

Compiler Design

Today's Class Topics

- Syntax directed translation (SDT)
- Types of attributes & SDT
- 3 Address code ✓
- Implementation of 3 address code
 - Quadrupole ✓
 - Triple ✓
 - Indirect Triple ✓
- Static Single Assignment(SSA) ✓
- Directed Acyclic graph(DAG) ✓



Compiler Design

Syntax Directed Translation(SDT): A grammar with semantic rule is called as SDT

Ex:-

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow id$$

$$\boxed{SDT = CFG + Semantic Rules}$$

Extra information

Source Code

↓
Lexical Analysis

↓ Tokens

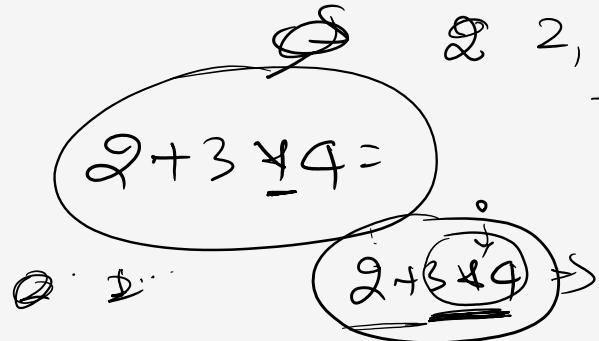
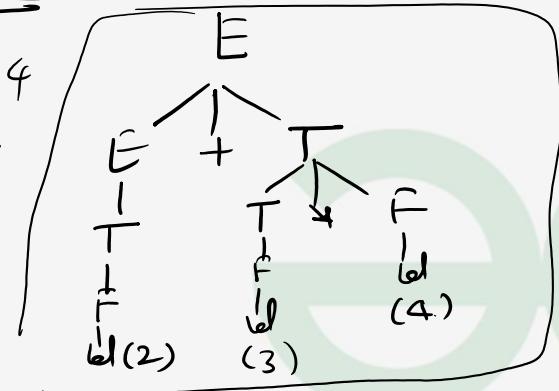
↓ Syntax Analysis

↓ Parse tree

↓ Semantic Analysis

$$\overbrace{2+3*4}$$

Parse tree



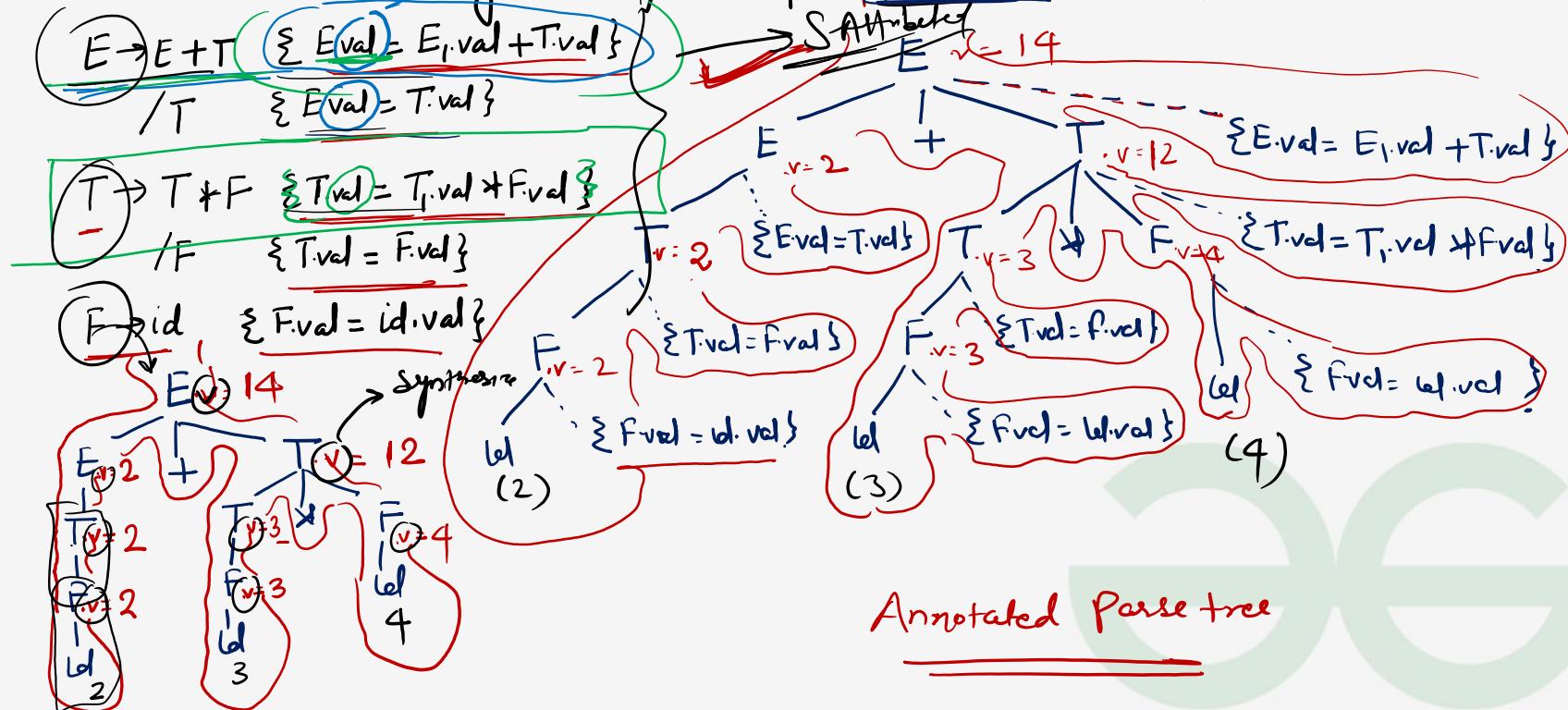
$$E \rightarrow E_i + T$$

$$E = \underline{\underline{E_1}}$$

Compiler Design

→ Attributed

Ex:- Consider the following SDT, if the e/p = $2+3+4$ Then o/p = ?



Compiler Design

SDT

Ex:- Consider the following SDT, if the i/p = $2 + 3 * 4$ then o/p = ?

$$E \rightarrow E + T \quad \{ \text{Point} ("+") \}$$

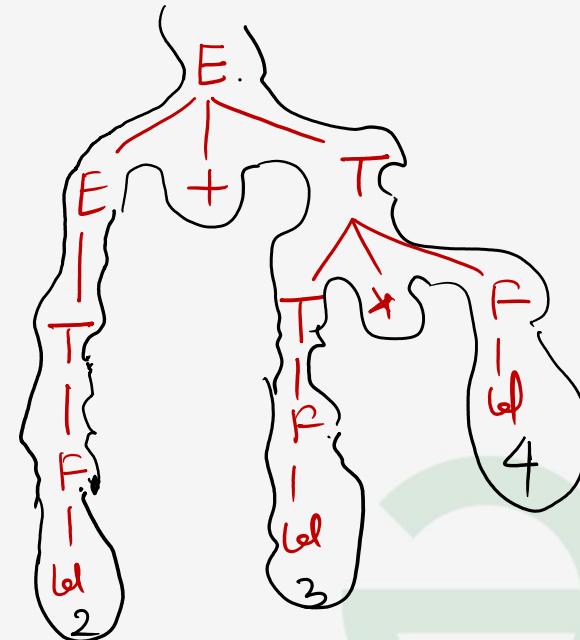
$$/ T \quad \{ \quad \}$$

$$T \rightarrow T * F \quad \{ \text{Point} ("*") \}$$

$$/ F \quad \{ \quad \}$$

$$F \rightarrow \text{id} \quad \{ \text{Point} (\text{id}, \text{val}) \}$$

✓ 2 3 4 * +

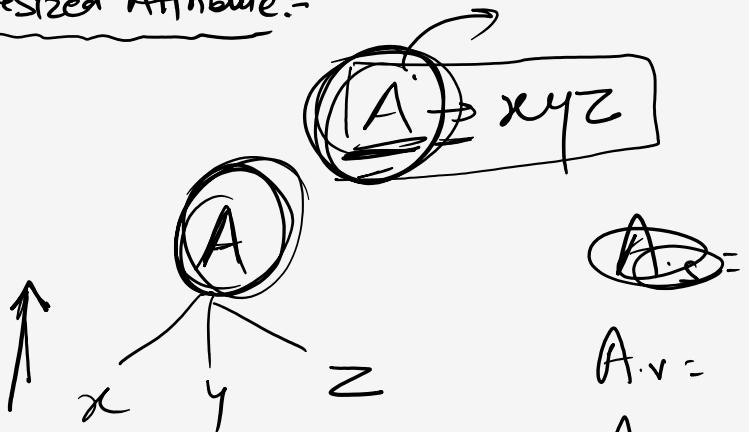


Compiler Design

Types of Attributes: *Synthesized Attribute*

(for Parent value)

Synthesized Attribute:-

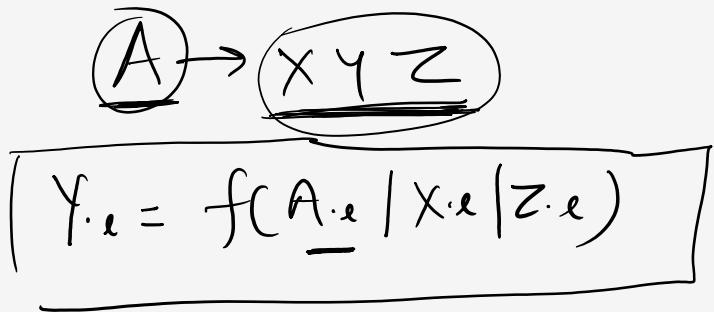


$$A.v = f(x.v | y.v | z.v)$$

$$A.s = f(x.s | y.s | z.s)$$

Compiler Design

(for children value)
Inherited Attribute



Compiler Design

Types of SDT:

S-Attributed SDT (only synthesized Attributes)

L-Attributed SDT (Synthesized and Inherited)

Compiler Design

~~SDD~~, ~~SDT~~

~~SDT~~

S-Attributed SDT

- 1- A SDD that uses only Synthesized Attribute
is called S-Attributed SDD

Ex: $A \rightarrow BCD$
 $A.v = B.v$
 $A.v = C.v$
 $A.v = D.v$



2) Bottom up



3) $A \rightarrow xyz \{ \}$
 Semantic Action on RHS

L-Attributed SDT

- 1- A SDD that uses both Synthesized and Inherited attributes
is called as L-Attributed SDD but each inherited attribute is restricted to inherit from Parent or Left sibling only

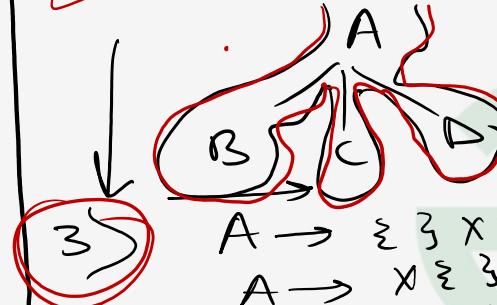
Ex: $A \rightarrow \underline{BCD}$

$C.v = A.v$
 $C.v = B.v$

~~$C.v = D.v$~~

2) Depth first Left to Right.

$A \rightarrow \underline{BCD}$

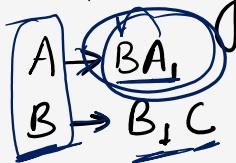


Semantic Action
Can be placed Anywhere
on RHS

$A \rightarrow \{ \underline{xyz} \}$
 $A \rightarrow x \{ \underline{yz} \}$
 $A \rightarrow x \underline{y \{ z \}}$
 $\underline{x} \underline{y \{ z \}}$
 $x \underline{y \{ z \}}$
 $x \underline{y \{ z \}} \Rightarrow xyz \{ \}$

Compiler Design

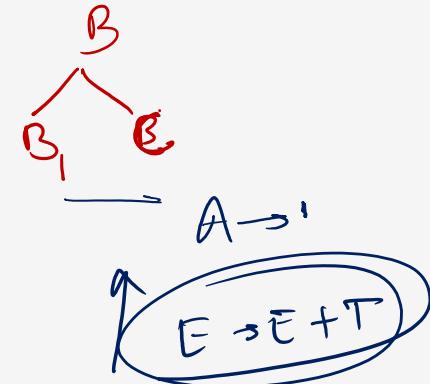
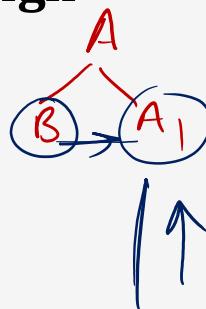
Ex:- The following SDT is



$$\{ \underline{A.x} = \underline{B.y} \}$$

$$\{ \underline{C.x} = \underline{B.x} - \underline{B_1.x} \}$$

- a) S-Attributed SDT
- b) L-Attributed SDT
- c) Both S & L-Attributed
- d) None



$$A.x = B.x$$

$$B.y = D.y$$

Ex:- $A \rightarrow SBX \{ \underline{x.i} = \underline{s.i} + \underline{A.i} ; \underline{B.i} = \underline{s.i} \}$

$$B \rightarrow ACD \{ \underline{C.i} = \underline{A.i} + \underline{B.i} - \underline{D.i} \}$$

- ~~a~~) S-Attributed
- ~~b~~) L-Attributed
- c) Both
- d) None



Compiler Design

Ex:- $A \rightarrow BC$ $\sum \{ B.i = A.i; C.i = B.i; A.i = C.i \}$

$B.v \Rightarrow A.v$
 $C.v \Rightarrow B.v$
 $A.v \Rightarrow C.v$

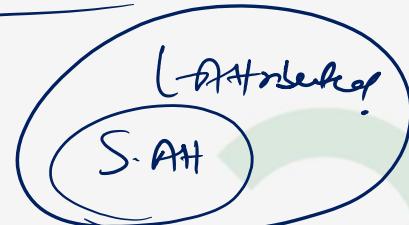
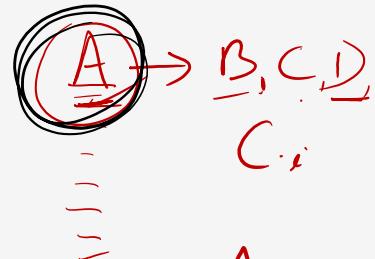
SL \Rightarrow L-Attributed

Ex:- $A \rightarrow QR$ $\sum \{ R.i = A.i; Q.i = R.i; A.i = Q.i \}$

not L-Attributed

Ex:- $A \rightarrow P Q$ $\sum \{ A.i = P.i, A.i = Q.i \}$

S-Attribute \Rightarrow L-Attributed



Compiler Design

Ex:- Consider the following SDT

Start $E \rightarrow E_1 + T \quad \{ E.val = E_1.val + T.val \}$

$E_1 \rightarrow E_1 + T \quad \{ E_1.val = T.val \}$

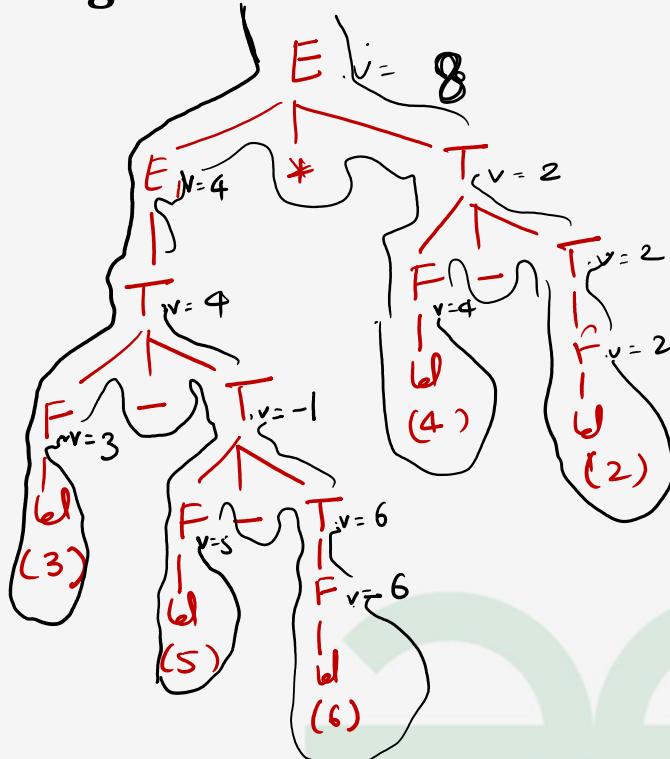
$T \rightarrow F - T_1 \quad \{ T.val = F.val - T_1.val \}$

$F \rightarrow F - T \quad \{ F.val = T.val \}$

$F \rightarrow id \quad \{ F.val = id.val \}$

If i/p string $((3 - 5) - 6) + (4 - 2)$ then o/p = ?

- $\Rightarrow *$
 - $\Leftarrow -$
 $* \Rightarrow +$



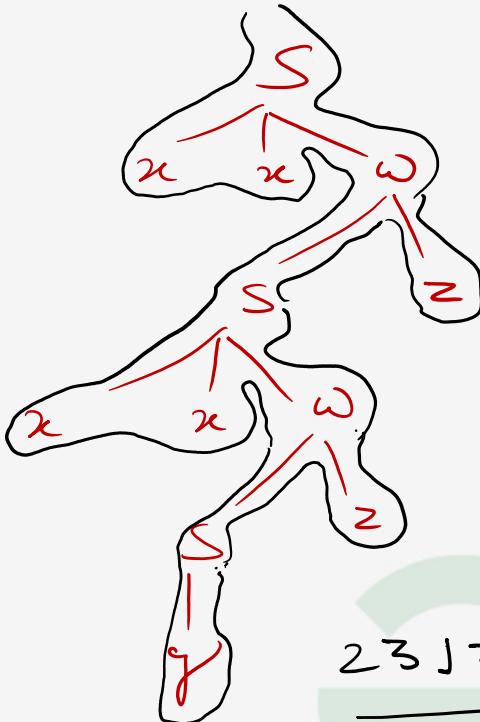
Compiler Design

Ex:- Consider the following SDT

$$\begin{aligned} S &\rightarrow xx\omega \quad \{ \text{Print("1");} \\ &\quad / y \quad \{ \text{Print("2");} \\ \omega &\rightarrow Sz \quad \{ \text{Print("3");} \end{aligned}$$

if input $xxxxyzz$ Then o/p = ?

$$o/p = \underline{\underline{23131}}$$

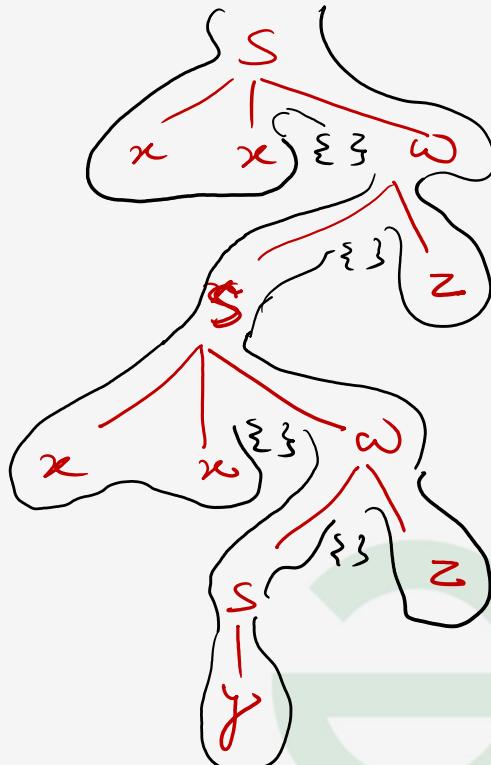


$$\underline{\underline{23131}}$$

Compiler Design

Ex:- $S \rightarrow x x \{ \text{print("1")}, \} \omega$
 $/ y \{ \text{print("2")}; \}$
 $\omega \rightarrow S \{ \text{print("3")}, \} z$
if input $xxxxyzz$ Then $O/p = ?$

$$O/p = \underline{11233}$$



Compiler Design

Question:- Consider the grammar with the following translation rules

$$\checkmark E \rightarrow E_1 * T \quad \{ E \cdot \text{val} = E_1 \cdot \text{val} * T \cdot \text{val} \}$$

$$/ T \quad \{ E \cdot \text{val} = T \cdot \text{val} \}$$

$$\checkmark \frac{T \rightarrow T_1 + F}{/F} \quad \left\{ \begin{array}{l} T.\text{val} = T_1.\text{val} + F.\text{val} \\ T.\text{val} = F.\text{val} \end{array} \right\}$$

$\checkmark F \rightarrow \text{num} \quad \{ F.\text{val} = \text{num}.\text{val} \}$

Compute Evaluate for the Root of the Parse tree

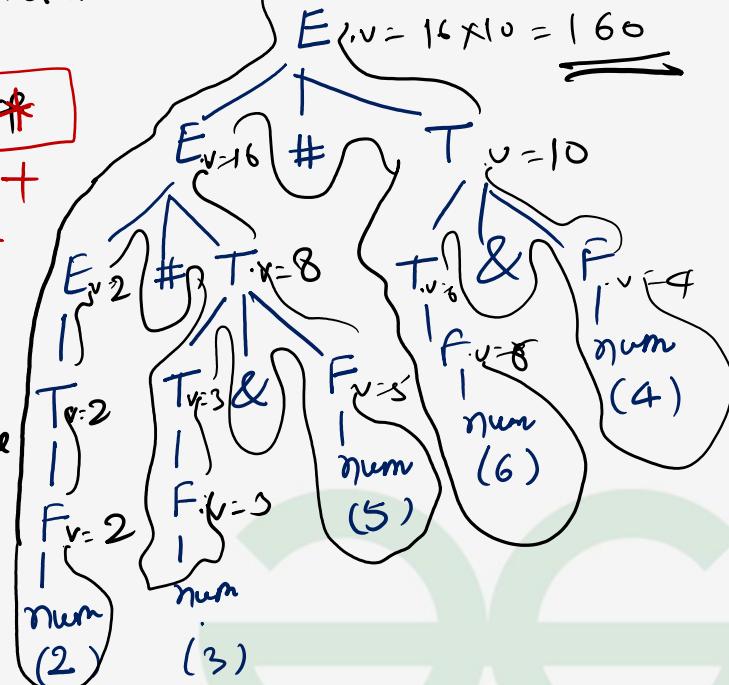
for the expression $2 \# 3 \& 5 \# 6 \& 4$

$$\left(\left(2 * \underline{3 + 5} \right) * \left(6 + 4 \right) \right)$$

160

160

160 Am



2#3&5#6&4

Compiler Design

Question:

Production

$$E \rightarrow E \# T$$

$$E \rightarrow T$$

$$T \rightarrow T \& F$$

$$T \rightarrow F$$

$$F \rightarrow \text{digit}$$

$$\begin{array}{l} \# \rightarrow * \\ \& \rightarrow * \quad + \rightarrow * \end{array}$$

Semantic Rules

$$E.\text{val} = E_1.\text{val} * E_2.\text{val}$$

$$E.\text{val} = T.\text{val}$$

$$T.\text{val} = ?$$

$$T.\text{val} = F.\text{val}$$

$$F.\text{val} = \text{digit}.v$$

✓1) If the expression $8 \# 12 \& 4 \# 16 \& 12 \# 4 \& 2$

is evaluated to 512 Then which of the following
is correct replacement of Blank

- a) $T.\text{val} = T.\text{val} * F.\text{val}$ c) ~~$T.\text{val} = T.\text{val} - F.\text{val}$~~
- b) $T.\text{val} = T.\text{val} + F.\text{val}$ d) None

$$\begin{array}{l} 10 * (8 - 6) * (9 - 4) * (5 - 2) \\ \checkmark 2) 10 \# 8 \& 6 \# 9 \& 4 \# 5 \& 2 ? \Rightarrow 10 * 2 * 5 * 3 \\ \hline 30 * 10 \\ \Rightarrow 300 \end{array}$$

$$\begin{array}{l} \# \rightarrow * \\ \& \rightarrow - \end{array} \quad \boxed{- \rightarrow *} \quad \boxed{*}$$

$$8 * (12 + 4) * (16 - 12) * (4 - 2)$$

$$\underline{8 * 16 * 28 * 8}$$

$$8 * (12 - 4) * (16 - 12) * (4 - 2)$$

$$\Rightarrow 8 * 8 * 4 * 2$$

$$\Rightarrow 2^3 * 2^3 * 2^2 * 2^1$$

$$\Rightarrow 2^9 \Rightarrow 512$$