta_2$$

Removal of Left Factoring:

$$A \rightarrow \alpha\ A'$$

$$A' \rightarrow \beta_1 | \beta_2$$

Example of Left Factoring

$$stmt \rightarrow if\ \exp\ then\ stmt$$
$$| if\ \exp\ then\ stmt\ else\ stmt$$

LEFT FACTORED GRAMMAR

$$stmt \rightarrow if\ \exp\ then\ stmt\ \ stmts$$
$$stmts \rightarrow \in | else\ \ stmt$$

Follow Set

Consider the following Grammar

$$A \rightarrow aBb$$

$$B \rightarrow c \mid \in$$

and suppose the input string is “ab” to parse.

Follow Set

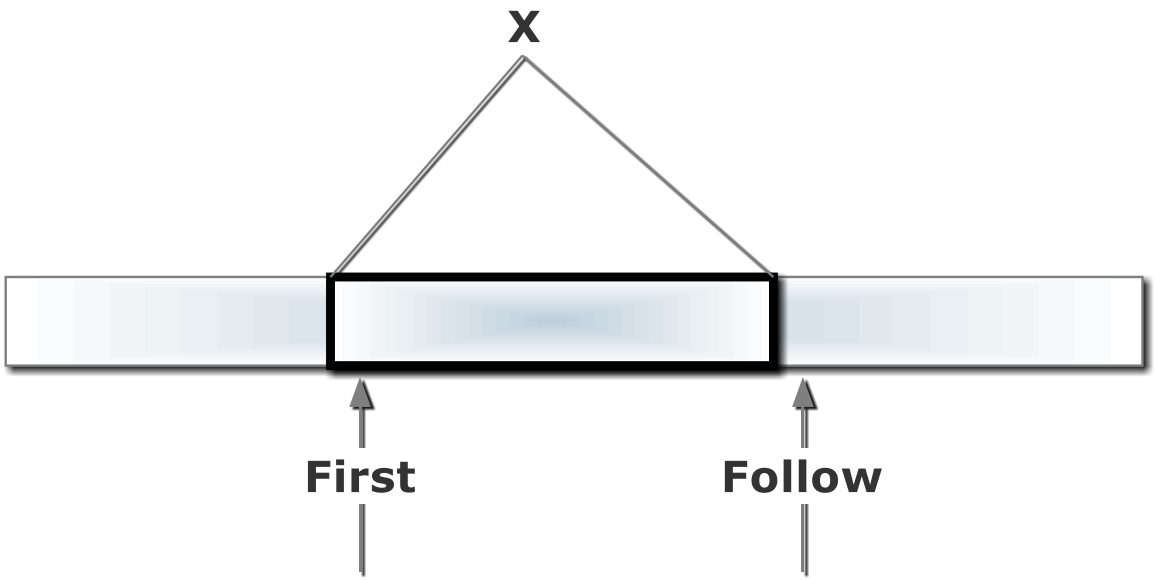
In RHS of A -> aBb, b follows Non-Terminal B, i.e. FOLLOW(B) = {b}, and the current input character to be read is also b.

- Hence the parser applies this rule. And it is able to get the string “ab” from the given grammar.

Therefore, FOLLOW can make a Non-terminal to vanish out if needed to generate the string from the parse tree.

Follow Sets

Follow (X) for a non-terminal X is the set of symbols that might follow the derivation of X in an input stream.



Steps for Computation of Follow Sets

- Always include **$** in **follow(S)**.
- if there is a production **A → αBβ**
  then everything in **first(β) (except ε)** is in **follow(B)**
- if there is a production **A → αBβ** and **First(β)** contains **ε**
  then everything in **follow(A)** is in **follow(B)**
- if there is a production **A → αB**
  then everything in **follow(A)** is in **follow(B)**

Example

Calculate the Follow of all non-terminals in the following grammar.

$S \rightarrow ABCDE$

$A \rightarrow a \mid \in$

$B \rightarrow b \mid \in$

$C \rightarrow c$

$D \rightarrow d \mid \in$

$E \rightarrow e \mid \in$

Variables/Non Terminals	Follow
S	{$}
A	{b, c}
B	{c}
C	{d, e, $}
D	{e, $}
E	{$}

Follow A= First(BCDE) = First (B) - ϵ $\cup$ First (C) = {b,c}
Follow C= First(DE) = First (D) - ϵ $\cup$ First (E) - ϵ $\cup$ Follow (S) = {d, e, $}

Example

Calculate the Follow of all non-terminals in the following grammar.

$$S \rightarrow Bb \mid Cd$$
$$B \rightarrow aB \mid \in$$
$$C \rightarrow cC \mid \in$$

Variables/Non Terminals	Follow
S	{$}
B	{b}
C	{d}

More Examples

Calculate the Follow of all non-terminals in the following two grammars.

$$S \rightarrow i\,E\,t\,S\,S^{'} \mid a$$

$$S^{'} \rightarrow e\,S \mid \in$$

$$E \rightarrow b$$

$$S \rightarrow ACB \mid CbB \mid Ba$$

$$A \rightarrow da \mid BC$$

$$B \rightarrow g \mid \in$$

$$C \rightarrow h \mid \in$$

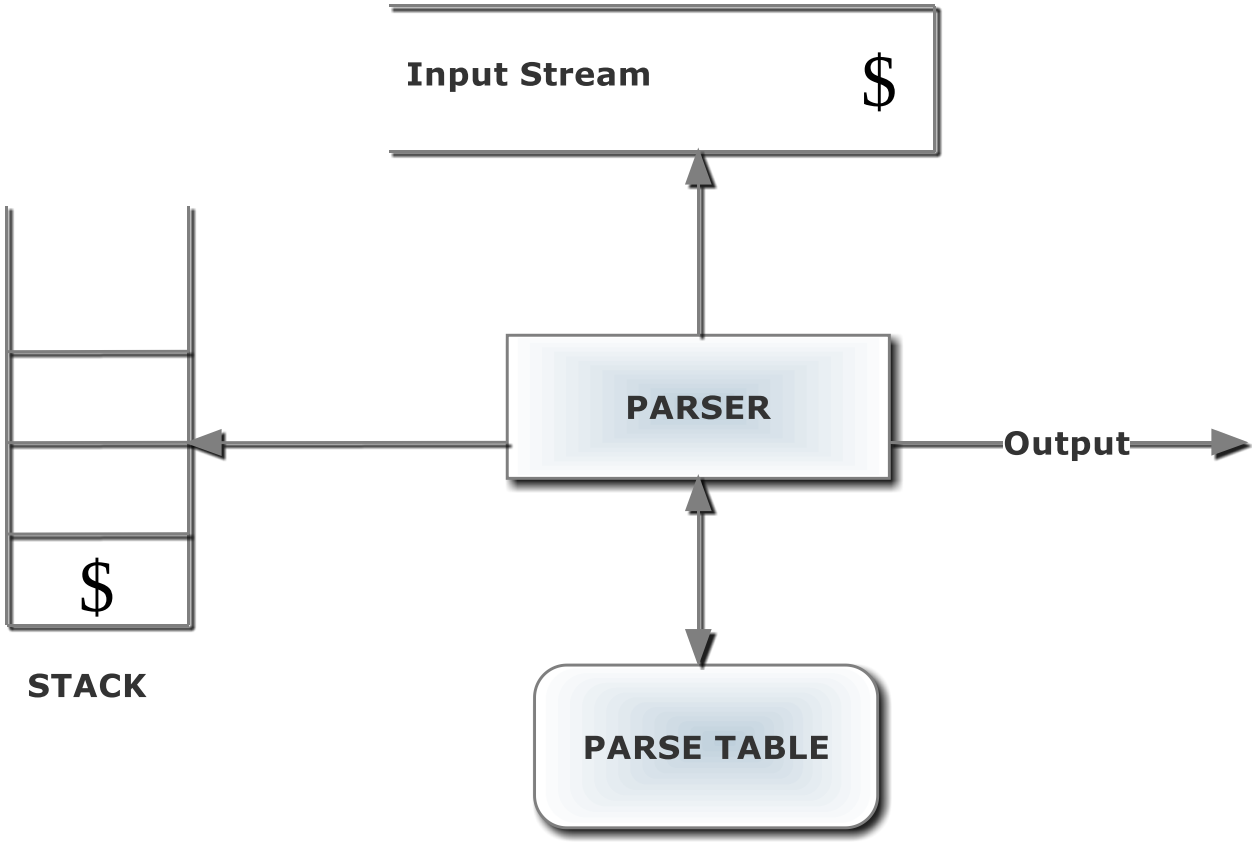
Predictive Parsing

A non-recursive top down parsing method.

Recognizes LL(1) languages.

- First ‘L’ means scanning of i/p stream from left to right.
- Second ‘L’ stands for left most derivation

Predictive Parser/LL(1) Parser



Parse Table is two dimensional array M[X, a] where 'X' is a non-terminal and 'a' is a terminal symbol of Grammar

Predictive Parser

Construct a predictive parser for the following grammar.

$$S \rightarrow (S) \mid \in$$

In addition, parse the following input stream of tokens

(( ))

Construction of First and Follow Sets

$$S \rightarrow (S) \mid \in$$

Non-Terminals	First	Follow
S	{ (, ε }	{ $,) }

Construction of Parse Table

$S \rightarrow (S) \mid \in$

Non-Terminals	First	Follow
S	{ (, ε }	{ $,) }

Terminals / Non-Terminals	(	)	$
S			

Terminals / Non-Terminals	(	)	$
S	$S \rightarrow (S)$		

Terminals / Non-Terminals	(	)	$
S	$S \rightarrow (S)$	$S \rightarrow \in$	$S \rightarrow \in$

LL(1) Parsing Table

$S \rightarrow (S) \mid \in$

Non-Terminals	First	Follow
S	{ (, ε }	{ $,) }

Terminals / Non-Terminals	(	)	$
S	$S \rightarrow (S)$	$S \rightarrow \in$	$S \rightarrow \in$