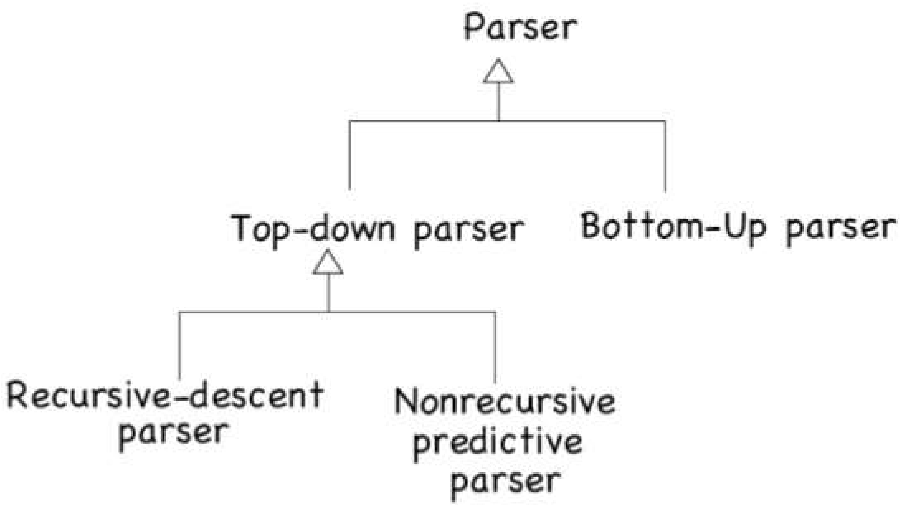
LL parser
Top Down Parsing

Relationship between parser types

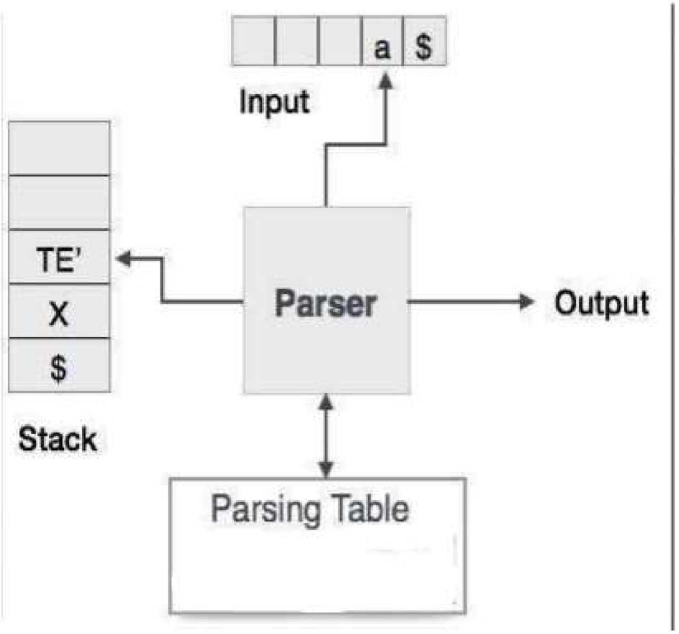


Predictive parser is a recursive descent parser, which has the capability to predict which production is to be used to replace the input string.

The predictive parser does not suffer from backtracking.

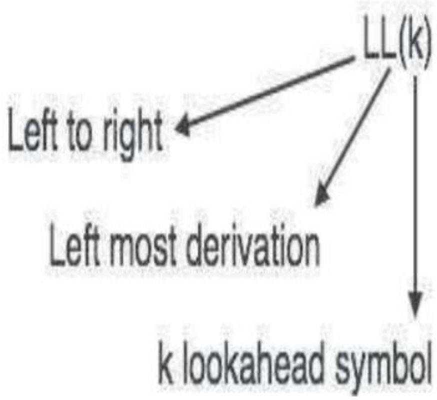
Predictive Parser

- Predictive parsing uses a stack and a parsing table to parse the input and generate a parse tree.
- Both the stack and the input contains an end symbol $to denote that the stack is empty and the input is consumed.
- The parser refers to the parsing table to take any decision on the input and stack element combination.



LL(k) Parser

- An LL parser is called an LL(k) parser if it uses *k* tokens of look ahead when parsing a sentence.
- LL grammars, particularly LL(1) grammars, as parsers are easy to construct, and many computer languages are designed to be LL(1) for this reason.
- The 1 stands for using one input symbol of look ahead at each step to make parsing action decision.



PRIME REQUIREMENT OF LL(1)

The grammar must be –

- no left factoring
- no left recursion

FIRST() & FOLLOW()

LOOKAHEAD symbols

Parsing Table

Using Stack Implementation

Parse Tree is constructed