Table of Contents

A parser that uses collection of recursive procedures for parsing the given input string is called Recursive descent (RD) parser.

Basic steps for construction of RD parser:

1. The R.H.S. of the rule is directly converted into program code symbol by symbol.
2. If the input is non-terminal then a call to the procedure corresponding to the non-terminal is made.
3. If the input is terminal then it is matched with the look ahead from input. The look-ahead pointer has to be advanced on matching of the input symbol.
4. If the production rule has many alternates then all these alternates has to be combine into a single body of procedure.
5. The parser should be activated by a procedure corresponding to the start symbol.

Predictive LL (1) parser

This top down parsing algorithm is of non- recursive type. In this type of parsing a table is built. for LL(1):

The first L means the input is scanned from left to right.

The second L means it uses left most derivation for input string. & the number 1 in the input symbol (look-ahead) to predict the parsing process.

The data structure used by LL(1)

- Input Buffer
- Stack
- Parsing table

Construction of Predictive LL (1) Parser

This parser is based on two very important function & those are FIRST and FOLLOW.

For construction of predictive LL(1) parser we have to follow the following steps:

1. Computation of FIRST and FOLLOW function.
2. Construct the predictive parsing table using FIRST and FOLLOW functions.
3. Parse the input string with the help of predictive parsing table.

FIRST function

Following are the rules used to compute the FIRST functions.

- If the terminal symbol a the FIRST (a) = {a}.
- If there is a rule X –> e then FIRST (X) = {e}.
- For the rule A–>X1 X2 X3 …..Xk FIRST (A) = (FIRST (X1) U FIRST (X2) U FIRST (X3) …. U FIRST (Xk).

Where K Xj <= n such that 1<= j <= k-1.

FOLLOW function

The rule of computing FOLLOW function are as given below:

- For the start symbol S place $ in FOLLOW(S).
- If there is a production A –> aBb then everything in FIRST(b) without e is to be placed in FOLLOW(B).
- If there is a production A –> aBb or A –> aB and FIRST (b) ={e} then FOLLOW (A) = FOLLOW(B) or FOLLOW(B) = FOLLOW (A). That means everything in FOLLOW (A) is in FOLLOW (B).