K·Jeevan Sai
AP18110010266
Set-1

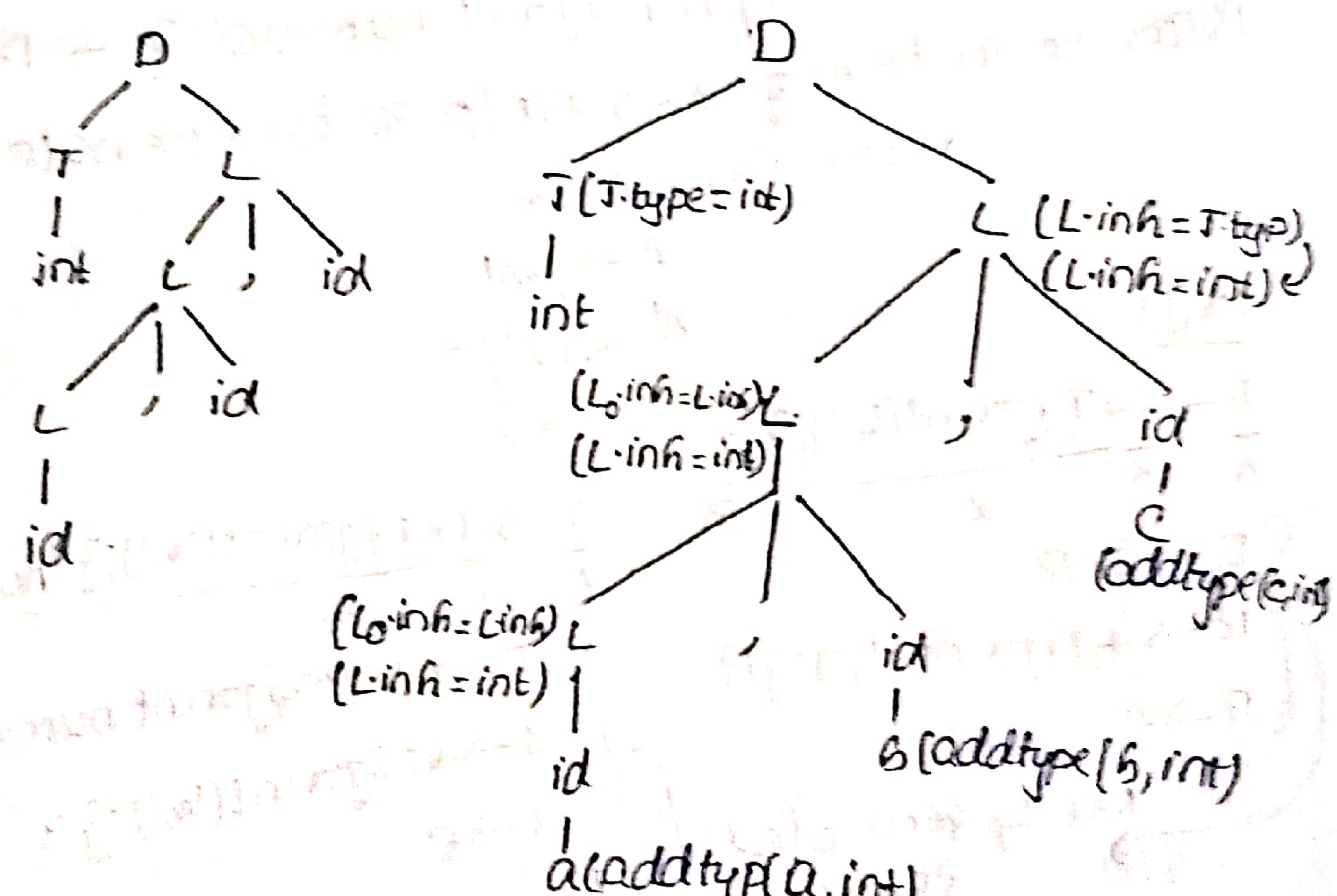
CD Assignment-6

1) Syntax directed declaration

Annotated parse tree for int a,b,c ?

Production rules	semantic rules
$D \to TL$	$L.inher = T.type$
$T \to int$	$T.type = int$
$T \to float$	$T.type = float$
$L \to L_0, id$	$L_0.inher = L.inher$ add type(id.entry, L.inher)
$L \to id$	add type(id.entry, L.inher)

Annotated parse tree



2) SyntaxDirected technique that translates infix to postfix

Expression $\Rightarrow 3*4+5*2$

Production rules   semantic rules

$E \to E+T$ {print('+');}

$E \to T$

$T \to T*F$ {print('*');}

$T \to F$

$F \to num$ {print num.val;}

$E \to E+T$ {print('+');} / T $\to$ Pr 1

$E \to T$

$T \to T*F$ {print('*');} / num {print num.val;} $\to$ Pr 2

Both are in form of $A \to A\alpha / \beta$ so to elemenate left recursion we do

$A \to \beta A'$

$A' \to \alpha A' / \epsilon$

Pr 1

$\underset{A}{E} \to \underbrace{\underset{A}{E}\,+T\{print('+');\}}_{\alpha} / \underset{\beta}{T}$

$E \to TR$

$R \to +T$ {print('+');} R

$R \to e$

Newly generated grammer

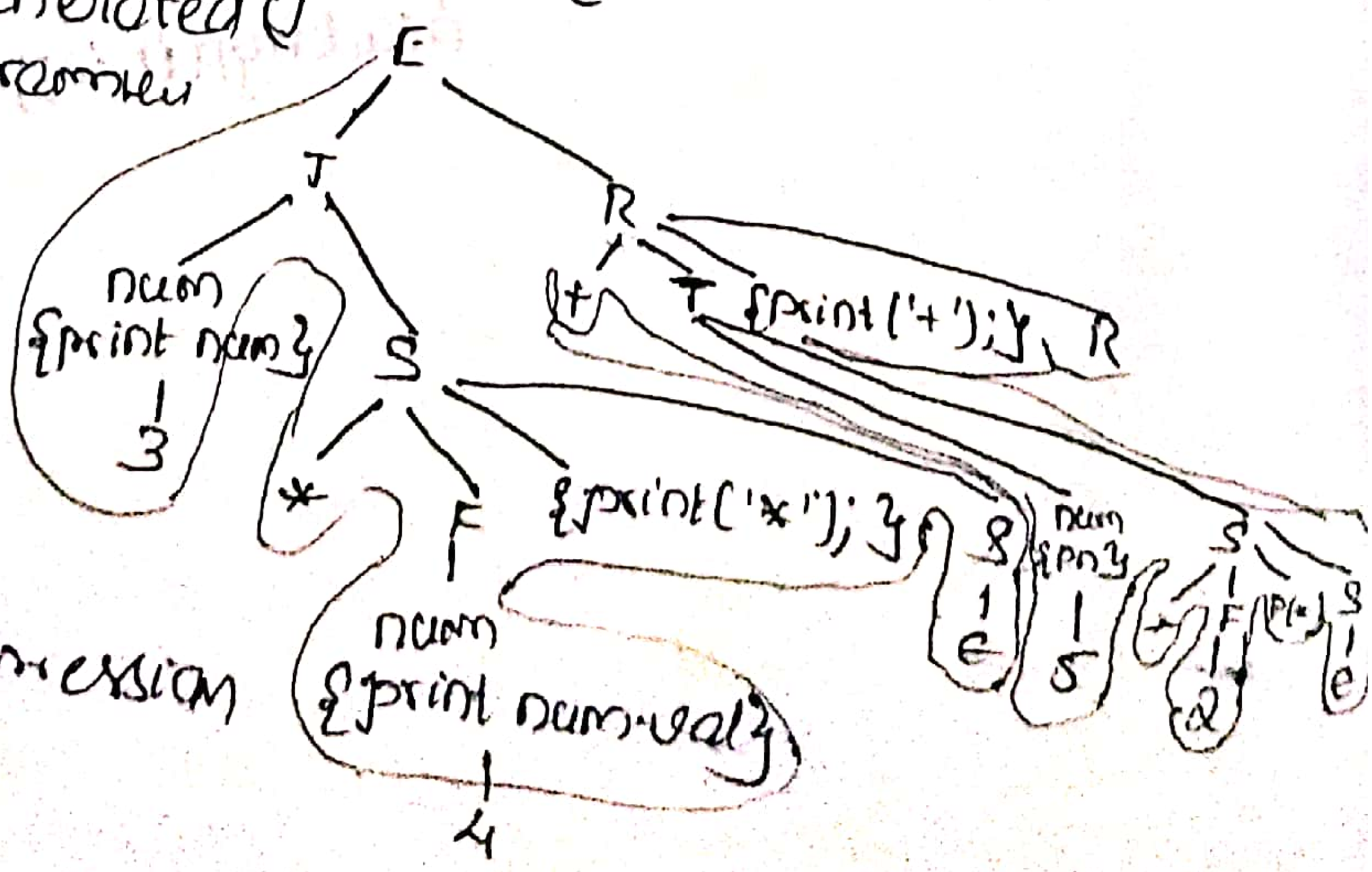
Pr 2

$\underset{A}{T} \to \underbrace{\underset{A}{T}\,*F\{print('*');\}}_{\alpha} / \underbrace{num\{print\ num\}}_{\beta}$

$T \to num$ {print num.val} S

$S \to *F$ {print('*');} S

$S \to e$



(34*52*+)

post-fix expression

4) 3-address code

i=0
sum=0
while (i<10)
    sum +=i

1. i=0
2. sum=0
3. if i≥10 goto 7
4. t= sum+i
5. sum=t
6 goto 3
7

3) quadriple, triple, indirect triple of

$$x=y+z*w/u-v$$

quadruple

	op	arg1	arg2	result
(0)	*	z	w	t_1
(1)	/	t_1	u	t_2
(2)	+	y	t_2	t_3
(3)	-	t_3	v	t_4
(4)	=	t_4		x

Three address code

$t_1=z*w$
$t_2=t_1/u$
$t_3=y+t_2$
$t_4=t_3-v$
$x=t_4$

triple

	op	arg1	arg2
(0)	*	z	w
(1)	/	(0)	u
(2)	+	y	(1)
(3)	-	(2)	v
(4)	=	x	(3)

Indirect triplet

#	statement
(0)	14
(1)	15
(2)	16
(3)	17
(4)	18

#	op	arg1	arg2
(14)	x	z	w
(15)	/	(14)	u
(16)	+	y	(15)
(17)	-	(16)	v
(18)	=	x	(17)