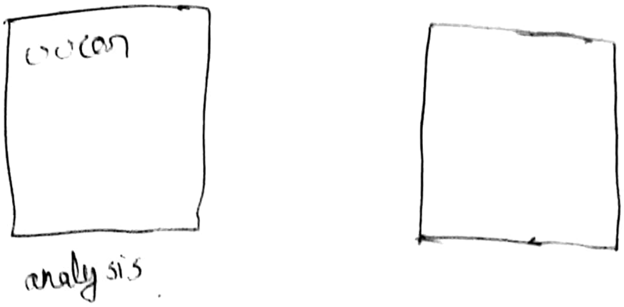
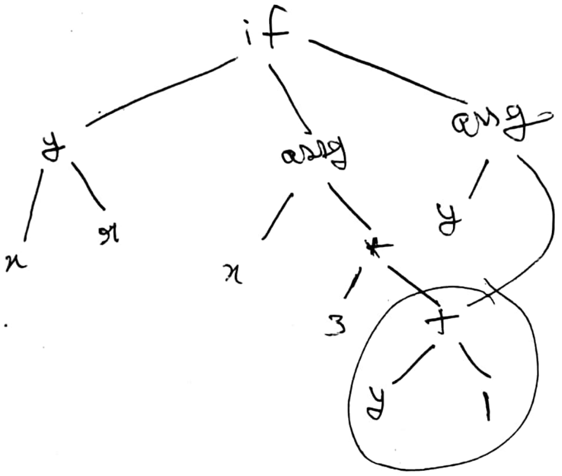
Compi
Joyati Ma

code generation



Intermediate
- high level (source language) — Parse / DAG (Directed Acyclic graph)
- Low level (Target language) — three address code



// if a<b then x=y+z ~~then~~ else p = z+r
converting to three address code.

① if a<b then goto ③
② goto ⑤
③ $t_1 = y + z$
④ $x = t_1$
⑤ $t_2 = z + r$
⑥ $p = t_2$

④

while $(a<b)$

do {

$x = y+z$

three address code:

① if $a<b$ goto ③

② ~~if~~ goto ~~⑤~~ ⑥

③ $t_1 = y+z$

④ $x = t_1$

~~⑤ other statements~~

⑤ goto ①

⑥ Next instruction outside ~~the~~ loop.

④

for $(i=1 ; i \leq 20 ; i{+}{+})$

$x = y+z$

three address code

① $i = 1$

② if $i \leq 20$ goto ⑦

③ goto ⑩

④ $t_1 = i+1$

⑤ $i = t_1$.

⑥ goto ②

⑦ $t_1 = y+z$

⑧ $x = t_1$

⑨ goto ④

⑩ next instruction ~~outer~~ outside the loop.

Boolean expression

if a<b && c<d then

$p=q+r$

three instruction code:

① if a<b goto ③

② goto ⑧

③ if c<d goto ⑤

④ goto ⑧

⑤ $t_1 = q+r$

⑥ $p = t_1$

⑦ goto ___

⑧ Next instruction

A<B or C<D and P<S

three address code:

① if A<B goto ⑦

② goto ~~⑧~~ ③

③ if C<D goto ⑥

④ goto ⑧

⑤ if P<S goto ⑦

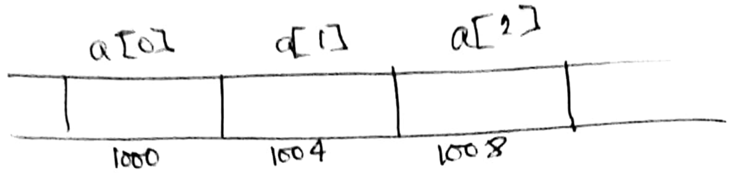
⑥ goto ⑧

⑦ next instruction in the body

⑧ out the body

Will generate three address code for array :-

1-D array



base address - address of first element

width - number of bytes occupied by each element

$\{ a[n] = \text{base address} + n * \text{width address} \}$

$$\boxed{\begin{array}{l} t_1 = n * w \\ t_1 = b + t_1 \end{array}}$$

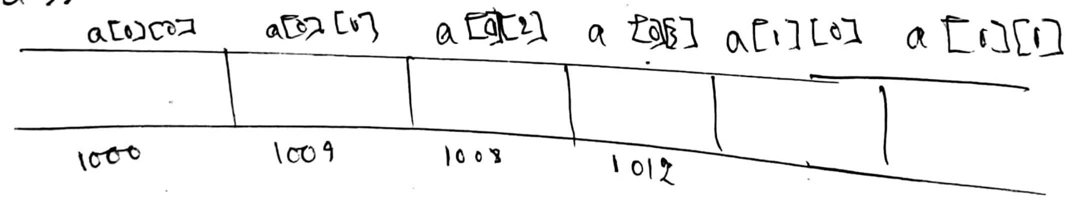
Three address code for this operation.

2-D array

1	2	3	4
5	6	7	8
9	10	11	12

A[10][4]

How it stores in memory



~~a[n][m]~~ =

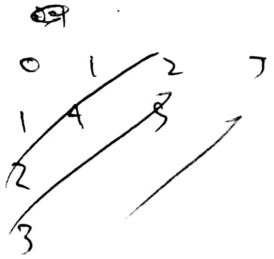
$a[i][j] = \text{base address} + w[N * i + j]$

3 2

$= 1000 + 4[4 \times 3 + 2]$

$= 1000 + 4 \times 14$

~~1036~~ = 1056



Scanned with CamScanner

$t_r = N * i$

$t_1 = t_1 + j$

$t_1 = t_1 * W$

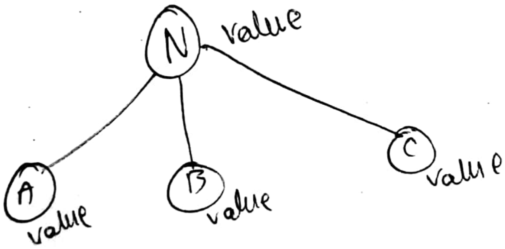
$t_1 = t_r + \text{Bare address}$

Syntax Directed Definition :-

Context free grammar + attribute + rules

Attributes -

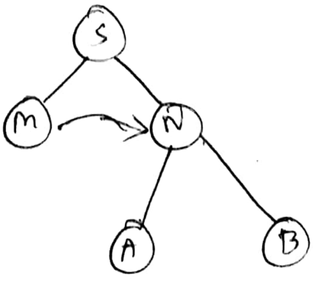
synthesized - synthesized attributes of Node N is defined in term of attributes value of ~~chd~~ children of N and itself.



$$N_{value} = A_{value} + B_{value} + C_{value}$$

$N \rightarrow ABC$

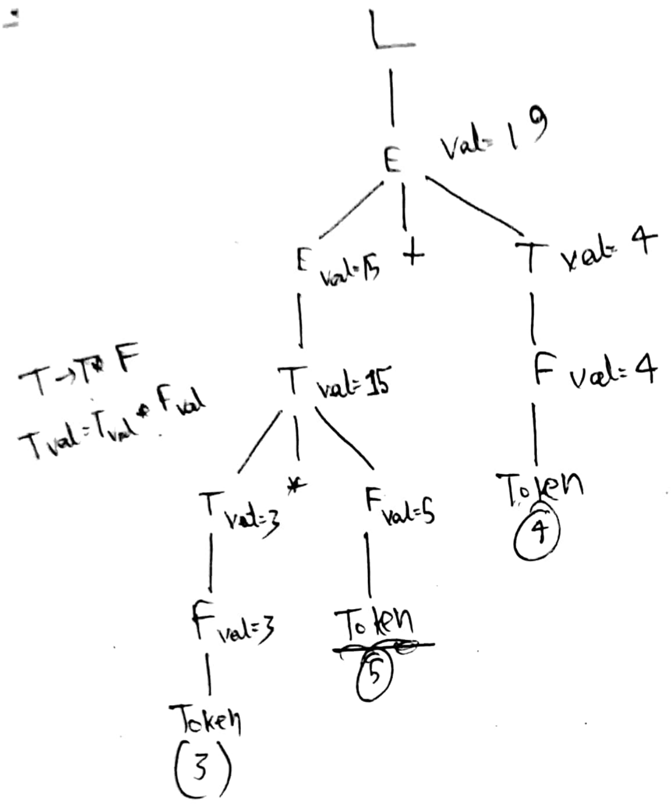
Inherited attributes -



Annoted Parse tree — Parse tree + attributes & its value

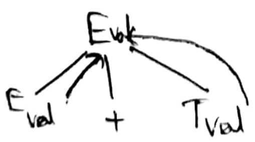
Example

3 * 5 + 4

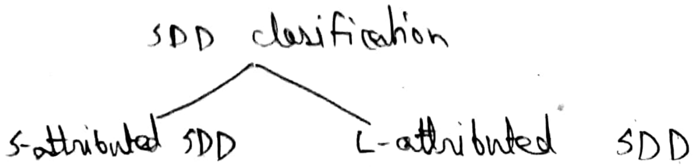


Dependency Graph :-

It is a tool for determining the evaluation ord of the attributes in a parse tree



SDD in two categories!

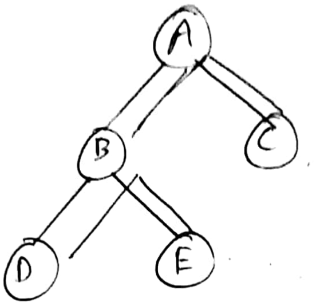

SDD clasification

S-attributed SDD

L-attributed SDD

(i) s-attributed SDD

- attributes will be synthesized

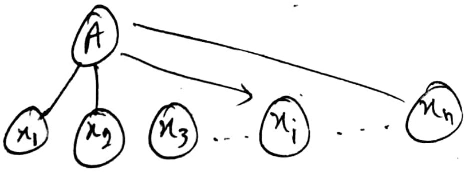


- Bottom up parsing

(ii) L-attributed SDD

- synthesized
- Inherited

$A \rightarrow x_1, x_2, x_3, \ldots, x_n$



- Top down

Application of SDD

- Type checking
- Intermediate code generation
- Construction of Syntax tree.