

## PRINCIPAL SOURCES OF OPTIMIZATION - FUNCTION PRESERVING TRANSFORMATIONS

### Function preserving transformations:

There are number of ways to improve the program without changing the function it computes. They are

1. Common subexpression elimination
2. Copy propagation
3. Dead-code elimination

Transformations:

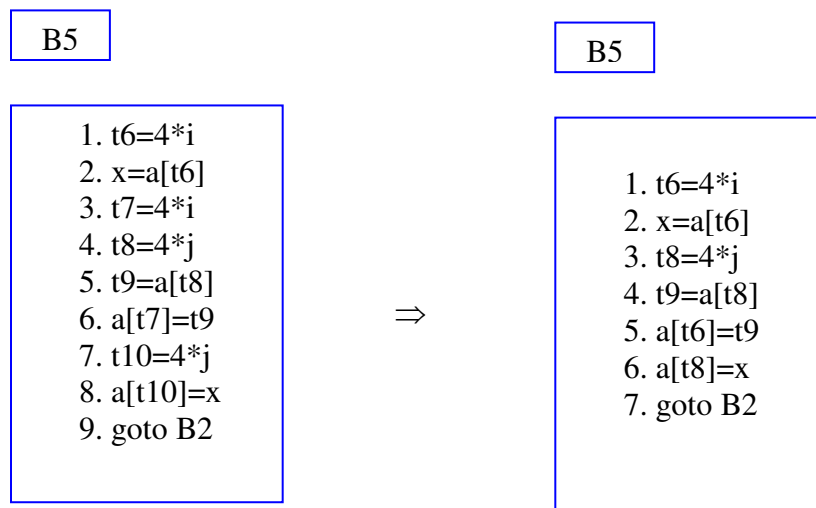
Transformation of a program is called **local** if it can be performed by looking only at the statements in a basic block, otherwise it is **global**.

### 1. Common subexpression elimination

Common subexpression: It is an expression E whose value is already computed and the values of variables in E have not changed since the previous computation is known as common sub expression.

Consider the block B5 and apply this transformation locally

*Local Transformation for the basic block B5*

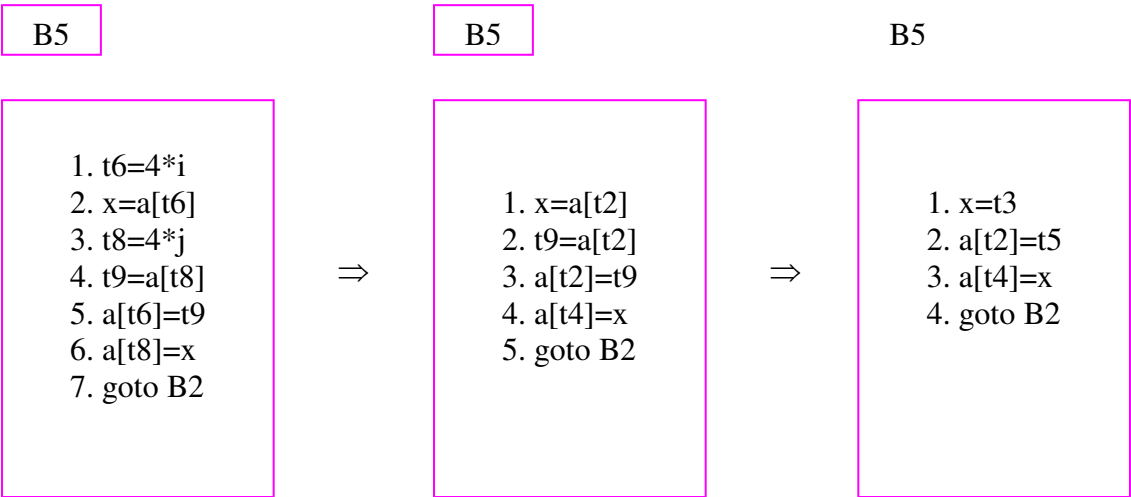


**Explanation:**

- TAC 3 is eliminated, because of the common sub expression 4\*i.
- TAC 7 is eliminated, because of the common sub expression 4\*j.
- Substitute t7 by t6 and t10 by t8, in all other TAC sequences will yield the transformed block B5

Same way do for the block B6 also the local transformation.

**Global Transformation for the basic block B5**



**Explanation:**

- TAC 1 is eliminated, because of the common sub expression 4\*i which was computed in blockB2.
- TAC 8 is eliminated, because of the common sub expression 4\*j which was computed in blockB3.
- .Substitute t6 by t2 and t8 by t4, in all other TAC sequences will yield the transformed block B5
- a[t6] is already computed so it can be replaced by t3
- a[t4] is already computed and the result is available in t5 so TAC 2 can be eliminated
- replace the necessary substitutions depends on the updation

Same way do for the block B6 also the local transformation.

## **2. Copy Propagation**

Removal of common subexpression leads to copy statement. It means that the value of that subexpression is available in more than one variable. So any variable can be used after that for reference.

Example

$x = a[t2]$  replaced by  $x = t3$ , after this copy, at any point value of  $a[t2]$  can be referred by  $x$  or  $t3$ .

## **3. Dead code elimination**

The code which is not useful can be eliminated. Ex. Unlabeled instruction after the unconditional branch instruction is dead code. Continuing the above discussion the value of the expression is available in  $t3$ , so the copy statement is dead. So it is eliminated. Transformed block B5 is

B5

1.  $a[t2] = t5$
2.  $a[t4] = x$
3. goto B2