

## School of Computing

### CIA II Test – May 2022

Course Code: CSE402

Course Name: Compiler Engineering

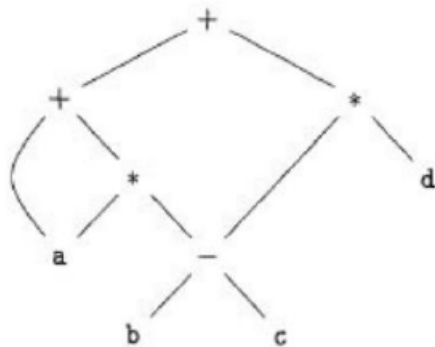
Duration: 90 minutes Max Marks: 50

#### PART A 10 x 2 = 20 Marks

- Parser cannot catch all the program errors. There is a level of correctness that is deeper than syntax analysis. ASDT is an associated snippet of code with each production which can the semantic while parsing the code.

	Production	Code Snippet
1	$Number \rightarrow Sign\ List$	$$$ \leftarrow \$1 \times \$2$
2	$Sign \rightarrow +$	$$$ \leftarrow 1$
3	$Sign \rightarrow -$	$$$ \leftarrow -1$
4	$List \rightarrow Bit$	$$$ \leftarrow \$1$
5	$List_0 \rightarrow List_1\ Bit$	$$$ \leftarrow 2 \times \$1 + \$2$
6	$Bit \rightarrow 0$	$$$ \leftarrow 0$
7	$Bit \rightarrow 1$	$$$ \leftarrow 1$

- 
- 
- Type of variable, the size of its runtime representation, the information needed to generate a runtime address and a number of type-dependent fields



- 
- 
- 
- 
- ```

a = 200;
b = x + y;
while(a > 0)
{
  if (a % b == 0)
  printf("%d", a);
}

```
- Statements are executed sequentially  
If any statement executes, the entire block executes, unless a runtime exception occurs
- $$a_0^3 \leftarrow x_0^1 + y_0^2$$

$$b_0^3 \leftarrow a_0^3$$

$$a_1^4 \leftarrow 17$$

$$c_0^3 \leftarrow a_0^3$$



Balance ( $t_4$ )

$q: \{ \}$

Rank( $t_4$ ) = ~~Rank~~  $\text{flatten}(a) + \text{flatten}(b)$

Rebuild( $q, t_4$ )

Rebuild:

$$t_4^2 \leftarrow b + a \quad \text{--- (I)}$$

$q: \{ \langle t_4, 2 \rangle \}$

$t_7 \in \text{Roots}$

Balance ( $t_7$ )

$q: \{ \}$

Rank( $t_7$ ) =  $\text{flatten}(a) + \text{flatten}(b)$

$q: \{ \langle b, 1 \rangle \langle a, 1 \rangle \}$

$$t_7^2 \leftarrow b + a \quad \text{--- (II)}$$

$q: \{ \langle t_7, 2 \rangle \langle t_4, 2 \rangle \}$

Rebuild( $q, -$ )

$$t_8^2 \leftarrow t_4^2 - t_7^2 \quad \text{--- (III)}$$

q: q

Roots:  $\{ \langle t_7, 1 \rangle, \langle t_6, 1 \rangle, \langle t_4, 1 \rangle \}$

Rank( $t_7$ ) > 0 return

Roots:  $\{ \langle t_6, 1 \rangle, \langle t_4, 1 \rangle \}$

var:  $\langle t_6, 1 \rangle$

Bal( $t_6$ )

q:  $\{ \}$

Rank( $t_6$ ) = flatten( $t_4$ ) + flatten( $t_6$ )

$t_4 \in \text{Roots}$

Bal( $t_4$ )

q:  $\{ \}$

Rank( $t_4$ ) = flatten

Rank( $t_4$ ) > 0

return 2

$\{ \langle t_4, 2 \rangle \}$

$\{ \langle t_6, 1 \rangle, \langle t_4, 2 \rangle \}$

q:  $\{ \langle t_6, 1 \rangle \}$

return 1.

$t_6 \leftarrow t_6 + t_4$  (IV)

Rebuild

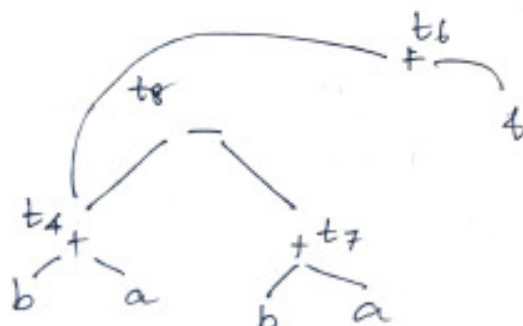
Roots =  $\{ \langle t_4, 1 \rangle \}$  return

$$t_4^2 \leftarrow b + a^1$$

$$t_7^2 \leftarrow b^1 + a$$

$$t_8^1 \leftarrow t_4^2 - t_7^2$$

$$t_6 \leftarrow b + t_4$$



If assuming t5 and t9 are members of LIVEOUT(b) then  
t5, and t9 also belong to Roots – then the solution will be different  
Balanced tree and new IR – 7 Marks

13 Finding UEVar and VARKill set – 3 Marks

Liveout iteration – 3 iteration – 7 Marks

| Iteration | LiveOut(n)  |             |             |             |             |
|-----------|-------------|-------------|-------------|-------------|-------------|
|           | B0          | B1          | B2          | B3          | B4          |
| Initial   | $\emptyset$ | $\emptyset$ | $\emptyset$ | $\emptyset$ | $\emptyset$ |
| 1         | $\{l\}$     | $\{s, l\}$  | $\{s, l\}$  | $\{s, l\}$  | $\emptyset$ |
| 2         | $\{s, l\}$  | $\{s, l\}$  | $\{s, l\}$  | $\{s, l\}$  | $\emptyset$ |
| 3         | $\{s, l\}$  | $\{s, l\}$  | $\{s, l\}$  | $\{s, l\}$  | $\emptyset$ |

14 Procedure placement explanation – 3 Marks

Steps to find the optimized list – 7 Marks

Optimized list is

**List(x)** {  $P_0, P_1, P_2, P_5, P_6, P_4, P_3$  }