CCA 3

21/4/25

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

SSCD Theory

Assignment 2

Krishnaraj PT.
PA15
1032210888

Q1. Explain types of SDD.

→ Syntax Directed definitions (SDD) are formal methods of attaching semantic information to the syntactic structure of a programing language.

→ They add a semantic rule set for every production of the gramar.

→ The rules described in these definitions state how to derive values such as types, memory locations, or fragments of a code from the structure of an input object.

Types

- S- Attributed
- L- ~~At~~ Attributed

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

(☆) S- attributed :

→ Attributes are synthesized
→ For evaluation, any bottom up sequence can be used
→ Evaluated in post order traversal
→ When traverses ~~so~~ crosses node N for the last time in a post order traversal, node N is evaluated.

(☆) L- Attributed

→ If the attributes of the nodes are synthesized or inherited
→ Parse tree can only be read Left to right

(Q.2) Develop an alg for construction of a DAG

→ Input : Basic block
→ Output : A DAG for the basic block containing the following Instructions
(1.) A label for each node. For leaves, the label is an identifier.
(2) For each node, a list of attached identifiers to hold the computed values.

Case i $x := y \ op \ z$
Case ii $x := op \ y$
Case iii $x := y$

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

Method :

step 1

(1) if y is undefined, then create node y.

(2) If z is undefined, create node z for call i.

step 2

1. For the call (i), create a node (op) whose left child is node y, and the right child is node (z). Let n be this node

2. For call (ii), determine if there is node (op) without child node. If ~~not~~ not, create such a node.

3. For call (iii) node n will be node (y)

step 3

Delete n ~~for~~ from the list of identifiers for node (n). Append n to list of attached identifiers for the node n; found in step 2 and set node(n) to n.

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

(Q.3) Build the Syntax directed definition for the following grammar. Draw the annotated parse tree for the input string 4 * 5 + 6 for computing synthesized attributes

$S \rightarrow E$

$E \rightarrow E1 + T$

$E \rightarrow T$

$T \rightarrow T1 * F$

$T \rightarrow F$

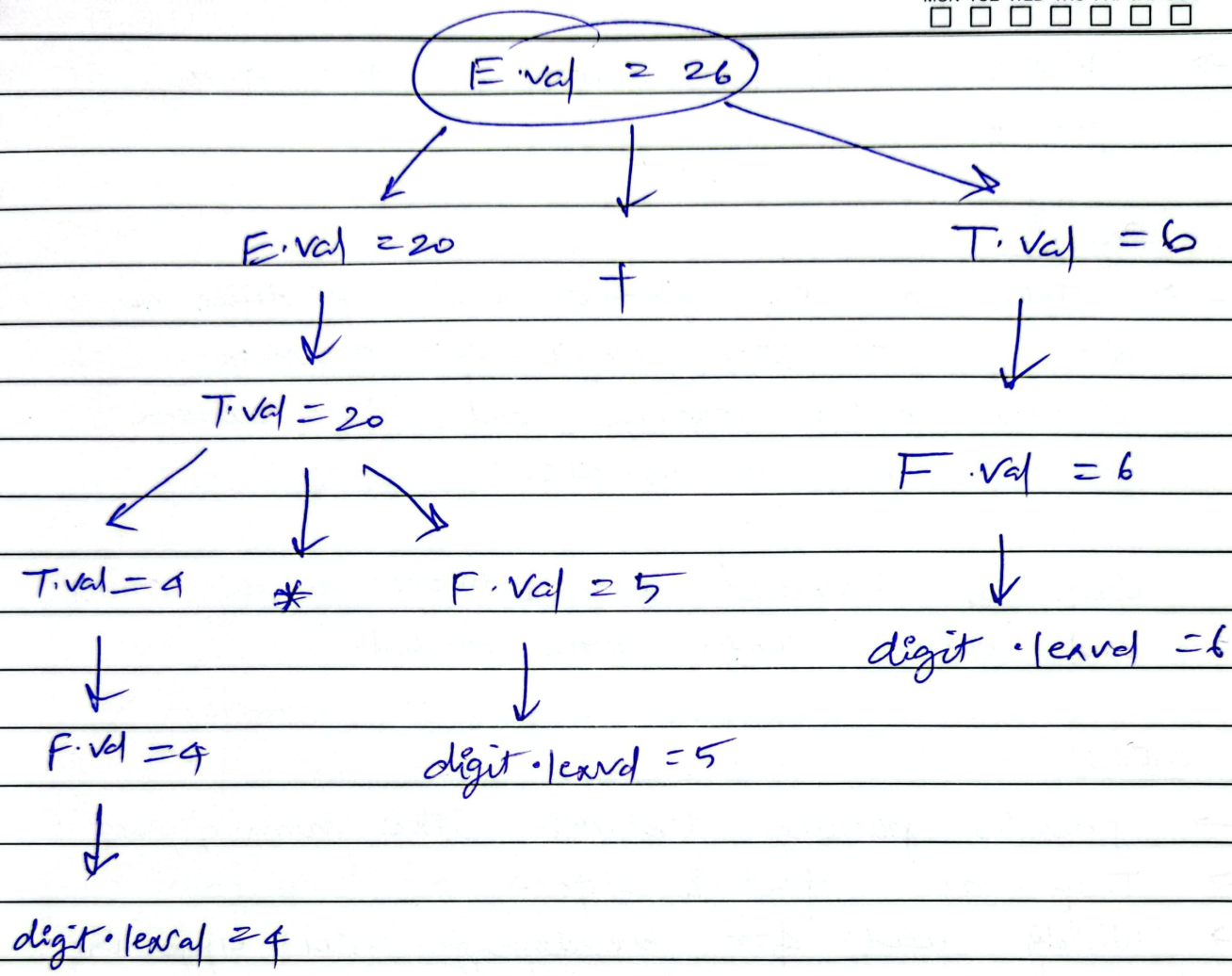
$F \rightarrow digit$

→ The SDD is :

Production	Semantic Rule
$S \rightarrow E$	$S.val \rightarrow E.val$
$E \rightarrow E1 + T$	$E.val = E1.val + T.val$
$E \rightarrow T$	$E.val = T.val$
$T \rightarrow T1 * F$	$T.val = T1.val * F.val$
$T \rightarrow F$	$T.val = F.val$
$F \rightarrow digit$	$F.val = digit.lexval.$

→ The annotated parse tree is as follows.

Date :.....
MON TUE WED THU FRI SAT SUN



Q.4 Explain the case studies of the different parsing tools like ANTLR, PEG.js, Rust.

(1) ANTLR :

→ it is a powerful parser generator that you can use to read, process, execute or translate structured text or binary files.

It uses CFG to define language syntax and can generate parsers in JAVA, C#, Python

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

→ Twitter search uses ANTLR for query passing.

(2) Peg.js

→ Takes a Peg grammar and produces a passer in JS. Its useful for parsing domain specific languages and other custom formats in web applications

→ Used in browsers, code editors when small and custom passers are needed.

(3) Rust

→ provides passing libraries like nom, pest.
→ They are fast & safe.
→ Widely used for developing cli apps in rust

(4) Real World applications

→ Rust compiler components.
→ Security tools like fire crackers.

(Q.5) List & Explain different types of transformations on basic blocks.

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

→ A number of transformations can be applied to a basic block

(*) Structure preserving :
1. Common expression elimination
2. Dead code elimination
3. Renaming temp variables
4. Interchange of statements.

(*) Algebraic transformations.

→ Common expression elimination
If an expr like a+b appears multiple times & the values of a & b have not changed the result can be reused.

```
x = a+b
y = a+b ;
```

→ Dead code elimination
If a computed value is never used in a computation, its considered dead and can be removed

```
x = 10
x = 20
```

Date :.....
MON TUE WED THU FRI SAT SUN
☐ ☐ ☐ ☐ ☐ ☐ ☐

→ Renaming temp variables

A statement $t = b + c$ can be changed to $a = b + c$ & all uses of this instance of + can be changed to 'a'. without changing the value of the basic block.

→ Interchange of statements.

$t_1 = b + c$
$t_2 = x + y$.

We can interchange the two statements without affecting the value if and only if neither x is t_2 or y, and neither b or c is t_1.

→ Algebraic tranformations

They can be used to change the set of expressions computed by a basic block into an algebraically equivalent set.
eg. $x = x + 0$ or $x = x * 1$ can be eliminated.

$x = y ** 2$ can be replaced by
$x = y * y$.