

Assignment 5

A1 (a) Arithmetic expression : Translation rules for grammar goes as:

① $L \rightarrow L S \mid n$ } Do nothing

② $L \rightarrow S \mid n$

③ $S \rightarrow id = E$

④ $E \rightarrow E_1 + E_2$

⑤ $E \rightarrow E_1 - E_2$

⑥ $E \rightarrow E_1 * E_2$

⑦ $E \rightarrow E_1 / E_2$

⑧ $E \rightarrow (E_1)$

⑨ $E \rightarrow -E_1$

⑩ $E \rightarrow \text{num}$

⑪ $E \rightarrow id$

→ Assume all variables are declared & placed in symbol table.

$E.\text{loc}$ → denote location of E in symbol table

num.val → value of numeric constant

$\text{gensym}()$ → creates new var & returns its location in symbol table.

→ Rule 3 → write ($id.\text{loc} = E.\text{loc}$)

→ Rule ④, ⑤, ⑥, ⑦: create temp variable for E & write ($E.\text{loc} = E_1.\text{loc}$ (operation) $E_2.\text{loc}$).

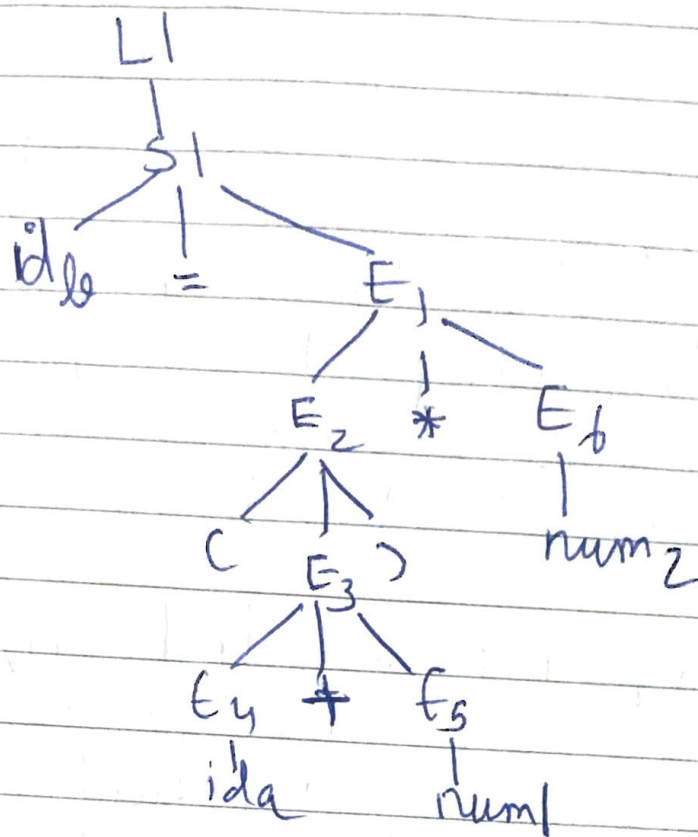
→ Rule ⑧: Need to assign variable E to the same loc as E_1 i.e. $E.\text{loc} = E_1.\text{loc}$

→ Rule ⑨: create temp var for E & write ($E.\text{loc} = -E_1.\text{loc}$)

→ Rule ⑩: Assign loc var to E & write ($E.\text{loc} = \text{num.val}$)

→ Rule ⑪ → $E.\text{loc} = id.\text{loc}$

eg: $b = (a+1)^* 2$



Steps:-

- (1) $E_4 \rightarrow id_a : E_4.loc = id_a.loc$
- (2) $E_5 \rightarrow num_1 : E_5 = gentemp()$
- (3) $E_3 \rightarrow E_4 + E_5 : E_3 = gentemp()$
- (4) $E_2 \rightarrow (E_3) : E_2.loc = E_3.loc$
- (5) $E_6 \rightarrow num_2 : E_6 = gentemp()$
- (6) $E_1 \rightarrow E_2 * E_6 : E_1 = gentemp()$
- (7) $S_1 \rightarrow id_b = E_1$
- (8) $L_1 \rightarrow S_1$

Final Translated TAC:

$$\begin{aligned}
 t_1 &= 1 \\
 t_2 &= t_1 + 1 \\
 t_3 &= 2 \\
 t_4 &= t_2 * t_3 \\
 de &= t_4
 \end{aligned}$$

→ $\text{makelist}(i)$: make a new list with i as the only element & return the pointer to the list.

→ merge(p_1, p_2) concatenate 2 lists with pointers p_1 & p_2 & backpatch(p, i) insert i as the label to all quads pointed by indlist p .

→ Taking these translation rules can be executed with following instructions to generate TAC: -

① $B \rightarrow B_1 \parallel M B_2$ & backpatch (B_1 , falselist, M.instr)
 $B_truelist = (B_1.truelist, B_2.truelist)$
 $B_falselist = B_2.falselist$

② $B \rightarrow B_1 \& \& M B_2 \{ \text{backpack}(B_1.\text{trueList}, M.\text{instr})$
 $B.\text{falseList} = (B_1.\text{falseList}, B_2.\text{instr})$
 $B.\text{trueList} = B_2.\text{trueList}$

(3) $B \rightarrow ! B \{ \begin{array}{l} B_{-} \text{false list} = B_{-} \text{true list} \\ B_{-} \text{true list} = B_{-} \text{false list} \end{array} \}$

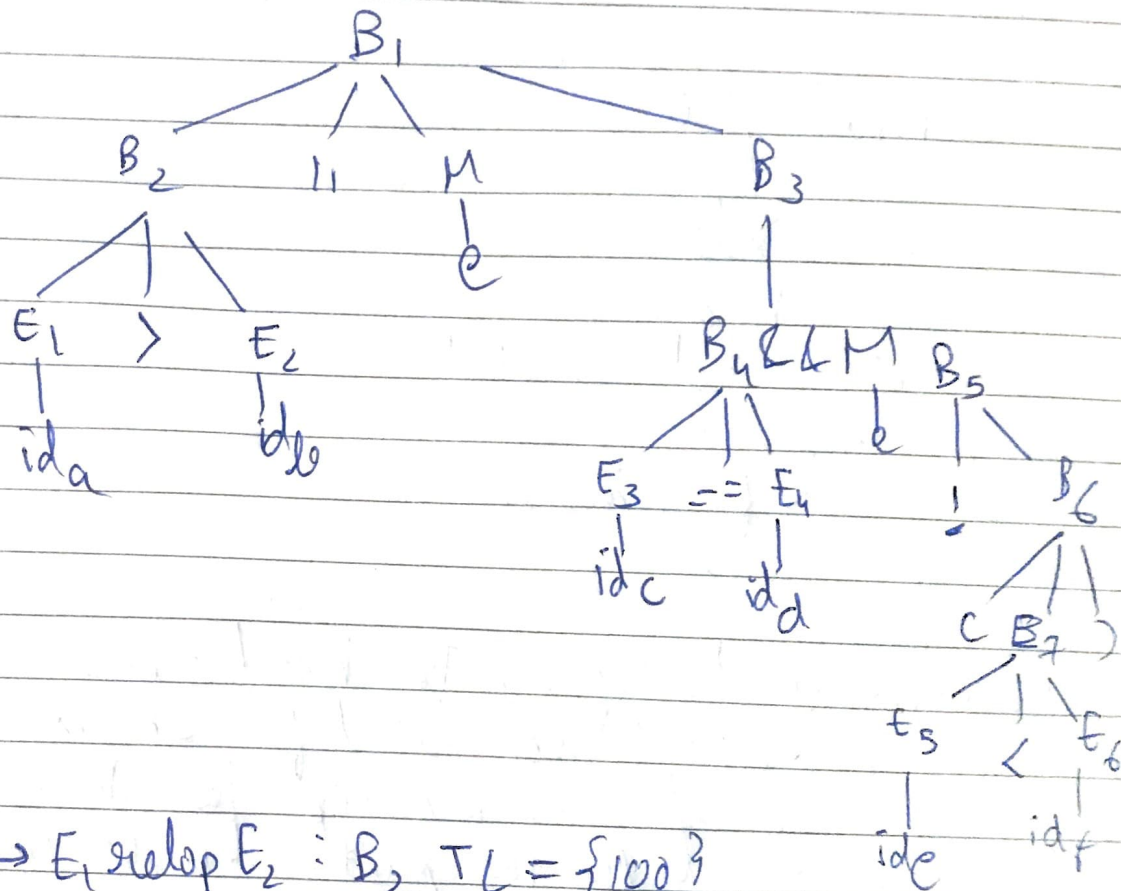
④ $B \rightarrow B, \{ B \cdot \text{travel} \cup = B, \text{travel} \}$

(5) $B \rightarrow E_1 \text{ relop } E_2$ { B.truelist = makelist(nextinstr)
+ B.falselist = makelist(nextinstr+1)
write(E_1 .loc op E_2 .loc goto
 \Rightarrow filled by B.truelist }

werte (else goto, ...)

Filled when B. false list
is back patched.

- ⑥ $B \rightarrow \text{true} \{ B.\text{trueid} = \text{nextid}(\text{next instr})$
 $\text{write}(\text{goto} \dots)$
- ⑦ $B \rightarrow \text{false} \quad \text{''} \quad \text{''}$
 $\text{''} \quad \text{''}$
- ⑧ $M \rightarrow E \{ M.\text{instr} = \text{nextinstr} \}$
- eg: $a > b \parallel c == d \&\& !(e < f)$



Steps:-

- (1) $B_2 \rightarrow E_1 \text{ relop } E_2 : B_2.TL = \{100\}$
 $B_2.FL = \{101\}$
- (2) $M \rightarrow e : M.\text{instr} = 102$
- (3) $B_4 \rightarrow E_3 \text{ relop } E_4 : B_4.TL = \{103\}$
 $B_4.FL = \{104\}$
- (4) $M \rightarrow e : M.\text{instr} = 105$
- (5) $B_7 \rightarrow E_5 \text{ relop } E_6 : B_7.TL = \{106\}$
 $B_7.FL = \{107\}$
- (6) $B_6 \rightarrow (B_7) : B_6.TL = \{106\}$
 $B_6.FL = \{107\}$

$$(7) B_5 \rightarrow !B_6 : \begin{aligned} B_5.TL &= \{107\} \\ B_6.FL &= \{106\} \end{aligned}$$

$$(8) B_3 \rightarrow B_4 \&\& MB_5 : \begin{aligned} B_3.TL &= B_5.TL = \{107\} \\ B_3.FL &= B_4.FL \cup B_5.FL = \{106, 104\} \end{aligned}$$

$$(9) B_1 \rightarrow B_2 \parallel MB_3 : \begin{aligned} B.TL &= B_2.TL \cup B_3.TL = \{107, 100\} \\ B.FL &= B_3.FL = \{106, 104\} \end{aligned}$$

∴ Final TAC:-

```

100  if a > b goto ...
101  goto 102
102  if c == d goto 104
103  goto ...
104  if e < f goto ...
105  goto ...

```

(c) Array references : To handle array we use following variables - $A.loc$ used for storing offset from array reference & $A.type$ for storing array type, $A.base$ for the pointer to the first element.

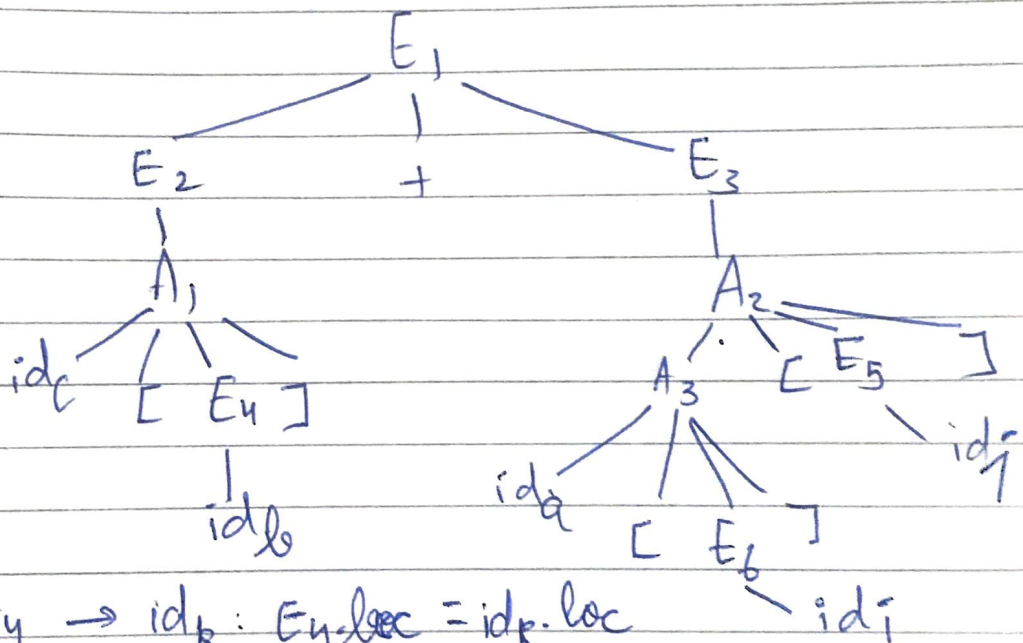
→ An array expression can be translated using the following rules:

- ① $E \rightarrow E_1 + E_2 \{ \begin{aligned} E.loc &= \text{gentemp}() \\ E.type &= E_1.type \\ \text{write}(E.loc) &= E_1.loc + E_2.loc \end{aligned} \}$
- ② $E \rightarrow id \{ \begin{aligned} E.loc &= id.loc \\ E.type &= id.type \end{aligned} \}$
- ③ $E \rightarrow A \{ \begin{aligned} E.loc &= \text{gentemp}() \\ E.type &= A.type \\ E.loc &= A.base[A.loc] \end{aligned} \}$

(4) $A \rightarrow id[E] \{ A.array = lookup[id]$
 $A.type = A.array.type.elem$
 $A.loc = gentemp()$
 $write(A.loc = E.loc * A.type.width) \}$

(5) $A \rightarrow A_1[E] \{ A.array = A_1.array$
 $A.type = A_1.type.elem$
 $A.loc = gentemp()$
 $write(t = E.loc * A.type.width);$
 $write(A.loc = A_1.loc + t)$

(eg) $c[k] + a[i][j]$



Steps :

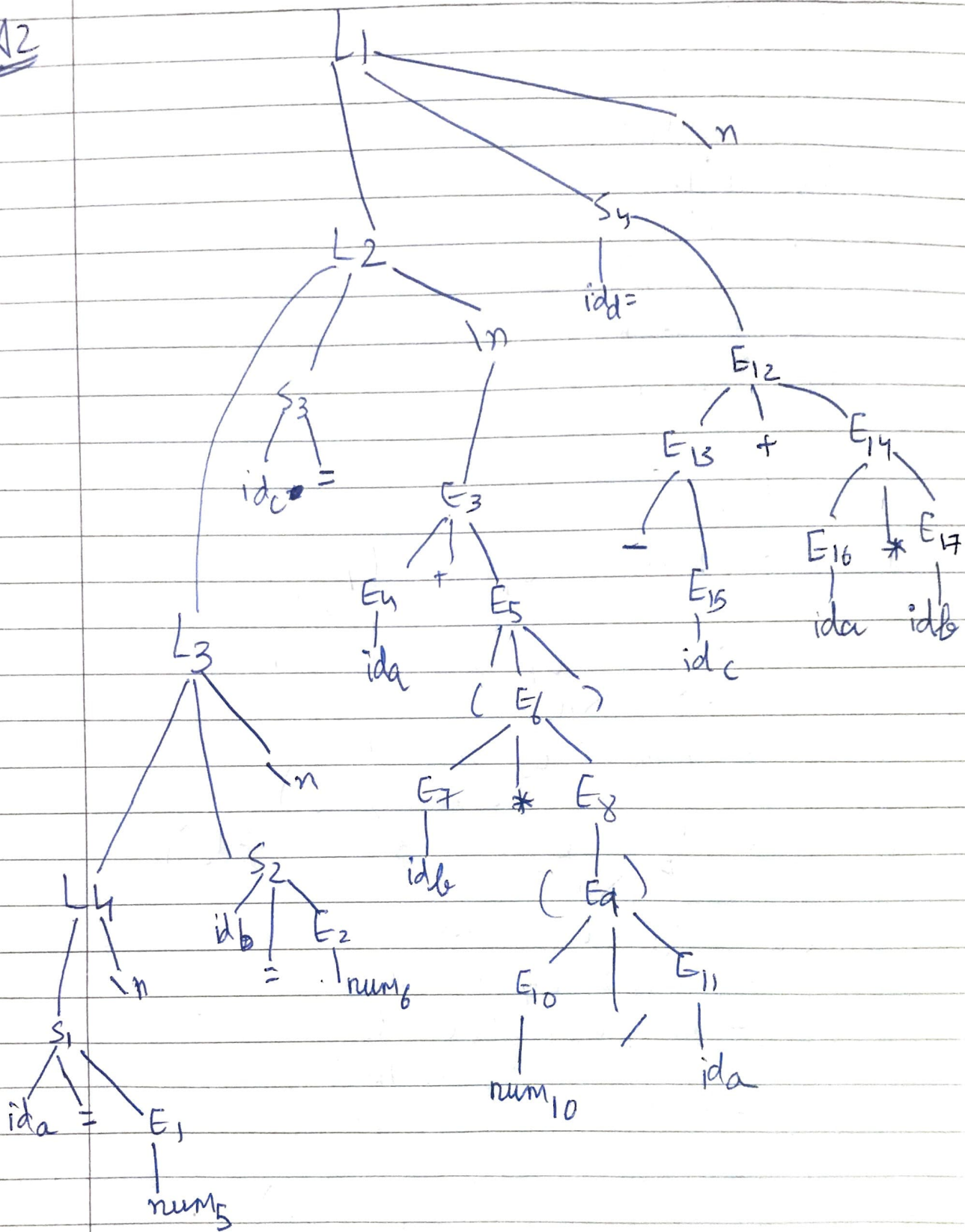
$E_4 \rightarrow id_k: E_4.loc = id_k.loc$
 $A_1 \rightarrow id_c[E_4]: A_1.type = E_4.type = int, A_1.loc = t_1$
 $E_2 \rightarrow A_1: E_2.loc = t_2, E_2.type = int$
 $A_3 \rightarrow id_a[]: E_6.loc = i, E_6.type = int$
 $A_3.array = ST[00]$
 $A_3.type = T_2$
 $A_3.loc = t_3$
 $E_5 \rightarrow id_i: E_5.type = int$
 $E_5.loc = j$

$A_2 \rightarrow A_3[E_5] : A_2.\text{array} = ST[00]$
 $A_2.\text{type} = T_2.\text{elem} = \text{int}$
 $A_2.\text{loc} = t_5$

$E_3 \rightarrow A_3 : E_3.\text{loc} = t_6$
 $E_3.\text{type} = \text{int}$

$E_1 \rightarrow E_2 + E_3 : E_1.\text{type} = E_2.\text{type}$
 $E_1.\text{loc} = t_7$
 $E_1.\text{type} = \text{int}$

A2



Final TAC:

100 $t_1 = k * 4$
 101 $t_2 = c[t_1]$
 102 $t_3 = i * 12$
 103 $t_4 = j * 4$
 104 $t_5 = t_3 + t_4$
 105 $t_6 = a[t_5]$
 106 $t_7 = t_2 + t_6$

Steps of reduction

- (i) $E_1 \rightarrow \text{num}_5 : E_1 \text{ loc} = t_1$
 $\text{write}(t_1 = 5)$
- (ii) $S_1 \rightarrow \text{ida} = E_1 : \text{write}(a = t_1)$
- (iii) $L_4 \rightarrow S_1 \setminus n$
- (iv) $E_2 \rightarrow \text{num}_6 : E_2 \text{ loc} = t_2$
 $\text{write}(t_2 = 6)$
- (v) $S_2 \rightarrow \text{id}_b = E_2 : \text{write}(b = t_2)$
- (vi) $L_3 \rightarrow S_2 \setminus n$
- (vii) $E_4 \rightarrow \text{ida} : E_4 \text{ loc} = a$
- (viii) $E_7 \rightarrow \text{id}_b : E_7 \text{ loc} = b$
- (ix) $E_{10} \rightarrow \text{num}_{10} : E_{10} \text{ loc} = t_3$
 $\text{write}(t_3 = 10)$
- (x) $E_{11} \rightarrow \text{ida} : E_{11} \text{ loc} = a$
- (xi) $E_9 \rightarrow E_{10}/E_{11} : E_9 \text{ loc} = t_4$
 $\text{write}(t_4 = t_3/a)$
- (xii) $E_8 \rightarrow (E_9) : E_8 \text{ loc} = t_4$
- (xiii) $E_6 \rightarrow E_7 * E_8 : E_6 \text{ loc} = t_5$
 $\text{write}(t_5 = b * t_4)$

- (xiv) $E_5 \rightarrow (E_1): E_5 \text{ loc} = t_5$
- (xv) $E_3 \rightarrow E_n + t_5: E_3 \text{ loc} = t_6$
 $\text{write}(t_6 = a + t_5)$
- (xvi) $S_3 \rightarrow id_c = E_3$ $\text{write}(c = t_6)$
- (xvii) $L_2 \rightarrow L_3 S_3 \backslash n$
- (xviii) $E_{15} \rightarrow id_c: E_{15} \text{ loc} = c$
- (xix) $E_{13} \rightarrow -E_{15}: E_{13} \text{ loc} = t_7$
 $\text{write}(t_7 = -c)$
- (xx) $E_{16} \rightarrow id_a: E_{16} \text{ loc} = a$
- (xxi) $E_{17} \rightarrow id_b: E_{17} \text{ loc} = b$
- (xxii) $E_{14} \rightarrow E_{16} * E_{17}: E_{14} \text{ loc} = t_8$
 $\text{write}(t_8 = a * b)$
- (xxiii) $E_{12} \rightarrow E_{13} + E_{14}: E_{12} \text{ loc} = t_9$
 $\text{write}(t_9 = t_7 + t_8)$
- (xxiv) $S_4 \rightarrow id_d = E_{12}: \text{write}(d = t_9)$
- (xxv) $L_1 \rightarrow L_2 S_4 \backslash n$

→ Final TAC

- 100 $t_1 = 5$
- 101 $a = t_1$
- 102 $t_2 = b$
- 103 $b = t_2$
- 104 $t_3 = 10$
- 105 $t_4 = t_3 / a$
- 106 $t_5 = b * t_4$
- 107 $t_6 = a + t_5$
- 108 $c = t_6$
- 109 $t_7 = -c$
- 110 $t_8 = a * t_7$
- 111 $t_9 = t_7 + t_8$
- 112 ~~write~~ $d = t_9$