CSA1455

Compiler Design for Lenical Analysis

CLASS TEST - 3

Name : Sai Lokesh. N

Reg.No : 192365023

Date : 18 - 02 - 2025

Serial.No: 27

QUESTIONS

1. Construct a syntan tree for the Arthimetic Enpression: (a+b) * (c-d)/e.

2. How does an S-attributed definition differ from an L-attributed definition? Construct the grammar for both L and S attribute.

3. Construct the annoated Parse tree for the following Enpression (2-3)* (4-2) and give the Syntan directed definition below:

$L \rightarrow E$

$E \rightarrow T$

$E \rightarrow E1 + T$

$T \rightarrow F$

$T \rightarrow T1 * F$

$F \rightarrow (E)$

$F \rightarrow digit$

Answers

1. operator Precedence and Associativity

1. Paranthesis

2. Multiplication

3. DIVISION

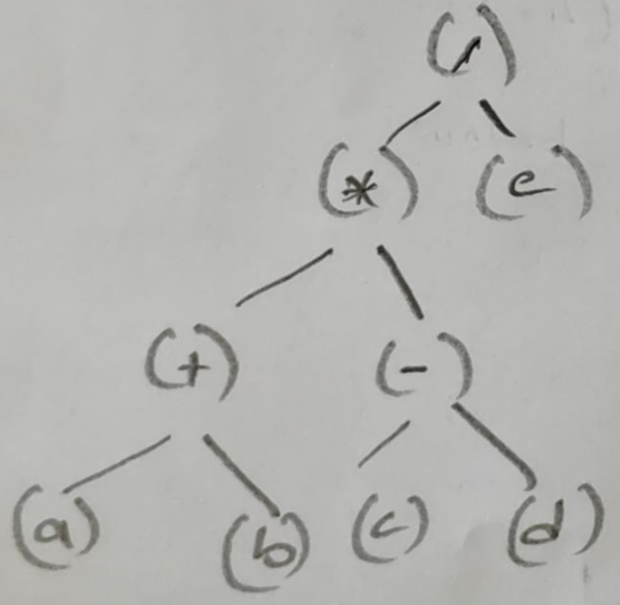
Syntan tree Construction

1. a + b and c - d form two subtrees.

2. These two Results are Multiplied.

3. The Multiplication Result is then divideb by e.

Syntan tree Representation.



2.

S - Attribute:

* It uses only synthesized attributes.
* Values are computed bottom-up in the Parse tree.
* Suitable for bottom-up Parsing
* Requires a Postorder traversal.
* Used in Evaluating Expressions where only synthesized values are Needed.

L- Attribute:

* Uses Both Synthesized and Inherited attributes.
* values are computed left to Right, allowing inherited attributes.
* Suitable for top-down Parsing.
* Requires a Pre-order traversal.

* Used in Semantic Rules like type checking, where inherited attributes influence children.

S-attributed Grammar:

E → E1 + T { E.val = E1.val + T.val }

E → T { E.val = T.val }

T → T1 * F { T.val = T1.val * F.val }

T → F { T.val = F.val }

F → (E) { F.val = E.val }

F → num { F.val = num.lexval }

L-Attributed Grammar;

E → T E'

E' → + T { E'.inh = E.inh + T.val }

E'

E' → ε { E.val = E.inh }

T → F T'

T' → * F { T'.inh = T.inh * F.val }

T'

T' → ε { T.val = T.inh }

F → (E) { F.val = E.val }

F → num { F.val = num.lexval }

3. Given Grammar:

E → E + T

E → E - T

E → T

T → T * F

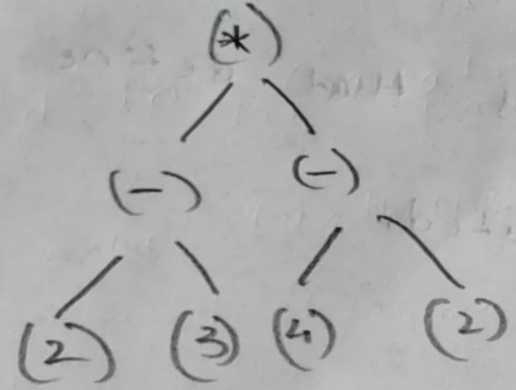
T → T|F

T → F

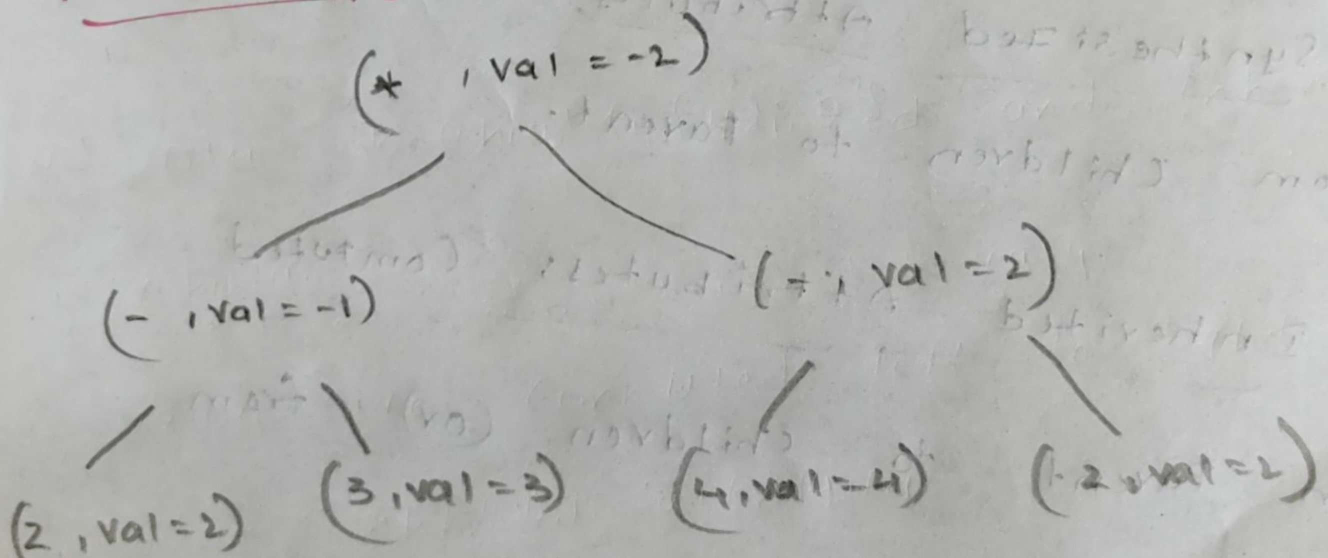
F → (E)

F → digit

Parse Tree for (2-3) * (4-2)



With Comptued values:



Step -by - step Evaluation:

1. 2 - 3 = -1

2. 4 - 2 = 2

3. (-1) * 2 = -2

So the final Result = -2

* A Syntan - directed Definition (SDD) is a Content free grammar augmented with semantic Rules that define how attributes are Computed.

Types of Attributes:

1. Synthesized Attributes: Computed from Children to Parent.

2. Inherited Attributes: Computed from Parent to children (or) from left siblings in a Parse tree.