

S attribute and L attribute:

Synthesized attribute:

* It takes Value from child node.

* It follows the bottom up traverse method.

* It is in both S attribute and L attribute.

Inherited attributes:

* It takes Value from parent or sibling node.

* It follows the top down approach.

* It is only in L attribute.

S attribute:

* S attribute is defined as it is of synthesized attribute

L attribute:

* L attribute is defined as it is of both synthesized and inherited attribute.

eg:

$E \rightarrow E + T$

$\{E.val : E.val + T.val\}$

$E \rightarrow T$

$T \rightarrow T * F$

$T \rightarrow F$

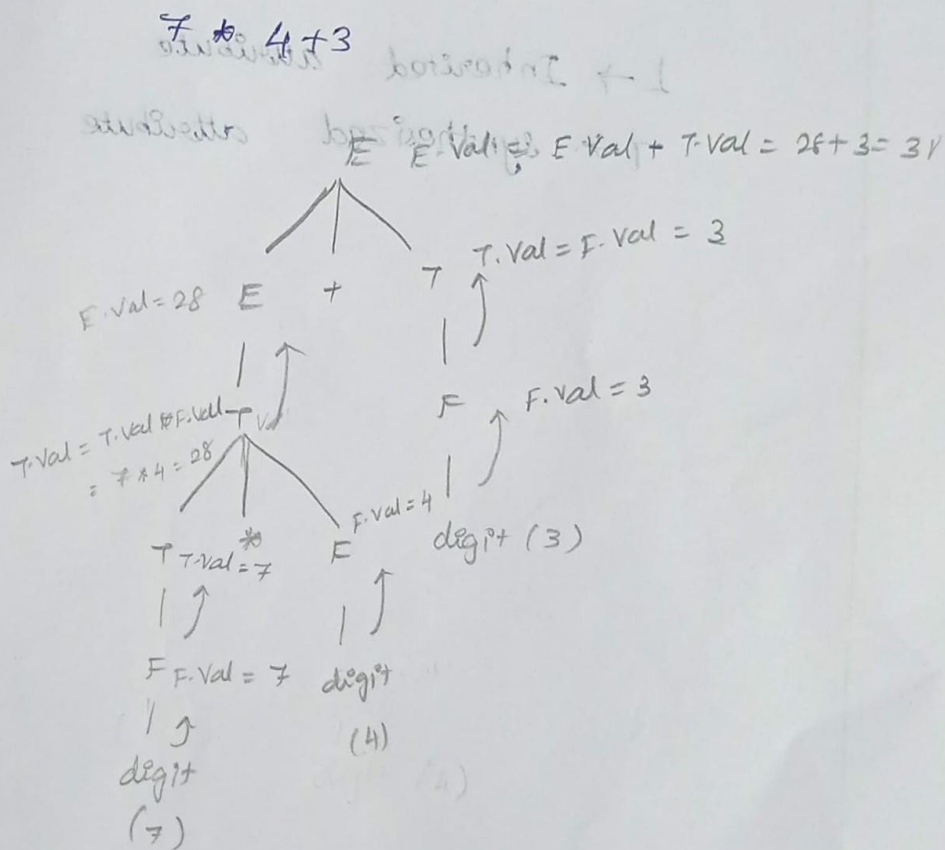
$F \rightarrow (E)$

$F \rightarrow digit$

$\{F.val = digit.lexval\}$

} write the semantic rules

bottom up - approach:



Inherited :

Top-down approach

$D \rightarrow T L$ (declaration \rightarrow type)

$L \rightarrow L, id$ (Lst inherit = T-Type)

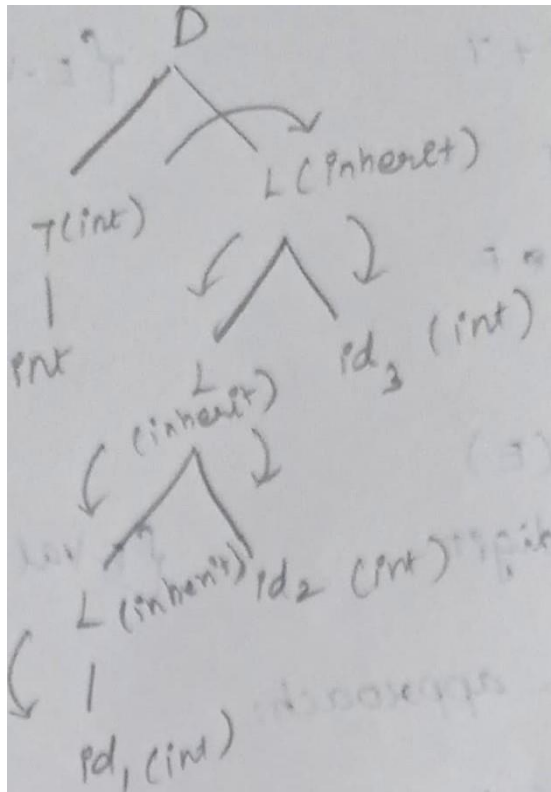
$T \rightarrow int$ (T-Type = int)

$T \rightarrow float$ (T.type = float)

$L \rightarrow L, id$ (L type (inherit))

$L \rightarrow id$ (add (L inherit))

add type (Lst inherit)



L-inherited attribute, T-synthesized attribute