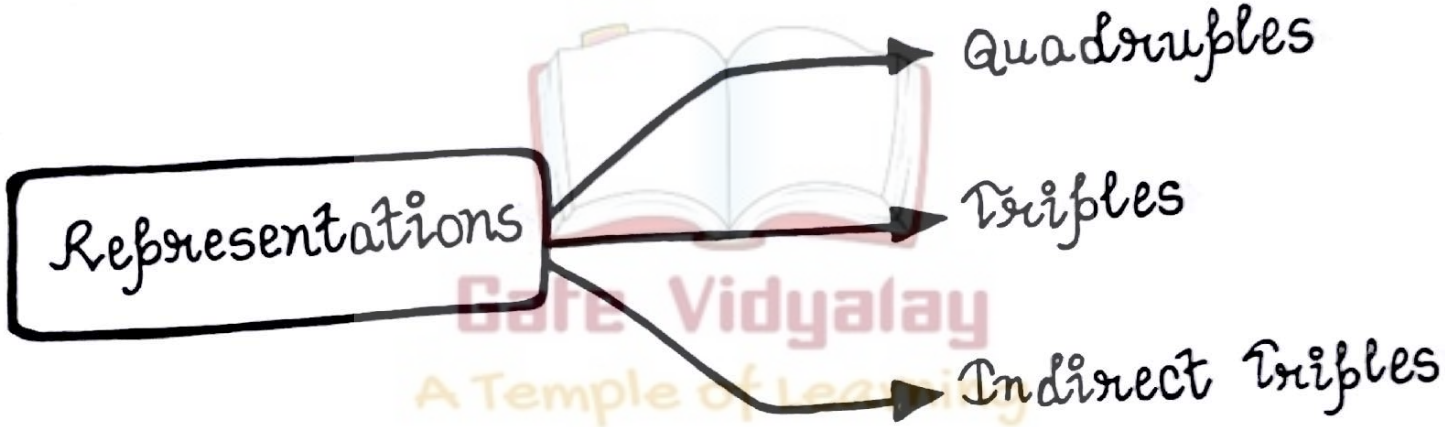
Implementation of Three Address Code

LearnVidFun....

Three address code can be implemented as a record with the address fields. There are three representations used for three address code -



1) Quadruples

In quadruple representation, each instruction is divided into four fields -

op , arg1 , arg2 , result

where -

→ The op field is used to represent the internal code for operator

→ The arg1 and arg2 fields represent the two operands used

→ The result field is used to store the result of an expression.

Exception:-

- The statement -

  $x = op\ y$ where op is a unary operator

  is represented by placing op in the operator field, y in the argument1 field and x in the result field. The argument2 field is not used.

- A statement like param t1 is represented by placing param in the operator field and t1 in the argument1 field. Neither argument2 nor the result field are used.

- Unconditional and conditional jump statements are represented by placing the target labels in the result field.

2> Triples :-

In triple representation, the use of temporary variables is avoided and instead references to instructions are made.

3> Indirect Triples :-

This representation is an enhancement over triples representation. It uses an additional instruction array to list the pointers to the triples in the desired order. Thus, it uses pointers instead of position to store results which enables the optimizers to freely reposition the sub-expression to produce an optimized code.

Illustration-01: Translate the following expression to quadruple, triple and indirect triple-

$$a + b * c / e \uparrow f + b * a$$

Solution- Three address code for the given expression is-

$$T1 = e \uparrow f$$

$$T2 = b * c$$

$$T3 = T2 / T1$$

$$T4 = b * a$$

$$T5 = a + T3$$

$$T6 = T5 + T4$$

Quadruple

Location	Op	Arg1	Arg2	Result
(0)	↑	e	f	T1
(1)	*	b	c	T2
(2)	/	T2	T1	T3
(3)	*	b	a	T4
(4)	+	a	T3	T5
(5)	+	T5	T4	T6

Triple :-

Location	Op	Arg1	Arg2
(0)	↑	e	f
(1)	*	b	c
(2)	/	(1)	(0)
(3)	*	b	a
(4)	+	a	(2)
(5)	+	(4)	(3)

Indirect Triple:-

	Statement
35	(0)
36	(1)
37	(2)
38	(3)
39	(4)
40	(5)

Location	Op	Arg1	Arg2
(0)	↑	e	f
(1)	*	b	c
(2)	/	(1)	(0)
(3)	*	b	a
(4)	+	a	(2)
(5)	+	(4)	(3)

XXXX

Illustration-02:- Translate the following expression to quadruple, triple and indirect triple-

$$a = b * -c + b * -c$$

Solution:-

Three address code for the given expression is-

$T1 = \text{uminus } c$

$T2 = b * T1$

$T3 = \text{uminus } c$

$T4 = b * T3$

$T5 = T2 + T4$

$a = T5$

Quadruple :-

Location	Op	Arg1	Arg2	Result
(1)	uminus	c		T1
(2)	*	b	T1	T2
(3)	uminus	c		T3
(4)	*	b	T3	T4
(5)	+	T2	T4	T5
(6)	=	T5		a

Triple :-

Location	Op	Arg1	Arg2
(1)	uminus	c	
(2)	*	b	(1)
(3)	uminus	c	
(4)	*	b	(3)
(5)	+	(2)	(4)
(6)	=	a	(5)

Indirect triple:-

	Statement
35	(1)
36	(2)
37	(3)
38	(4)
39	(5)
40	(6)

Location	Op	Arg1	Arg2
(1)	uminus	c	
(2)	*	b	(1)
(3)	uminus	c	
(4)	*	b	(3)
(5)	+	(2)	(4)
(6)	=	a	(5)

xxxxx