7.9 REPRESENTATION OF THREE ADDRESS STATEMENTS

There are three Representations used for three address code which are given below :

1. Quadruples
2. Triples
3. Indirect Triples.

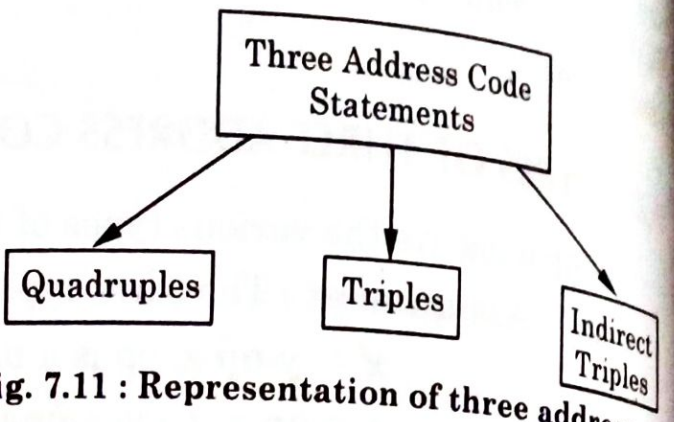

Fig. 7.11 : Representation of three address code

7.9.1 Quadruples

Quadruple is a structure which contains atmost four fields *i.e* Operator, Argument 1, Argument 2 and Result

Operator	Argument 1	Argument 2	Result

For a statement a = b + c, Quadruple Representation places + in the operator field, **a** in the Argument 1 field, **b** in Argument 2 and **c** in Result field.

For example : Consider the statement

a = b + c * d

First convert this statement in Three Address code

∴ Three Address code will be

t1 = c*d
t2 = b + t1
a = t2.

After construction of Three Address code, it will be changed to Quadruple representation as follows :

Quadruple

	Operator	arg 1	arg 2	Result
(0)	*	c	d	t1
(1)	+	b	t1	t2
(2)	=	t2		a

The contents of fields arg1, arg2 and Result are basically pointers to symbol table entries for names represented by these entries.

7.9.2 Triples

This three address code representation contain three (3) fields *i.e.* one for operator and two for Arguments (*i.e.* Argument 1 and Argument 2)

Operator	Argument 1	Argument 2

In this representation, temporary variables are not used. Instead of temporary variables we use a number in parenthesis to represent pointer to that particular record of symbol table. For example consider statement

$$a = b + c * d$$

First of all, it will be converted to Three Address code.

$$t1 = c * d$$
$$\therefore \quad t2 = b + t1$$
$$a = t2$$

Triple for this Three-Address code will be:

Triple

	Operator	arg 1	arg 2
(0)	*	c	d
(1)	+	b	(0)
(2)	=	a	(1)

Here (0) represents a pointer which refer the result c*d, which can be used in further statements *i.e.* when c*d is added with b. This result will be saved at position pointed by (1). Pointer (1) will be used further when it is assigned to a.

7.9.3 Indirect Triples

In Indirect triples, all the pointers used in Triples are indexed such that one pointer will reference another pointer & that pointer will consist of the triple.

For the previous example, Indirect triple will be

Indirect Triples

	Statement
(0)	(11)
(1)	(12)
(2)	(13)

	Operator	arg1	arg2
(11)	*	c	d
(12)	+	b	(11)
(13)	=	a	(12)

In this, we only need to refer to pointers (0), (1), (2) which will further refer pointers (11), (12), (13) respectively & then pointers (11), (12), (13) point to triples, that is why this representation is called Indirect Triple Representation.

Example 15. *Write quadruples, triples and indirect triples for the expression* $-(a + b) * (c + d) - (a + b + c)$

Ans. First of all this statement will be converted into Three Address c

$t1 = a + b$
$t2 = - t1$
$t3 = c + d$
$t4 = t2 * t3$
$t5 = t1 + c$
$t6 = t4 - t5$

Quadruple :

	Operator	arg1	arg2	Result
(0)	+	a	b	t1
(1)	–	t1		t2
(2)	+	c	d	t3
(3)	*	t2	t3	t4.
(4)	+	t1	c	t5
(5)	–	t4	t5	t6

Triple

	Operator	arg1	arg2
(0)	+	a	b
(1)	–	(0)	
(2)	+	c	d
(3)	*	(1)	(2)
(4)	+	(0)	c
(5)	–	(3)	(4)

Indirect Triple

	Statement
(0)	(11)
(1)	(12)
(2)	(13)
(3)	(14)
(4)	(15)
(5)	(16)

	Operator	arg1	arg2
(11)	+	a	b
(12)	–	(11)	
(13)	+	c	d
(14)	*	(12)	(13)
(15)	+	(11)	c
(16)	–	(14)	(15)