$S \rightarrow id = E;$ {gen (top.get (id.lexeme) '=' E.addr);}

| $L = E;$ {gen (L.addr.base '[' L.addr ']' '=' E.addr);}

$E \rightarrow E_1 + E_2$ {E.addr = new Temp();
gen (E.addr '=' E_1.addr '+' E_2.addr);}

| id {E.addr = top.get (id.lexeme);}

| L {E.addr = new Temp();
gen (E.addr '=' L.array.base '[' L.addr ']');}

$L \rightarrow id\ [E]$ {L.array = top.get (id.lexeme);
L.type = L.array.type.elem;
L.addr = new Temp()
gen (L.addr '=' E.addr '*' L.type.width);}

| $L_1\ [E]$ {L.array = L_1.array;
L.type = L_1.type.elem;
t = new Temp();
L.addr = new Temp();
gen (t '=' E.addr '*' L.type.width);
gen (L.addr '=' L_1.addr '+' t);}

base address বা তাছা বোঝাতে direct take the variable name.

it's not mandatory যে root node সবসময় S হবে এই example এ root node E.

L.type = L.array.type.elem = array (3, integer)

Reduce operation এর সময় dependency arrow দেয়া mandatory; if the question asks to show the dependency graph.

Let's denote a 2x3 array of integers and let c, i, j all denote integers. Then, the type of a is array (2, array (3, integer)). It's 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.

Now, based on the SDD and given info draw the syntax tree, calculate the attribute value, generate the three address code representation for the given input string.

input string: c + a [i] [j]

(P.T.O)

intermediate three address code representation

$$t_1 = i * 12$$
$$t_2 = j * 4$$
$$t_3 = t_1 + t_2$$
$$t_4 = a[t_3]$$
$$t_5 = c + t_4$$