

EXPERIMENT No. 4

Aim: write a program to implement parser

Theory:

Parser is that phase of compiler which takes token string as input & with the help of the existing grammar, converts it into the corresponding parse tree. Parser is also known as syntax analyser.

There are two types of parser:

- 1) Top down parser
- 2) Bottom up parser

LL1 parser:

Here the first L represent that the scanning of the input will be done from left to right manner & the second L represent that the parsing technique we are going to use leftmost derivation tree. '1' represent the number of look-ahead, which means how many symbol we can see when you want to make a decision.

Algorithm to construct LL1 parsing table:

1. First check for left recursion in the grammar. If there is left recursion in the grammar, remove that & move to next step.
2. Calculate $first()$ & follow $()$ for all non-terminal.
 - $first()$: If there is a variable, & from that variable if we try to derive all strings then the beginning terminal symbol is called the $first()$.
 - $follow()$: The terminal symbol that follows a variable in the process of derivation.
3. For each production $A \rightarrow \alpha$, find $first(\alpha)$ & for each terminal in $first(\alpha)$, make entry $A \rightarrow \alpha$ in the table.

- If $\text{first}(\alpha)$ contains ϵ as terminal then find $\text{follow}(A)$ & for each terminal in $\text{follow}(A)$ make entry $A \rightarrow \alpha$ in the table.
- If $\text{first}(\alpha)$ contains ϵ & $\text{follow}(A)$ contains $\$$ as terminal then make entry $A \rightarrow \alpha$ in the table for $\$$.
- In the table, rows will contain the non-terminal & the column will contain the terminal. All production of the grammar will go under the follow element & the remaining production will lie under the elements of first set.

Example:

Grammar: $E \rightarrow TE' \mid E' \rightarrow +TE' \mid E$
 $T \rightarrow FT' \mid T' \rightarrow *FT' \mid E \mid F \rightarrow id \mid (CE)$

	first	follow
$E \rightarrow TE'$	$\{id, (, +\}$	$\{\$, \epsilon\}$
$E' \rightarrow +TE' \mid E$	$\{+, \epsilon\}$	$\{\$, \epsilon\}$
$T \rightarrow FT'$	$\{id, (, \epsilon\}$	$\{+, \$, \epsilon\}$
$T' \rightarrow *FT' \mid E$	$\{*, \epsilon\}$	$\{+, \$, \epsilon\}$
$F \rightarrow id \mid (CE)$	$\{id, (, \epsilon\}$	$\{*, +, \$, \epsilon\}$

LL(1) parsing table.

	$E \rightarrow TE'$	$E' \rightarrow +TE' \mid E$	$T \rightarrow FT'$	$T' \rightarrow *FT' \mid E$	$F \rightarrow id \mid (CE)$
E	$E \rightarrow TE'$	$E' \rightarrow E$			
E'	$E' \rightarrow +TE'$	$E' \rightarrow E$			
T	$T \rightarrow FT'$	$T' \rightarrow E$	$T \rightarrow FT'$	$T' \rightarrow E$	
F	$F \rightarrow id$		$T' \rightarrow E$	$T' \rightarrow E$	
T'			$F \rightarrow id$	$F \rightarrow (CE)$	

Δ

Δ