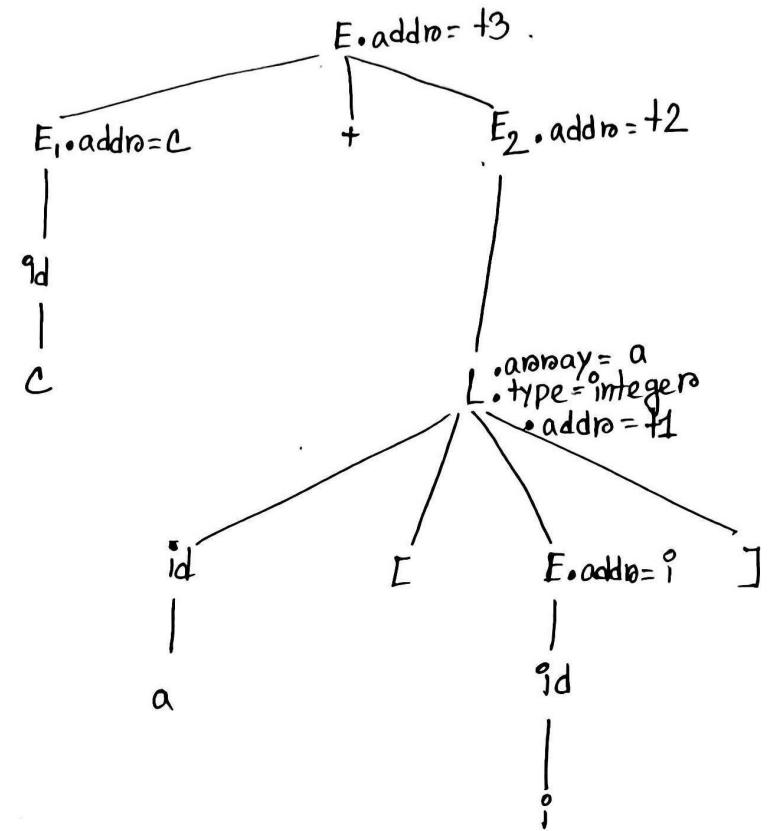


Example 01

$$\begin{aligned}
S &\rightarrow \mathbf{id} = E ; \quad \{ \text{gen}(\text{top.get}(\mathbf{id.lexeme}) \neq E.addr); \} \\
&\quad | \quad L = E ; \quad \{ \text{gen}(L.addr.base '[' L.addr '] \neq E.addr); \} \\
E &\rightarrow E_1 + E_2 \quad \{ E.addr = \mathbf{new Temp}(); \\
&\quad \text{gen}(E.addr \neq E_1.addr '+' E_2.addr); \} \\
&\quad | \quad \mathbf{id} \quad \{ E.addr = \text{top.get}(\mathbf{id.lexeme}); \} \\
&\quad | \quad L \quad \{ E.addr = \mathbf{new Temp}(); \\
&\quad \text{gen}(E.addr \neq L.array.base '[' L.addr ']'); \} \\
L &\rightarrow \mathbf{id} [E] \quad \{ L.array = \text{top.get}(\mathbf{id.lexeme}); \\
&\quad L.type = L.array.type.elem; \\
&\quad L.addr = \mathbf{new Temp}(); \\
&\quad \text{gen}(L.addr \neq E.addr '*' L.type.width); \} \\
&\quad | \quad L_1 [E] \quad \{ L.array = L_1.array; \\
&\quad L.type = L_1.type.elem; \\
&\quad t = \mathbf{new Temp}(); \\
&\quad L.addr = \mathbf{new Temp}(); \\
&\quad \text{gen}(t \neq E.addr '*' L.type.width); \} \\
&\quad \text{gen}(L.addr \neq L_1.addr '+' t); \}
\end{aligned}$$

Let's denote a one dimensional array of integers, and let c, i , all denote integers. Then, the type of a is $\text{array}(8, \text{integer})$. Its width w is 32, assuming that the width of an integer is 4. The type of $a[i]$ is integer.

Input String: c + a[i]



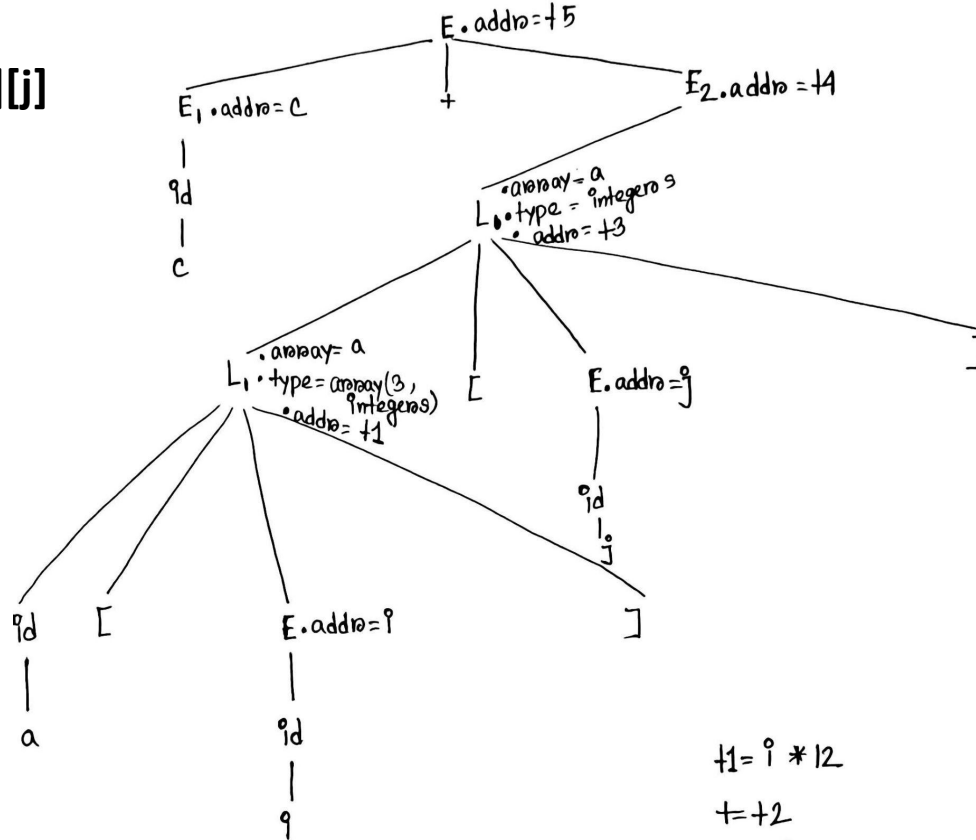
$+1 = i * 4$
 $+2 = a[+1]$
 $+3 = c + +2$

Example 02

$$\begin{array}{ll}
S \rightarrow \text{id} = E ; & \{ \text{gen}(top.get(\text{id.lexeme}) \neq E.addr); \} \\
| L = E ; & \{ \text{gen}(L.addr.base '[' L.addr '] \neq E.addr); \} \\
E \rightarrow E_1 + E_2 & \{ E.addr = \text{new Temp}(); \\
& \text{gen}(E.addr \neq E_1.addr '+' E_2.addr); \} \\
| \text{id} & \{ E.addr = top.get(\text{id.lexeme}); \} \\
| L & \{ E.addr = \text{new Temp}(); \\
& \text{gen}(E.addr \neq L.array.base '[' L.addr ']'); \} \\
L \rightarrow \text{id} [E] & \{ L.array = top.get(\text{id.lexeme}); \\
& L.type = L.array.type.elem; \\
& L.addr = \text{new Temp}(); \\
& \text{gen}(L.addr \neq E.addr '*' L.type.width); \} \\
| L_1 [E] & \{ L.array = L_1.array; \\
& L.type = L_1.type.elem; \\
& t = \text{new Temp}(); \\
& L.addr = \text{new Temp}(); \\
& \text{gen}(t \neq E.addr '*' L.type.width); \\
& \text{gen}(L.addr \neq L_1.addr '+' t); \}
\end{array}$$

Let's denote a 2 x 3 array of integers, and let c, i, and j all denote integers. Then, the type of a is array(2, array(3, integer)). Its width w is 24, assuming that the width of an integer is 4. The type of a[i] is array(3, integer), of width w = 12. The type of a[i][j] is integer.

Input String: $c + a[i][j]$



$$+1 = i * 12$$

$$+ = +2$$

$$+2 = j * 4$$

$$+3 = +1 + +2$$

$$+4 = a[+3]$$

$$+5 = c + +4$$

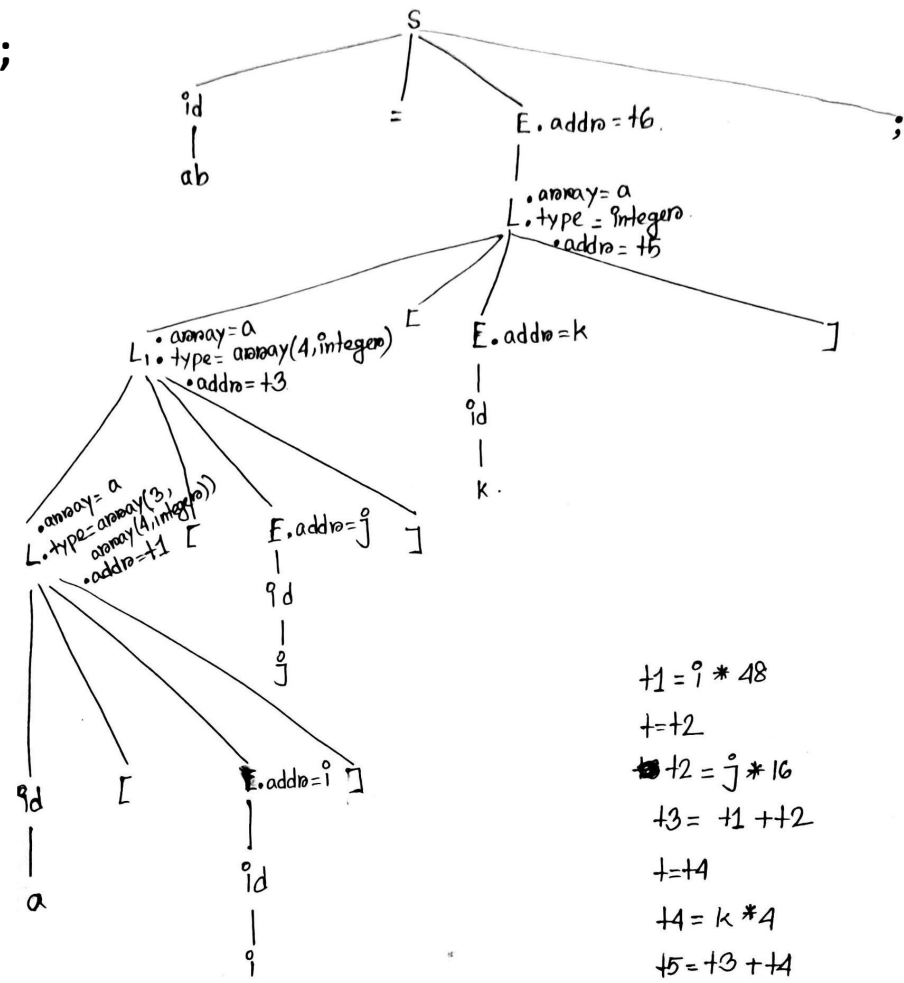
Prepared By: Sukarna Sarker

Example 03

$$\begin{array}{ll}
S \rightarrow \text{id} = E ; & \{ \text{gen}(top.get(\text{id.lexeme}) \neq E.addr); \} \\
| L = E ; & \{ \text{gen}(L.addr.base '[' L.addr '] \neq E.addr); \} \\
E \rightarrow E_1 + E_2 & \{ E.addr = \text{new Temp}(); \\
& \text{gen}(E.addr \neq E_1.addr '+' E_2.addr); \} \\
| \text{id} & \{ E.addr = top.get(\text{id.lexeme}); \} \\
| L & \{ E.addr = \text{new Temp}(); \\
& \text{gen}(E.addr \neq L.array.base '[' L.addr ']'); \} \\
L \rightarrow \text{id} [E] & \{ L.array = top.get(\text{id.lexeme}); \\
& L.type = L.array.type.elem; \\
& L.addr = \text{new Temp}(); \\
& \text{gen}(L.addr \neq E.addr '*' L.type.width); \} \\
| L_1 [E] & \{ L.array = L_1.array; \\
& L.type = L_1.type.elem; \\
& t = \text{new Temp}(); \\
& L.addr = \text{new Temp}(); \\
& \text{gen}(t \neq E.addr '*' L.type.width); \\
& \text{gen}(L.addr \neq L_1.addr '+' t); \}
\end{array}$$

Let's denote a 2 x 3 x 4 array of integers, and let ab, i, j and k all denote integers. Then, the type of a is array(2, array(3, array(4, integer))). Its width w is 96, assuming that the width of an integer is 4. The type of a[i] is array(3, array(4, integer)), of width w = 48. The type of a[i][j] is array(4, integer) of width = 16. The type of a[i][j][k] is integer.

Input String: `ab = a[i][j][k];`



$t1 = i * 48$
 $t = t2$
 $t2 = j * 16$
 $t3 = t1 + t2$
 $t = t4$
 $t4 = k * 4$
 $t5 = t3 + t4$
 $t6 = a[t5]$
 $ab = t6.$