



## CD(UNIT -5) - Compiler design in jntuh helps in semester exams with neat explanation

Compiler Design (Jawaharlal Nehru Technological University, Hyderabad)



Scan to open on Studocu

## Unit - 5

### Principal source of optimization

Refer to Unit - 4

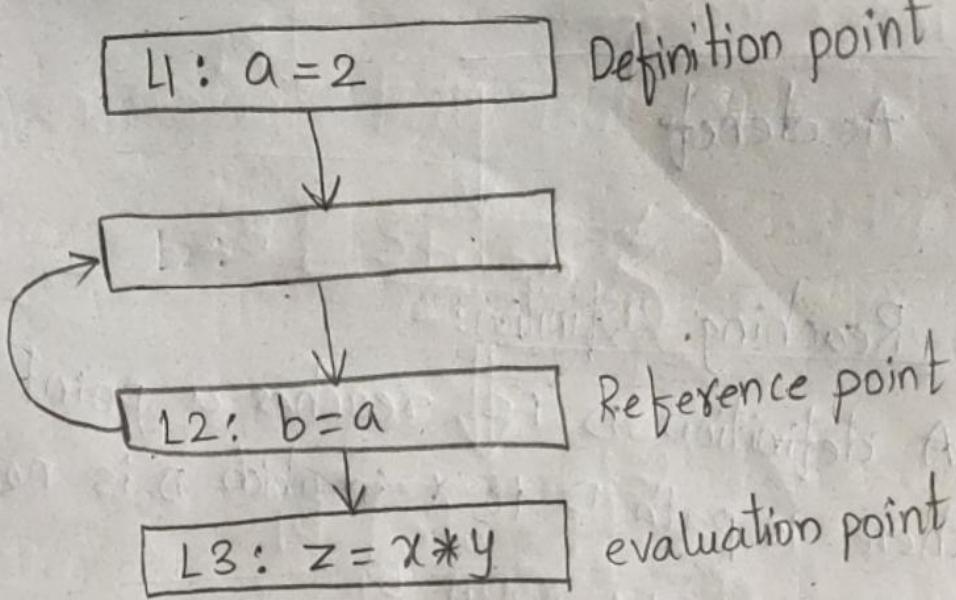
- optimization of Basic Block
  - peephole optimization
- } only contents i.e inside topics

### Data flow Analysis :-

- \* It is the analysis of flow of data in the control flow graph.
- \* The analysis that determines the information regarding the definition and use of data in the program.
- \* with the help of this analysis optimization can be done.

### Basic technologies :-

- Definition point :- A point in a program containing some definition
- Reference point :- A point in a program containing a reference to a data item
- Evaluation point :- A point in a program containing evaluation of expression.

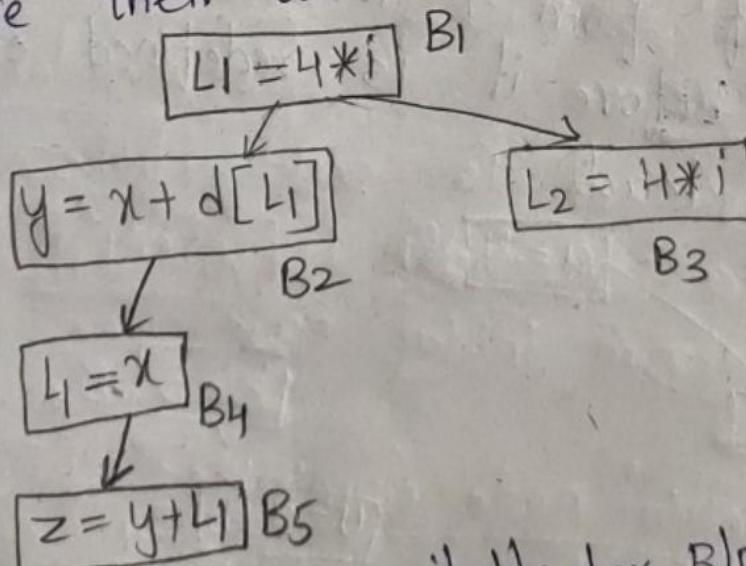


Dataflow properties:-

\* ~~An expr~~ Available expression

\* An expression is said to available at a program point  $x$  if along path it is reaching to  $x$ .

\* An expression  $a+b$  is said to be available if none of the operands gets modified before their use.

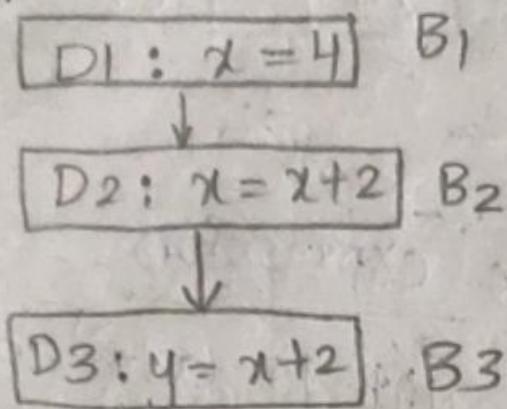


- \* Expression  $4*i$  is available for Block B<sub>2</sub>, B<sub>3</sub>
- \* But at Block B<sub>4</sub> its gets modified
- \* At Block B<sub>5</sub> it is unavailable.

## Reaching Definition:-

A definition D is reaches a point x if there is path D to x in which D is not killed

ex:-

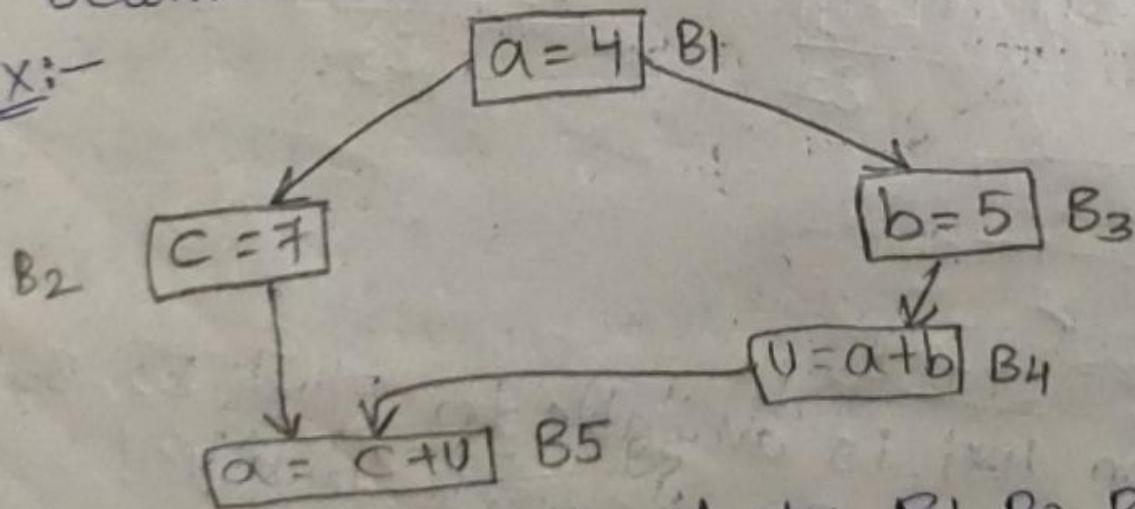


D1 is reaching definition for B2 but not for B3 (As x is modified / killed at D2)

## Live Variable:-

\* A variable is said to be live at some point P if from P to end of the variable is used before it is redefined else it becomes dead.

ex:-

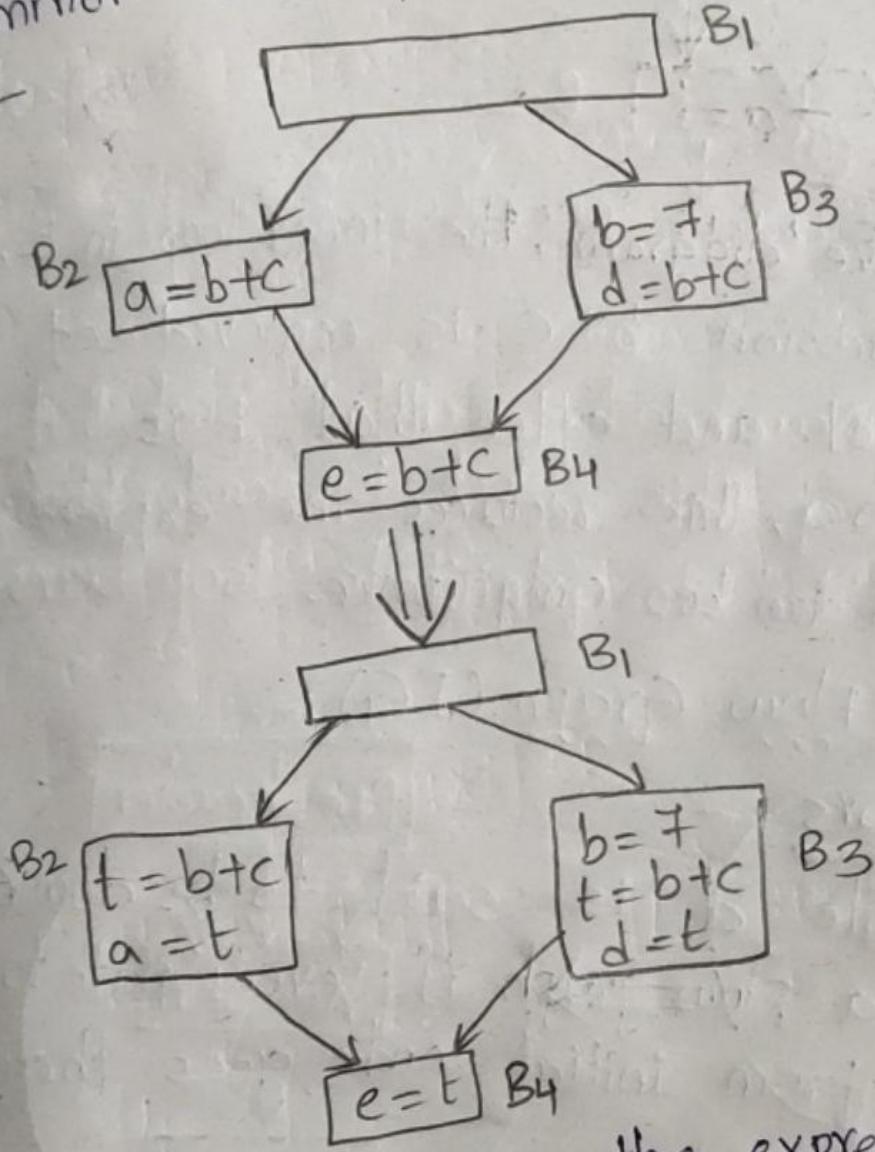


Here a is live at block: B1, B2, B3, B4

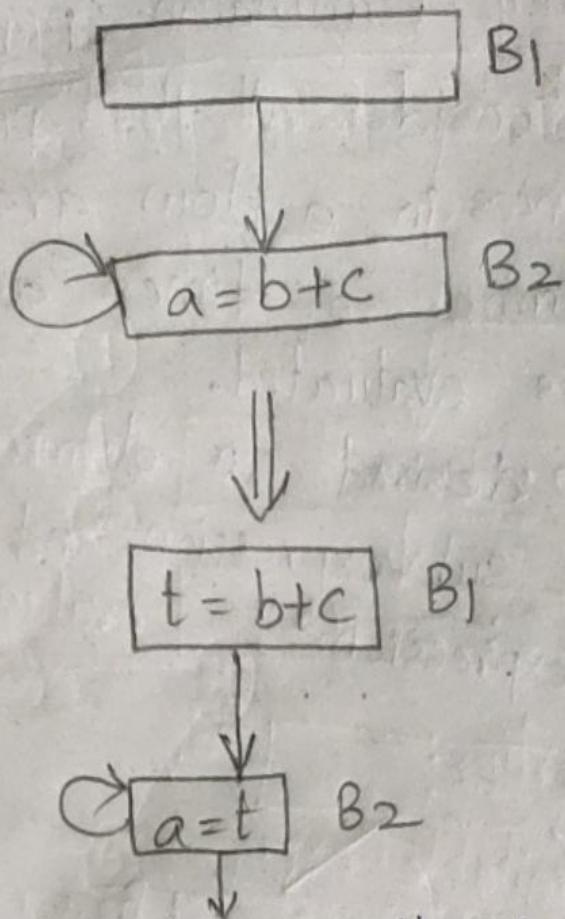
## Partial Redundancy Elimination (PRE):-

- \* In PRE, we consider all the possible execution sequences in a flow graph, and look at the number of times an expression such as  $x, y$  is evaluated.
- \* PRE can be desired to eliminate both global common sub expression and local common sub expression.

ex:-



In the above diagram, the expression  $b + c$  computed in block B4 is redundant. So in block B2 and B3,  $t = b + c$  is added and in block B4,  $t$  is assigned to  $e = t$ .



In the above diagram, the inner loop in B2,  
 if the expression  $b + c$  is removed and placed in B1 and after that  $t$  is assigned to  $a$ , this reduces the expression evaluated to be only once.

Loops in Flow Graph (FG) :-

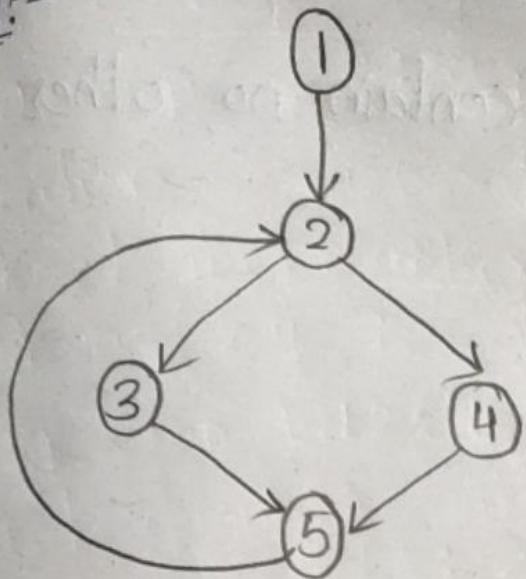
i) Dominators :-

A node  $d$  is said to be dominant node in a flow graph if every path to none  $n$  from initial node goes through  $d$  only.

- \* Every initial node dominates all the remaining nodes in a Flow Graph.

- \* Every node dominates itself.

ex:-

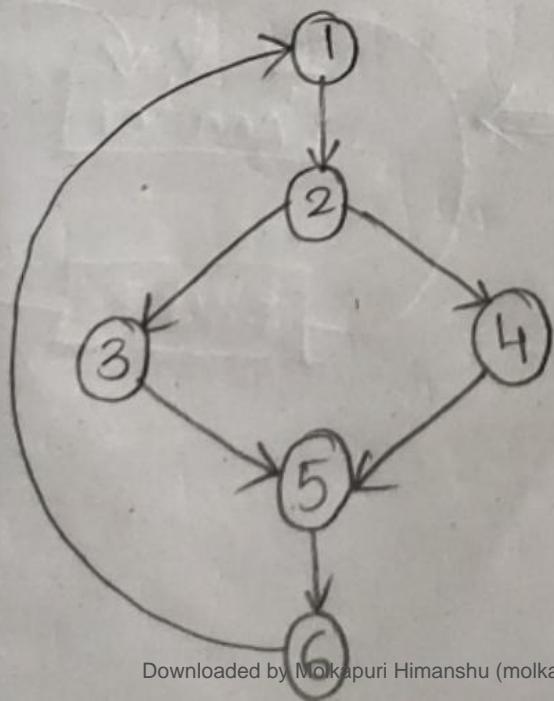


Node 1	dominates	2, 3, 4 and 5
Node 2	dominates	3, 4, 5
Node 3	dominates	only it self
Node 4	dominates	only it self
Node 5	dominates	only it self

## 2) Natural Loop :-

A Natural loop can be defined by a back edge  $n \rightarrow d$  such that there exists a collection of all the nodes that can reach to  $n$  without going through  $d$ .

ex:-



Natural loops:-

$$6 \rightarrow 1$$

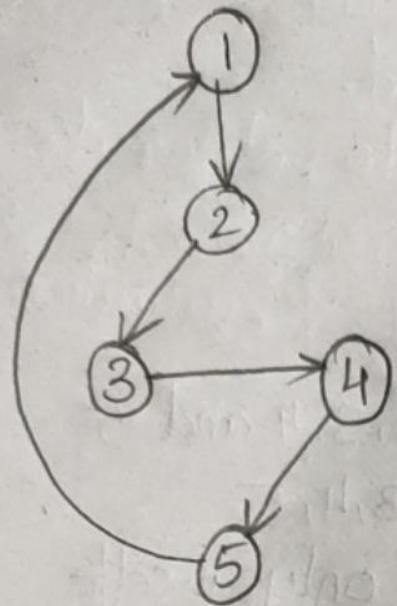
$$n \rightarrow d$$

$$\{2, 3, 4, 5, 6, 1\}$$

## Inner loop:-

It is a loop that contain no other loop.

ex:-



Here 2 to 4 is inner loop  $\{2, 3, 4\}$

## Pre header:-

It is a new header related such that successor of this block is header block.  
\* It is added to facilitate loop transformations.

