

## Introduction to Data Flow Analysis

GCC Resource Center  
([www.cse.iitb.ac.in/grc](http://www.cse.iitb.ac.in/grc))

Department of Computer Science and Engineering,  
Indian Institute of Technology, Bombay



July 2010

Notes

---

### Outline

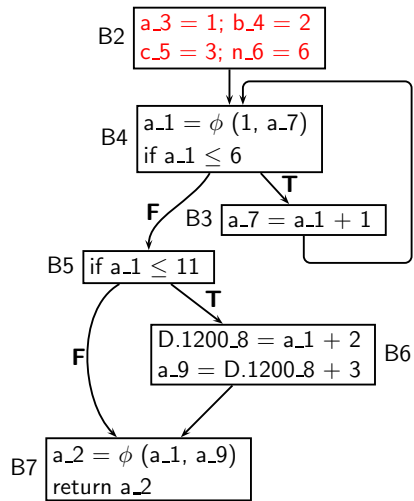
- Motivation
- Live Variables Analysis
- Available Expressions Analysis

*Part 2*

*Motivation*



## Dead Code Elimination



- No uses for variables  $a_3$ ,  $b_4$ ,  $c_5$ , and  $n_6$
- Assignments to these variables can be deleted

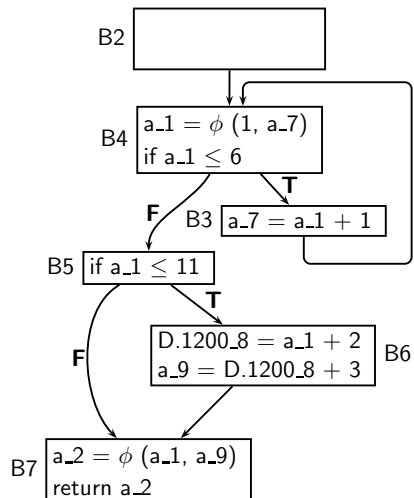


## Dead Code Elimination

## Notes



## Dead Code Elimination



- No uses for variables  $a_3$ ,  $b_4$ ,  $c_5$ , and  $n_6$
- Assignments to these variables can be deleted

How can we conclude this systematically?



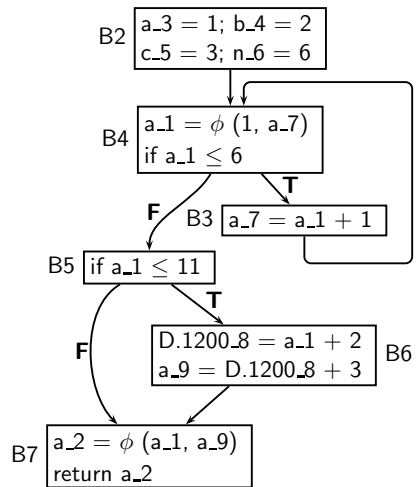
## Dead Code Elimination

## Notes



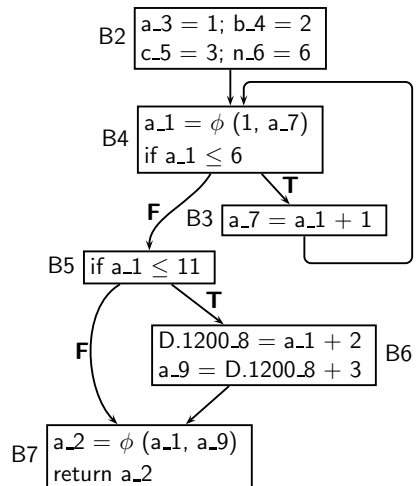
## Liveness Analysis of Variables

Find out at each program point  $p$ , the variables that are used beyond  $p$



## Liveness Analysis of Variables

Find out at each program point  $p$ , the variables that are used beyond  $p$



Which variables are used beyond this point?

$\emptyset$



## Liveness Analysis of Variables

# Notes



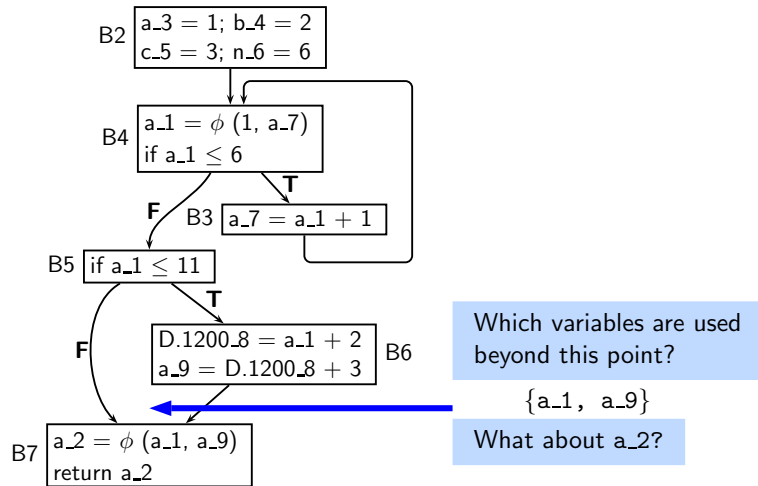
## Liveness Analysis of Variables

# Notes



## Liveness Analysis of Variables

Find out at each program point  $p$ , the variables that are used beyond  $p$



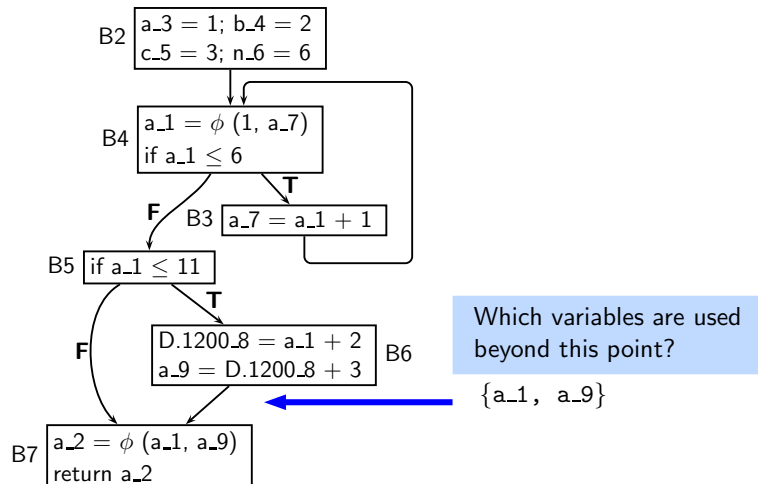
## Liveness Analysis of Variables

# Notes



## Liveness Analysis of Variables

Find out at each program point  $p$ , the variables that are used beyond  $p$



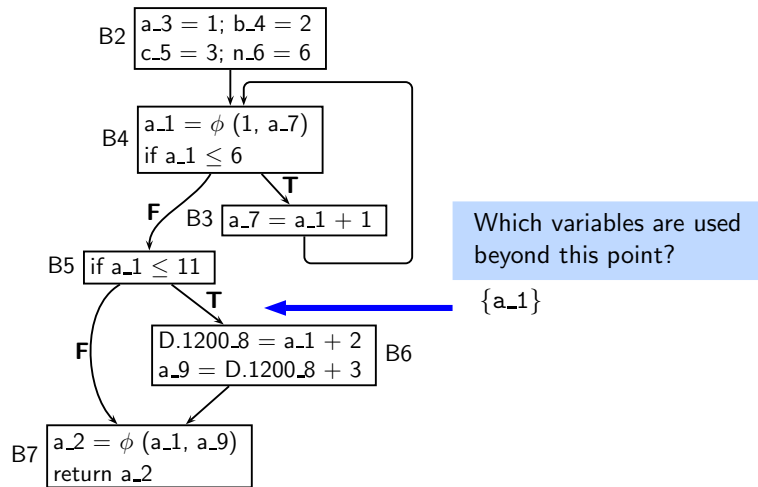
## Liveness Analysis of Variables

# Notes



## Liveness Analysis of Variables

Find out at each program point  $p$ , the variables that are used beyond  $p$



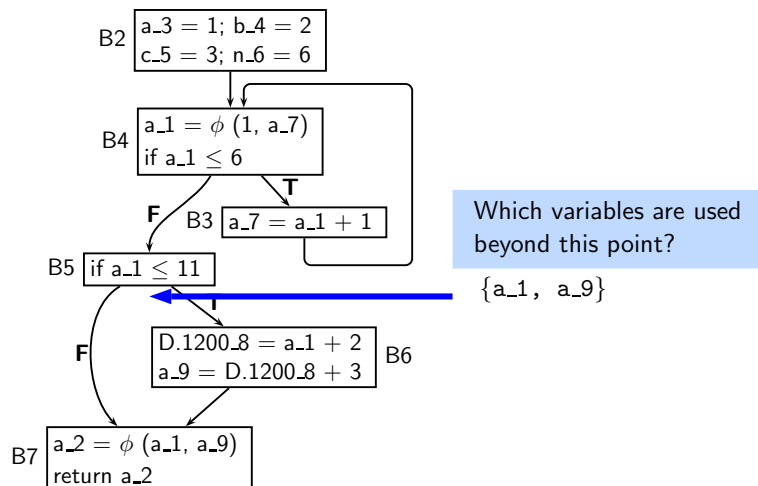
## Liveness Analysis of Variables

# Notes



## Liveness Analysis of Variables

Find out at each program point  $p$ , the variables that are used beyond  $p$



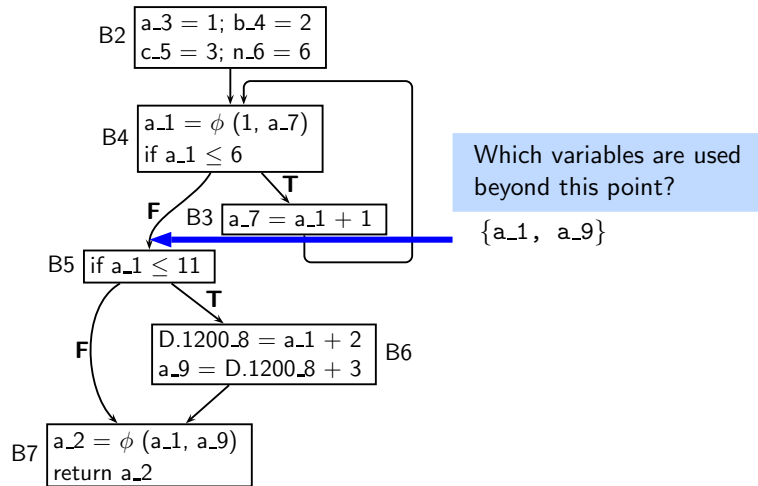
## Liveness Analysis of Variables

# Notes



## Liveness Analysis of Variables

Find out at each program point  $p$ , the variables that are used beyond  $p$



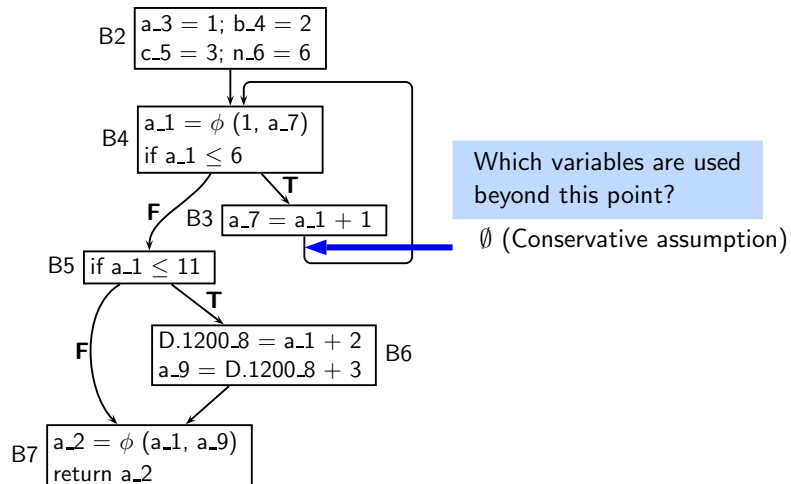
## Liveness Analysis of Variables

# Notes



## Liveness Analysis of Variables

Find out at each program point  $p$ , the variables that are used beyond  $p$



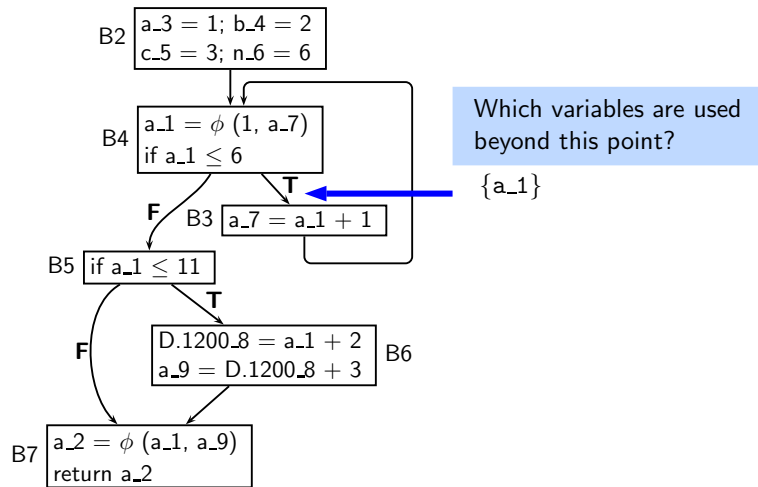
## Liveness Analysis of Variables

# Notes



## Liveness Analysis of Variables

Find out at each program point  $p$ , the variables that are used beyond  $p$



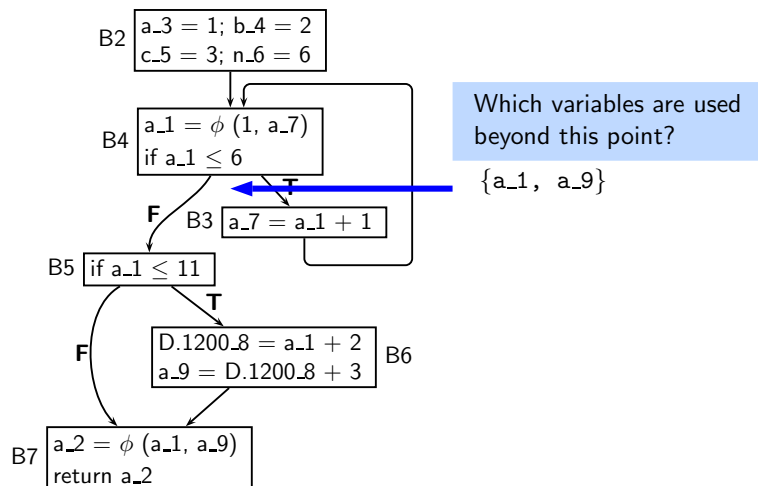
## Liveness Analysis of Variables

## Notes



## Liveness Analysis of Variables

Find out at each program point  $p$ , the variables that are used beyond  $p$



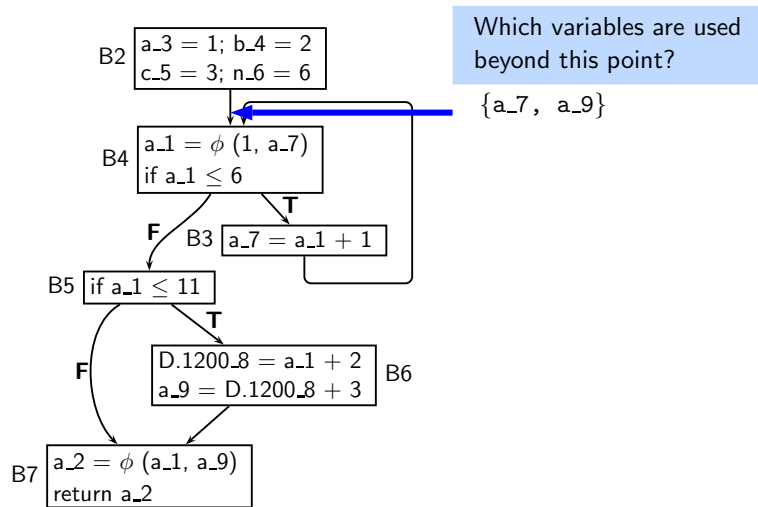
## Liveness Analysis of Variables

## Notes



## Liveness Analysis of Variables

Find out at each program point  $p$ , the variables that are used beyond  $p$



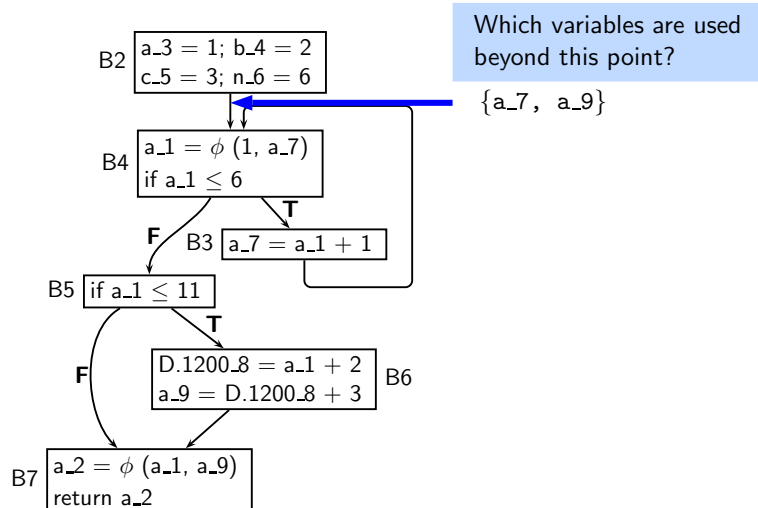
## Liveness Analysis of Variables

# Notes



## Liveness Analysis of Variables

Find out at each program point  $p$ , the variables that are used beyond  $p$



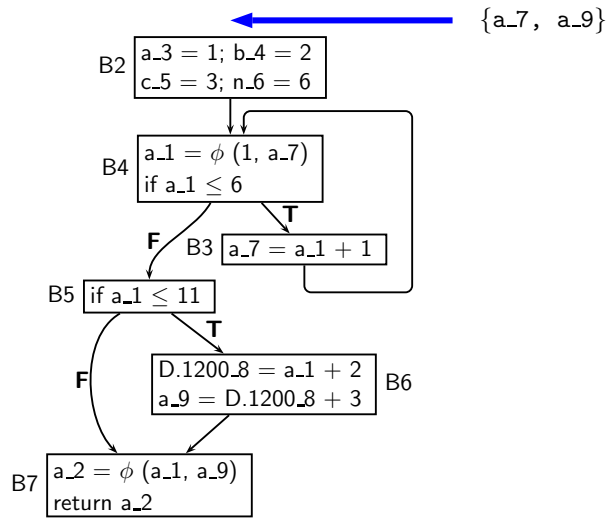
## Liveness Analysis of Variables

# Notes



### Liveness Analysis of Variables

Find out at each program point  $p$ , the variables that are used beyond  $p$



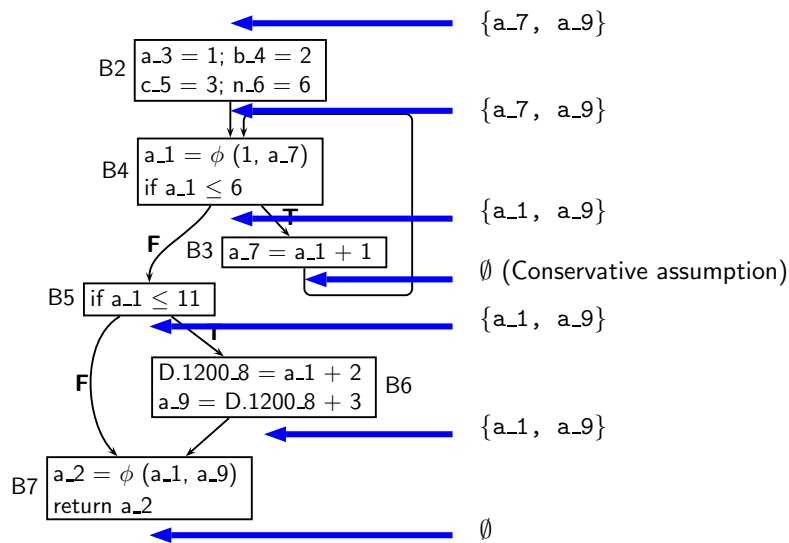
### Liveness Analysis of Variables

Notes



### Liveness Analysis of Variables

Find out at each program point  $p$ , the variables that are used beyond  $p$



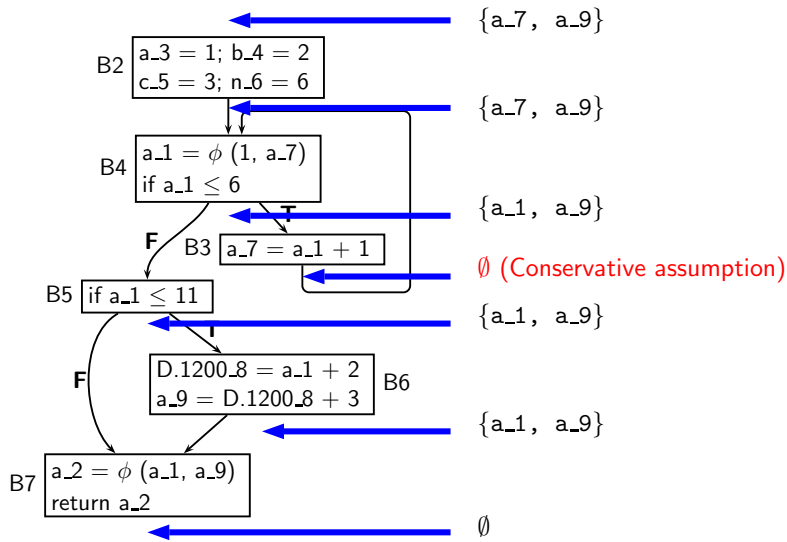
### Liveness Analysis of Variables

Notes



### Liveness Analysis of Variables: Iteration 2

Find out at each program point  $p$ , the variables that are used beyond  $p$



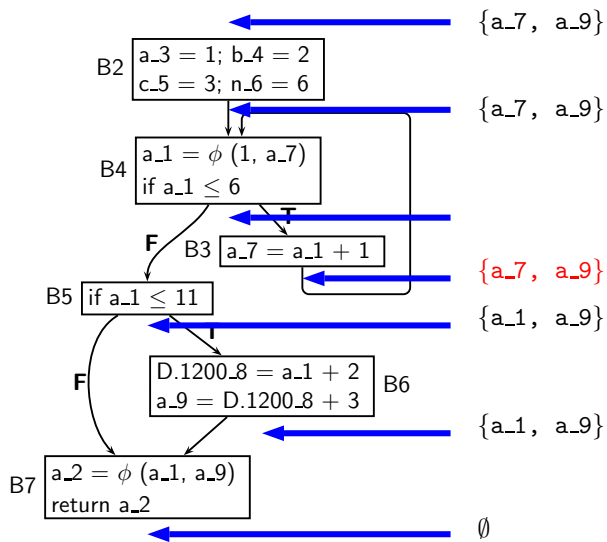
### Liveness Analysis of Variables: Iteration 2

Notes



### Liveness Analysis of Variables: Iteration 2

Find out at each program point  $p$ , the variables that are used beyond  $p$

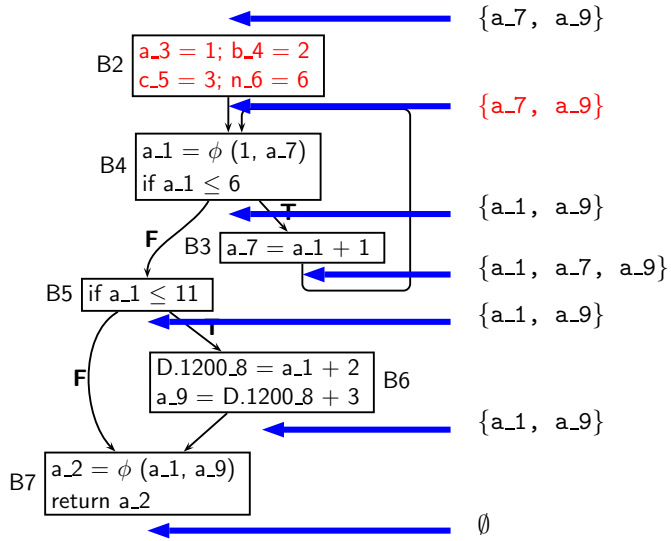


### Liveness Analysis of Variables: Iteration 2

Notes



## Using Liveness Analysis for Dead Code Elimination

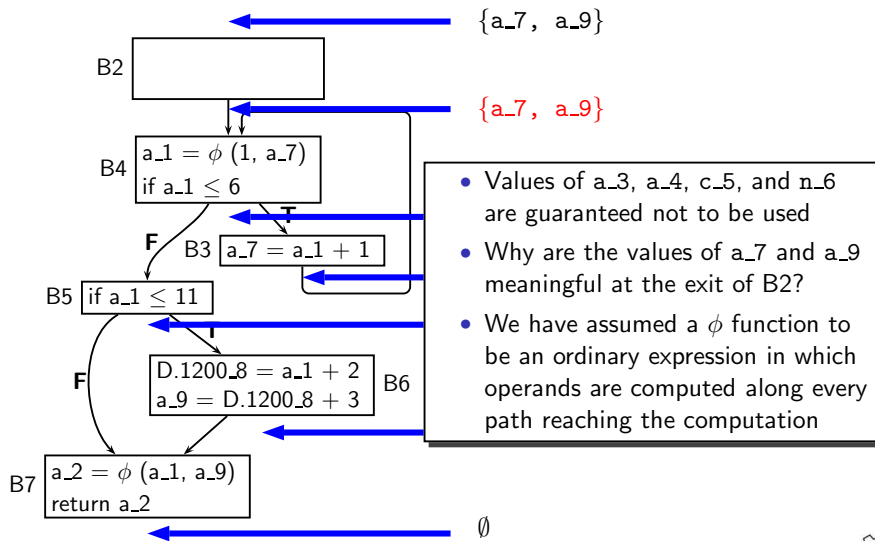


## Using Liveness Analysis for Dead Code Elimination

Notes



## Using Liveness Analysis for Dead Code Elimination

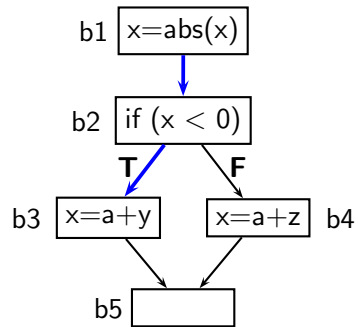


## Using Liveness Analysis for Dead Code Elimination

Notes



## Conservative Nature of Analysis (1)



- $\text{abs}(n)$  returns the absolute value of  $n$
- Is  $y$  live on entry to block  $b2$ ?
- By execution semantics, no  
Path  $b1 \rightarrow b2 \rightarrow b3$  is an infeasible execution path
- A compiler make conservative assumptions: *All branch outcomes are possible*  
 $\Rightarrow$  Consider every path in CFG as a potential execution execution path
- Our analysis concludes that  $y$  is live on entry to block  $b2$

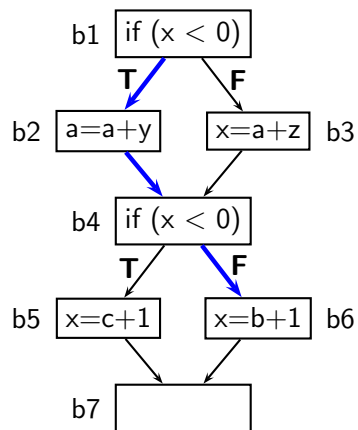


## Conservative Nature of Analysis (1)

# Notes



## Conservative Nature of Analysis (2)



- Is  $b$  live on entry to block  $b2$ ?
- By execution semantics, no  
Path  $b1 \rightarrow b2 \rightarrow b4 \rightarrow b6$  is an infeasible execution path
- Is  $c$  live on entry to block  $b3$ ?
- Path  $b1 \rightarrow b3 \rightarrow b4 \rightarrow b6$  is a feasible execution path
- A compiler make conservative assumptions  
 $\Rightarrow$  our analysis is *path insensitive*  
Note: It is *flow sensitive* (i.e. information is computed for every control flow points)
- Our analysis concludes that  $b$  is live at the entry of  $b2$  and  $c$  is live at the entry of  $b3$



## Conservative Nature of Analysis (2)

# Notes



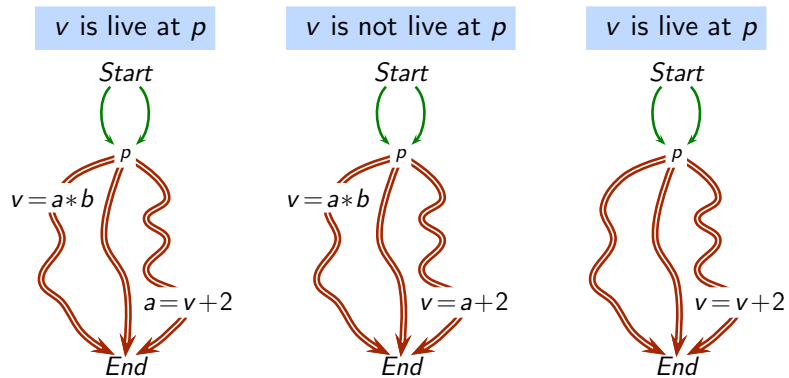
# Live Variables Analysis

Notes

## Defining Live Variables Analysis

A variable  $v$  is live at a program point  $p$ , if **some** path from  $p$  to program exit contains an r-value occurrence of  $v$  which is not preceded by an l-value occurrence of  $v$ .

Path based specification

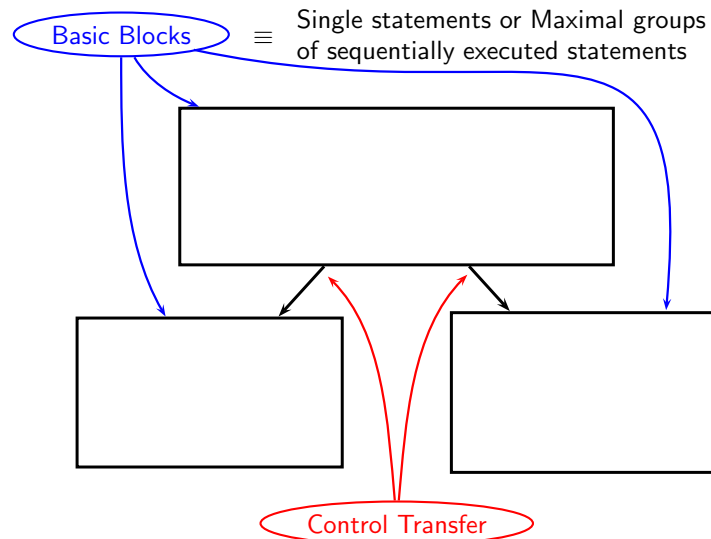


## Defining Live Variables Analysis

Notes



## Defining Data Flow Analysis for Live Variables Analysis

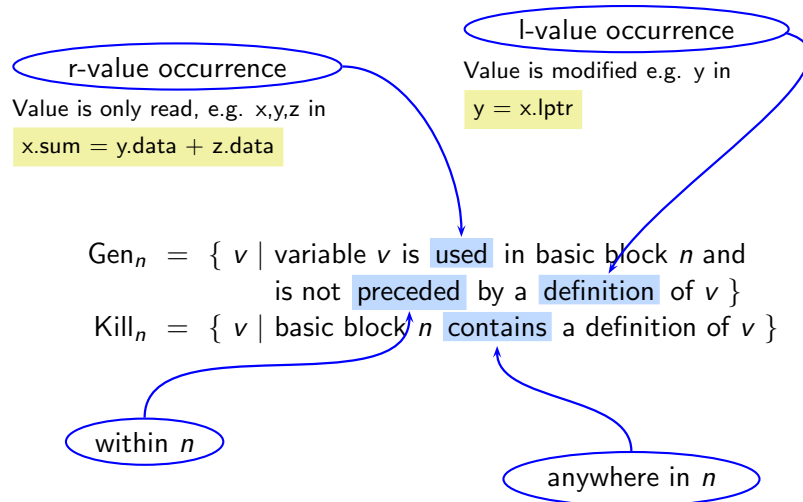


## Defining Data Flow Analysis for Live Variables Analysis

# Notes



## Local Data Flow Properties for Live Variables Analysis



## Local Data Flow Properties for Live Variables Analysis

# Notes

## Local Data Flow Properties for Live Variables Analysis

- $Gen_n$  : Use not preceded by definition

Upwards exposed use

- $Kill_n$  : Definition anywhere in a block

Stop the effect from being propagated across a block



## Local Data Flow Properties for Live Variables Analysis

Case	Local Information		Effect on Liveness
1	$v \notin Gen_n$	$v \notin Kill_n$	Liveness of $v$ is unaffected in block $n$
2	$v \in Gen_n$	$v \notin Kill_n$	Liveness of $v$ is generated in block $n$
3	$v \notin Gen_n$	$v \in Kill_n$	Liveness of $v$ is killed in block $n$
4	$v \in Gen_n$	$v \in Kill_n$	Liveness of $v$ is killed in block $n$ but is re-generated in the same block



## Local Data Flow Properties for Live Variables Analysis

# Notes

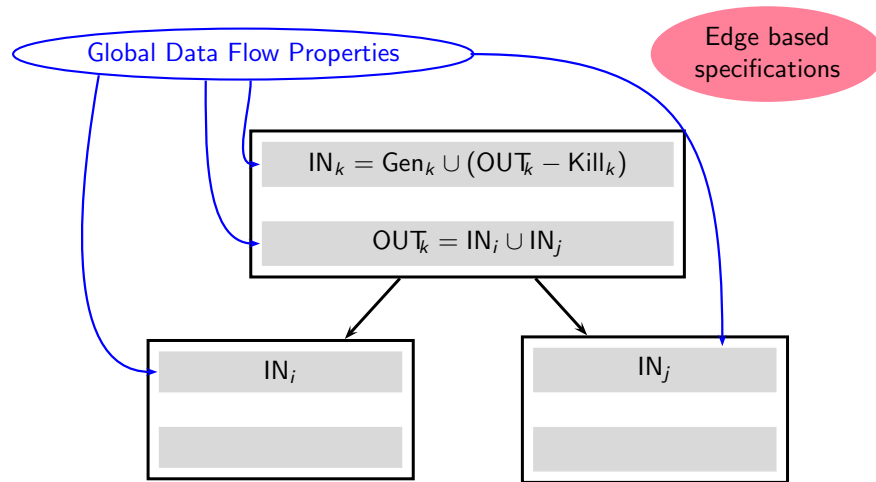


## Local Data Flow Properties for Live Variables Analysis

# Notes



## Defining Data Flow Analysis for Live Variables Analysis



## Data Flow Equations For Live Variables Analysis

$$IN_n = (OUT_n - Kill_n) \cup Gen_n$$

$$OUT_n = \begin{cases} BI & n \text{ is End block} \\ \bigcup_{s \in succ(n)} IN_s & \text{otherwise} \end{cases}$$

$IN_n$  and  $OUT_n$  are sets of variables.



## Defining Data Flow Analysis for Live Variables Analysis

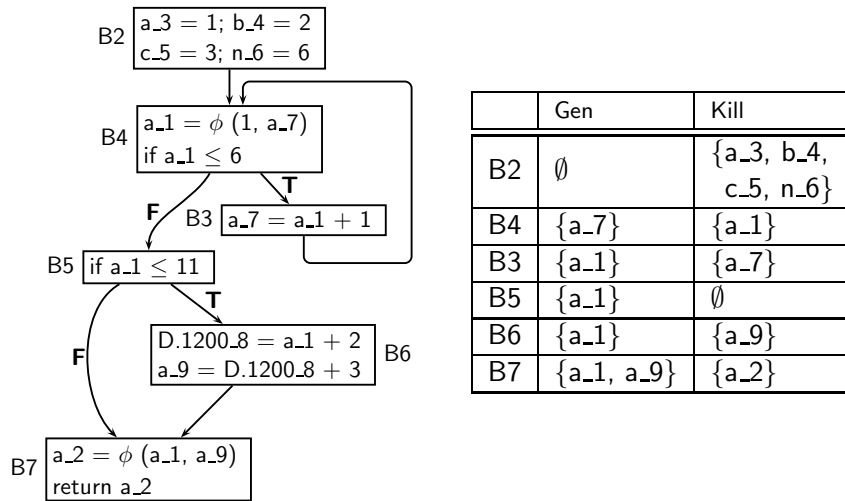
Notes

## Data Flow Equations For Live Variables Analysis

Notes



### Performing Live Variables Analysis

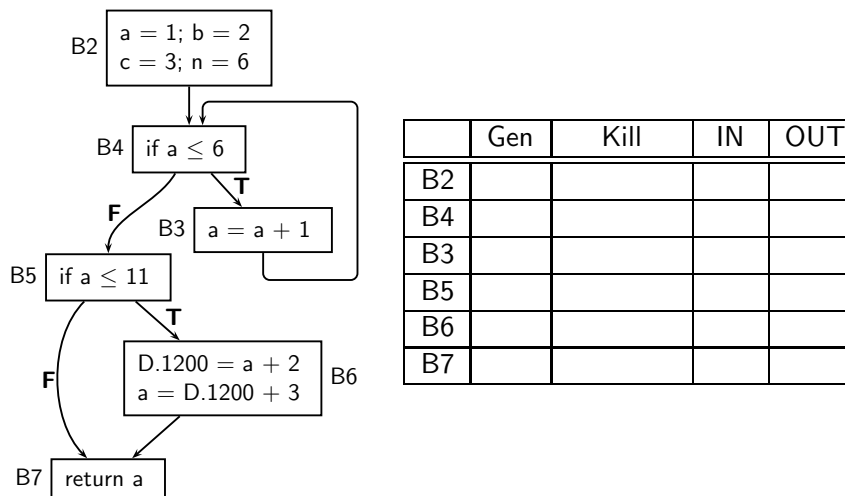


### Performing Live Variables Analysis

Notes



### Tutorial Problem for Live Variables Analysis



### Tutorial Problem for Live Variables Analysis

Notes



## Using Data Flow Information of Live Variables Analysis

- Used for register allocation.  
If variable  $x$  is live in a basic block  $b$ , it is a potential candidate for register allocation.
- Used for dead code elimination.  
If variable  $x$  is not live after an assignment  $x = \dots$ , then the assignment is redundant and can be deleted as dead code.



## Using Data Flow Information of Live Variables Analysis

# Notes



## Using Data Flow Information of Live Variables Analysis

# Notes



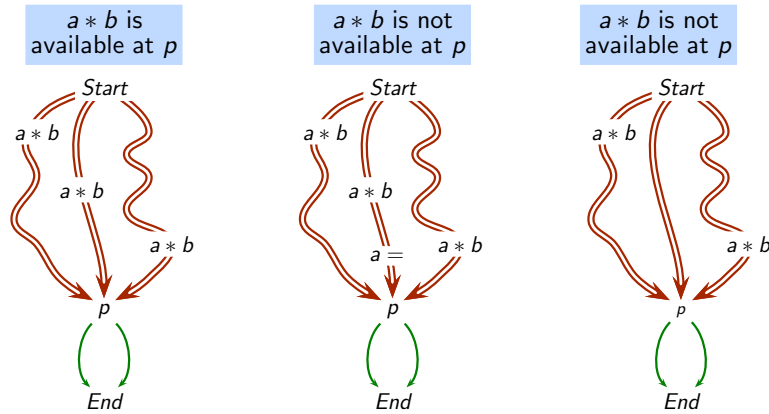
Part 4

*Available Expressions Analysis*



## Defining Available Expressions Analysis

An expression  $e$  is available at a program point  $p$ , if every path from program entry to  $p$  contains an evaluation of  $e$  which is not followed by a definition of any operand of  $e$ .



## Defining Available Expressions Analysis

# Notes



## Local Data Flow Properties for Available Expressions Analysis

$Gen_n = \{ e \mid \text{expression } e \text{ is evaluated in basic block } n \text{ and this evaluation is not followed by a definition of any operand of } e \}$

$Kill_n = \{ e \mid \text{basic block } n \text{ contains a definition of an operand of } e \}$

	Entity	Manipulation	Exposition
$Gen_n$	Expression	Use	Downwards
$Kill_n$	Expression	Modification	Anywhere



## Local Data Flow Properties for Available Expressions Analysis

# Notes



## Data Flow Equations For Available Expressions Analysis

$$IN_n = \begin{cases} BI & n \text{ is Start block} \\ \bigcap_{p \in \text{pred}(n)} OUT_p & \text{otherwise} \end{cases}$$

$$OUT_n = Gen_n \cup (IN_n - Kill_n)$$

Alternatively,

$$OUT_n = f_n(IN_n), \quad \text{where}$$

$$f_n(X) = Gen_n \cup (X - Kill_n)$$

$IN_n$  and  $OUT_n$  are sets of expressions.



## Using Data Flow Information of Available Expressions Analysis

- Common subexpression elimination
  - ▶ If an expression is available at the entry of a block ***b*** and
  - ▶ a computation of the expression exists in ***b*** **such that**
  - ▶ it is not preceded by a definition of any of its operands

Then the expression is redundant

- Redundant expression must be **upwards exposed**
- Expressions in  $Gen_n$  are **downwards exposed**



## Data Flow Equations For Available Expressions Analysis

# Notes

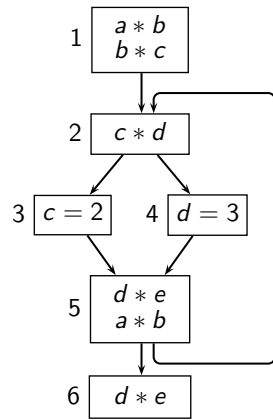


## Using Data Flow Information of Available Expressions Analysis

# Notes



# An Example of Available Expressions Analysis



Let  $e_1 \equiv a * b$ ,  $e_2 \equiv b * c$ ,  $e_3 \equiv c * d$ ,  $e_4 \equiv d * e$

Node	Computed	Killed	Available	Redund.
1	{ $e_1, e_2$ }	1100	$\emptyset$	0000
2	{ $e_3$ }	0010	$\emptyset$	0000
3	$\emptyset$	0000	{ $e_2, e_3$ }	0110
4	$\emptyset$	0000	{ $e_3, e_4$ }	0011
5	{ $e_1, e_4$ }	1001	$\emptyset$	0000
6	{ $e_4$ }	0001	$\emptyset$	0000



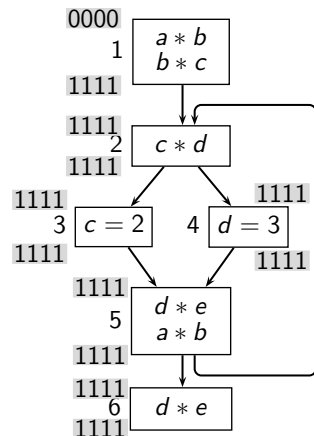
# An Example of Available Expressions Analysis

# Notes



# An Example of Available Expressions Analysis

## Initialisation



Let  $e_1 \equiv a * b$ ,  $e_2 \equiv b * c$ ,  $e_3 \equiv c * d$ ,  $e_4 \equiv d * e$

Node	Computed	Killed	Available	Redund.
1	{ $e_1, e_2$ }	1100	$\emptyset$	0000
2	{ $e_3$ }	0010	$\emptyset$	0000
3	$\emptyset$	0000	{ $e_2, e_3$ }	0110
4	$\emptyset$	0000	{ $e_3, e_4$ }	0011
5	{ $e_1, e_4$ }	1001	$\emptyset$	0000
6	{ $e_4$ }	0001	$\emptyset$	0000



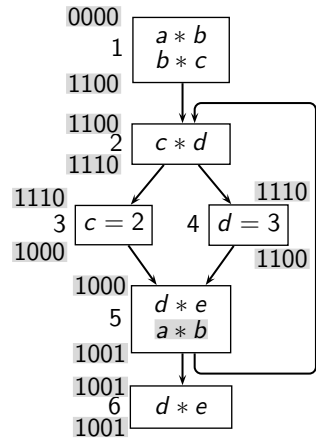
# An Example of Available Expressions Analysis

# Notes



## An Example of Available Expressions Analysis

### Iteration #1



Let  $e_1 \equiv a * b$ ,  $e_2 \equiv b * c$ ,  $e_3 \equiv c * d$ ,  $e_4 \equiv d * e$

Node	Computed	Killed	Available	Redund.
1	{ $e_1, e_2$ }	1100	$\emptyset$	0000
2	{ $e_3$ }	0010	$\emptyset$	0000
3	$\emptyset$	0000	{ $e_2, e_3$ }	0110
4	$\emptyset$	0000	{ $e_3, e_4$ }	0011
5	{ $e_1, e_4$ }	1001	$\emptyset$	0000
6	{ $e_4$ }	0001	$\emptyset$	0000



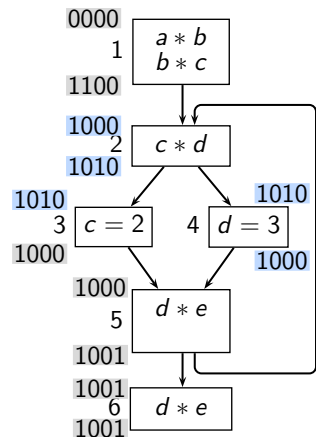
## An Example of Available Expressions Analysis

# Notes



## An Example of Available Expressions Analysis

### Iteration #2



Let  $e_1 \equiv a * b$ ,  $e_2 \equiv b * c$ ,  $e_3 \equiv c * d$ ,  $e_4 \equiv d * e$

Node	Computed	Killed	Available	Redund.
1	{ $e_1, e_2$ }	1100	$\emptyset$	0000
2	{ $e_3$ }	0010	$\emptyset$	0000
3	$\emptyset$	0000	{ $e_2, e_3$ }	0110
4	$\emptyset$	0000	{ $e_3, e_4$ }	0011
5	{ $e_1, e_4$ }	1001	$\emptyset$	0000
6	{ $e_4$ }	0001	$\emptyset$	0000



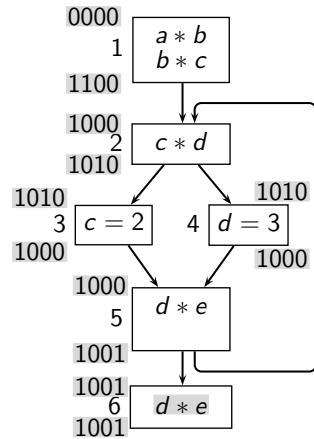
## An Example of Available Expressions Analysis

# Notes



## An Example of Available Expressions Analysis

### Final Result



Let  $e_1 \equiv a * b$ ,  $e_2 \equiv b * c$ ,  $e_3 \equiv c * d$ ,  $e_4 \equiv d * e$

Node	Computed	Killed	Available	Redund.
1	$\{e_1, e_2\}$	$\emptyset$	$\emptyset$	$\emptyset$
2	$\{e_3\}$	$\emptyset$	$\{e_1\}$	$\emptyset$
3	$\emptyset$	$\{e_2, e_3\}$	$\{e_1, e_3\}$	$\emptyset$
4	$\emptyset$	$\{e_3, e_4\}$	$\{e_1, e_3\}$	$\emptyset$
5	$\{e_1, e_4\}$	$\emptyset$	$\{e_1\}$	$\{e_1\}$
6	$\{e_4\}$	$\emptyset$	$\{e_1, e_4\}$	$\{e_4\}$

## An Example of Available Expressions Analysis

### Notes

