

CS & IT ENGINEERING

Compiler Design

Intermediate code and code optimization

Lecture No. 1



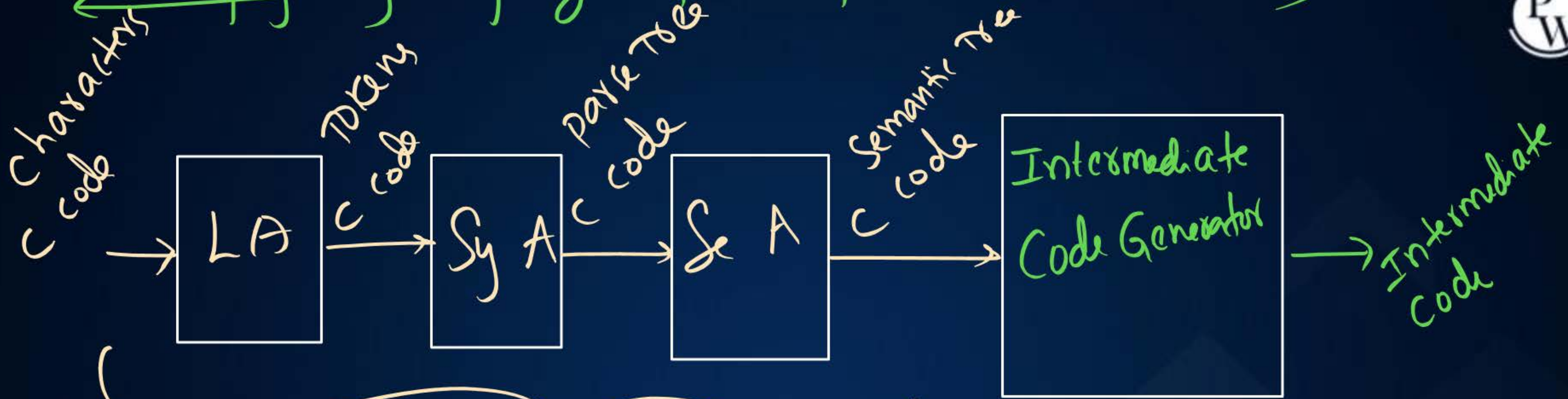
By- DEVA Sir



Intermediate code

- Three Address Code (TAC/3AC)
- Static Single Assignment Code (SSA)
- Directed Acyclic Graph (DAG)
- Control Flow Graph (CFG)

Programming Language Dependent Phases



Free From
Compiler errors
Lexical
syntax
semantic

Lexical, syntactically & semantically
Analyzed C code

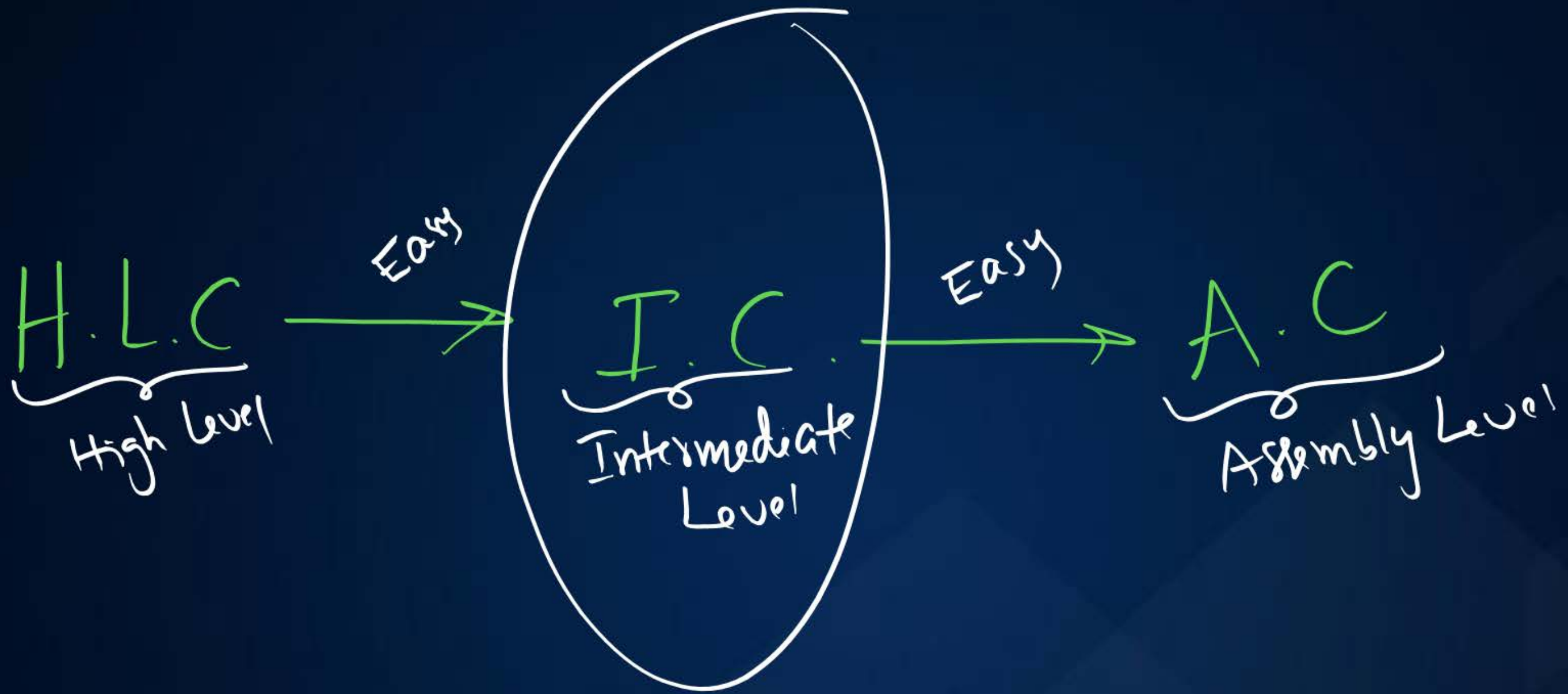


Intermediate code

$x = y + z * a;$

$t_1 = z * a$
 $x = y + t_1$

x	t ₁	int	int 10
---	----------------	-----	--------



Intermediate Code Representations:



- ↳ 1. Postfix code
 - *** 2. Three Address code
 - 3. SSA code
 - 4. Syntax Tree
 - 5. DAG
 - *** 6. Control Flow Graph
- Linear Forms
- Non-linear Forms

$$x = (a+b) * (a+b)$$

→ Postfix code :

$$x = (a+b) * (a+b)$$

$\underbrace{ab+}$ $\underbrace{ab+}$
 $\underbrace{ab+ \quad ab+ \quad *}$
 $x \quad ab+ab+* =$

postfix code
(Intermediate code)

$p+q$

$pq+$

$$x = (a+b) * (a+b)$$

→ 3AC/TAC/3 Address code/Three Address Code

$$x = \underbrace{(a+b)} * \underbrace{(a+b)}$$

(optimized)
Best 3AC

$a = a+b$
 $a = a*a$

2 variables
 a, b

OR

$t_1 = a+b$
 $t_2 = a+b$
 $x = t_1 * t_2$

5 variables

OR

$t_1 = a+b$
 $x = t_1 * t_1$

4 variables
 x, a, b, t_1

3AC

↳ Every statement
has max 3 Address

x Assignment

x Definition

x meaning

$$\boxed{x} = \overset{\text{use}}{y} + \overset{\text{use}}{z};$$

x got new assignment
(definition)
(meaning)

$$x = (a+b) * (a+b)$$

→ Static Single Assignment Code:

$$x = \underbrace{(a+b)} * \underbrace{(a+b)}$$

(optimized)
Best 3AC

$$\begin{aligned} \boxed{a} &= a+b \\ \boxed{a} &= a*a \end{aligned}$$

OR

$$\begin{aligned} t_1 &= a+b \\ t_2 &= a+b \\ x &= t_1 * t_2 \end{aligned}$$

OR

$$\begin{aligned} t_1 &= a+b \\ x &= t_1 * t_1 \end{aligned}$$

It is 3AC but every variable should contain single assignment

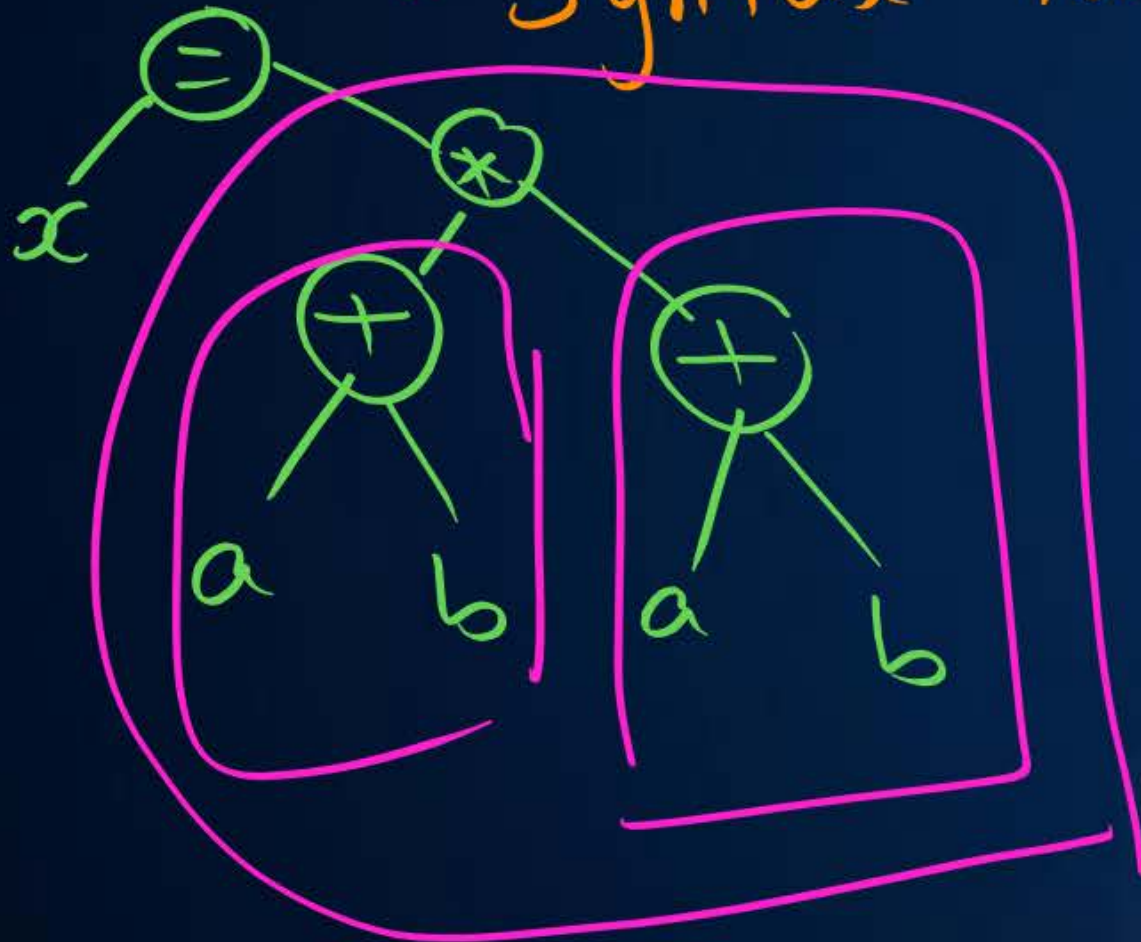
2 variables
a, b
3 has assignments

5 variables
SSA Code

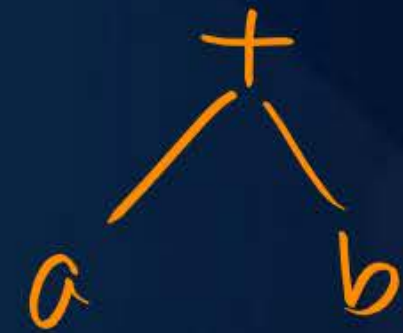
4 variables
Best SSA Code

$$x = (a+b)*(a+b)$$

Syntax Tree :



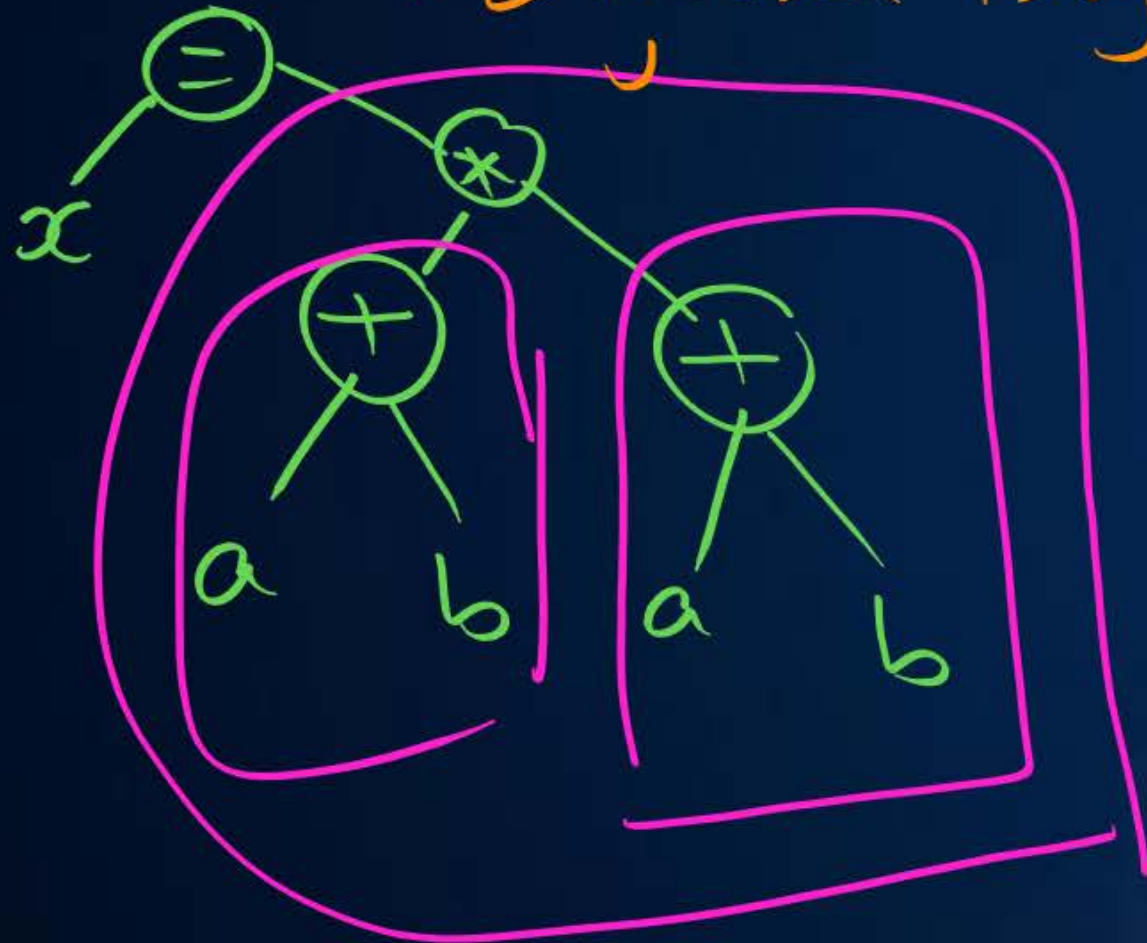
$a + b$



Leaf node: operand
Nonleaf node: operator

$$x = (a+b) * (a+b)$$

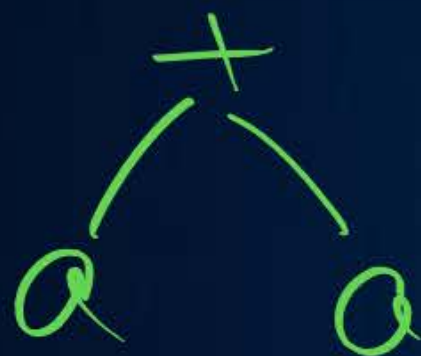
Directed Acyclic Graph



→ It eliminates
Common Sub-expressions

$a + a$

a is common sub exp



Syntax Tree



DAG

$(a+b) * (a+b)$



Syntax Tree



DAG

①

$$x = (a * b) + a + (b * c)$$



Best 3AC

$$x = (a * b) + a + (b * c)$$

Diagram showing variable reuse: a is used in $(a * b)$ and a . b is used in $(a * b)$ and $(b * c)$. c is used in $(b * c)$. The first b in $(a * b)$ is labeled t_1 and the second b in $(b * c)$ is labeled b_{2nd} .

~~$$\begin{aligned}
 t_1 &= a * b \\
 b &= b * c \\
 a &= t_1 + a \\
 a &= a + b
 \end{aligned}$$~~

Not Best

$$x = (a * b) + a + (b * c)$$

Diagram showing variable reuse: a is used in $(a * b)$ and a . b is used in $(a * b)$ and $(b * c)$. c is used in $(b * c)$. The first b in $(a * b)$ is labeled b and the second b in $(b * c)$ is labeled b . The first c in $(b * c)$ is labeled c .

Reordering

$$c = b * c$$

c will be free after $b * c$

$$b = a * b$$

$$b = b + a$$

$$b = b + c$$

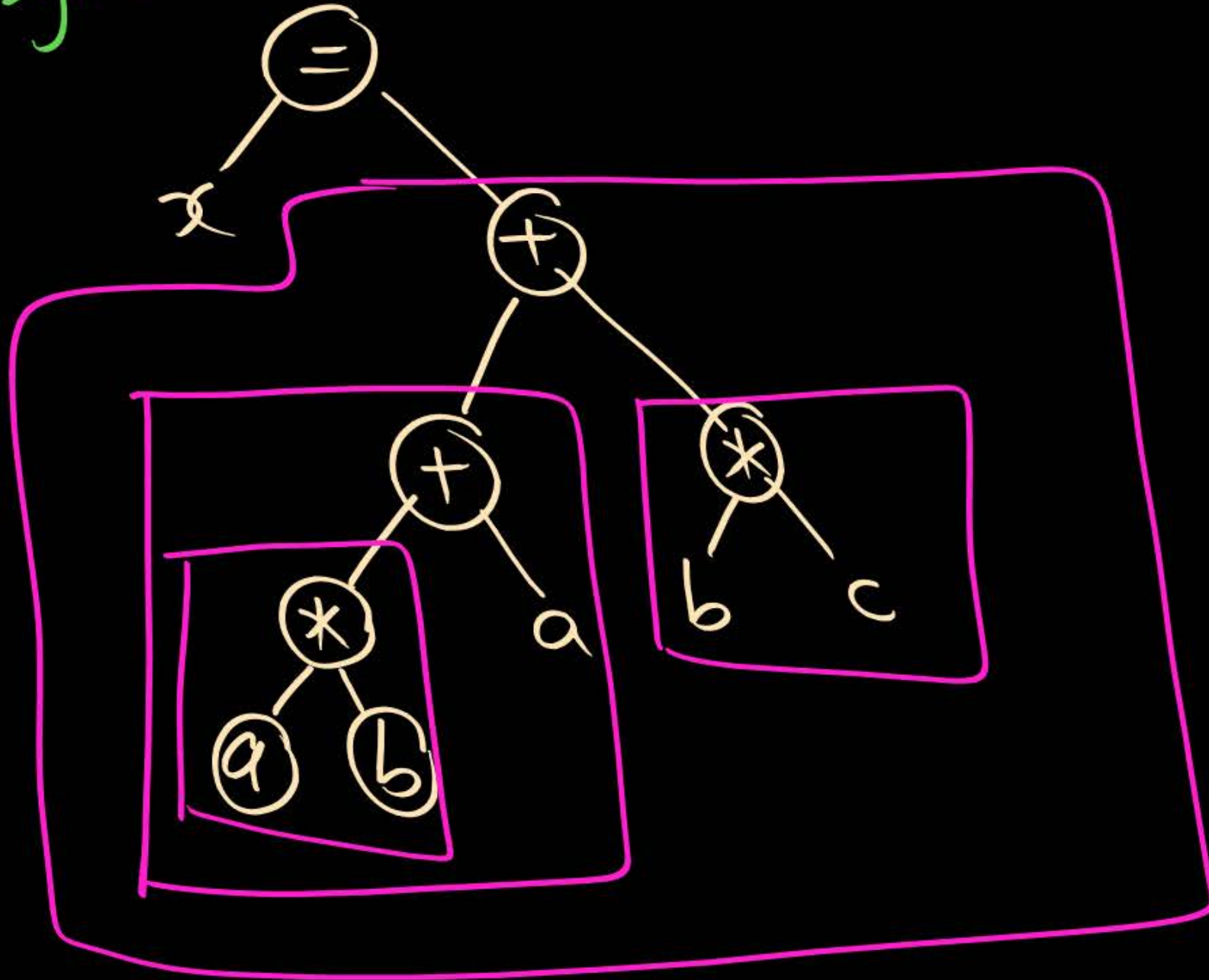
Min 3 variables required

②

$$x = (a * b) + a + (b * c) = \underline{a * b} + a + \underline{b * c}$$



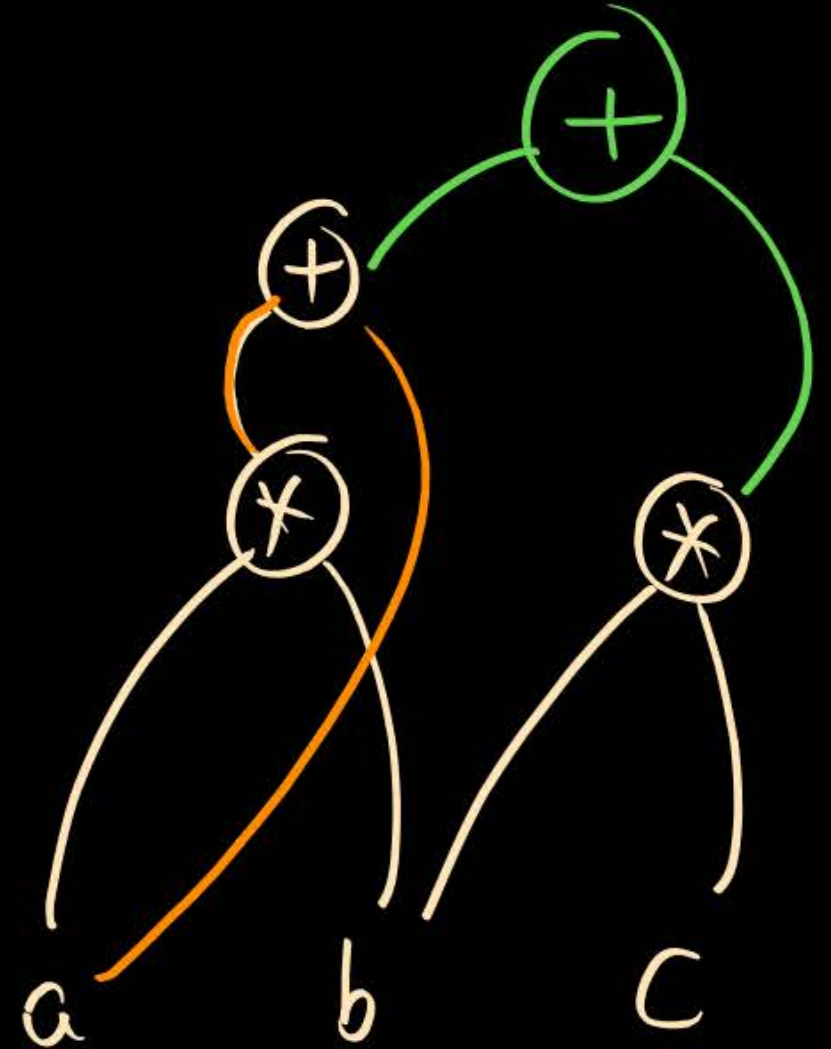
Syntax tree



Find no. of nodes & edges in DAG.

DAG:

7 nodes
8 edges



③

$$\underbrace{a+a+a}_{t_1}$$

Find min no. of variables in 3AC

$$t_1 = a + a$$

$$a = t_1 + a$$

2 Variables
 a, t_1



④

$$a + a + a$$



Find min no. of variables in SSA code.

Best 3AC:

$$\begin{array}{l} \boxed{t_1}^{\text{new}} = a + a \\ \boxed{a}^{\text{existing}} = t_1 + a \end{array}$$



Best SSA Code:

$$\begin{array}{l} \boxed{t_1}^{\text{new}} = a + a \\ \boxed{a_1}^{\text{new}} = t_1 + a \end{array}$$

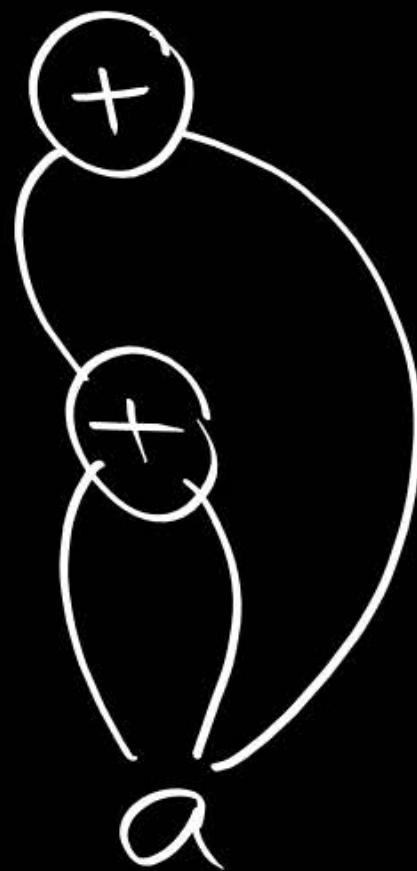
} a, t_1, a_1
3 variables

⑤

$$a + a + a$$



Find min no. of nodes & edges in DAG



3 nodes
4 edges

⑥

$$\underbrace{a+a}_{a} + \underbrace{a+a}_{a}$$

Find min no. of variables in 3AC

$$\underbrace{\underbrace{(a+a)}_a + \underbrace{(a+a)}_a}_a$$

$$\boxed{a} = a + a$$

$$a = a + a$$

$\Rightarrow 1$ variable

$$a = \underbrace{a + a}$$

$$a = \boxed{a} + \boxed{a}$$

Find equivalent expression.

$$a = \boxed{a} + \boxed{a}$$

$$a = (a + a) + (a + a) \checkmark$$

7

$$a + a + a + a$$



Find min no. of variables in SSA code.

Best 3AC :

existing

$$\boxed{a} = a + a$$

existing

$$\boxed{a} = a + a$$

Best SSA code :

$$\Rightarrow \begin{cases} a_1 = a + a \\ a_2 = a_1 + a_1 \end{cases}$$

= 3 variables

⑧ $\underline{a+a+a+a}$

Find min no. of nodes & edges in DAG

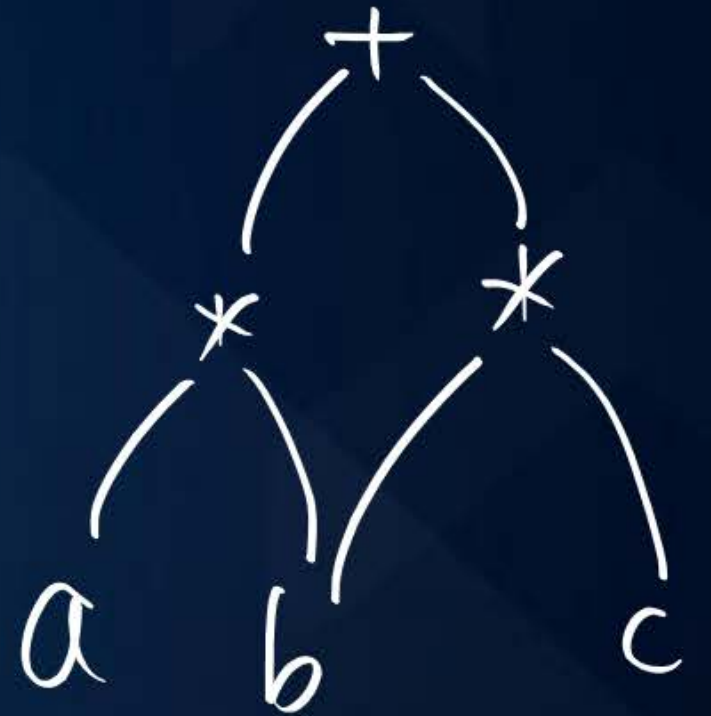


3 nodes
and
4 edges

⑨ Find DAG for following 3AC

$$\begin{aligned} a &= a * b \\ b &= b * c \\ a &= \boxed{a} + \boxed{b} \end{aligned}$$

$$\Rightarrow a = a + b = (a * b) + (b * c) \Rightarrow \text{DAG:}$$



Approach: 3AC \Rightarrow expression \Rightarrow DAG

⑩ Find min no. of variables in equivalent 3AC for following 3AC.

Approach :

3AC

↓
expression

↓
Best 3AC

$$\begin{aligned} c &= a + b \\ d &= a + b \\ e &= c * a \\ f &= d * b \\ g &= e + f \end{aligned}$$

Given

$$\begin{aligned} g &= [e] + [f] \\ &= (c * a) + (d * b) \end{aligned}$$

$$g = (\underbrace{[a+b]}_{t_1} * a) + (\underbrace{[a+b]}_{t_1} * b)$$

$$t_1 = a + b$$

$$\left. \begin{aligned} a &= t_1 * a \\ b &= t_1 * b \end{aligned} \right\} \text{independent!}$$

$$a = a + b$$

= 3 variables



postfix ✓

3AC ✓

SSA ✓

DAG ✓

Syntax Tree ✓

