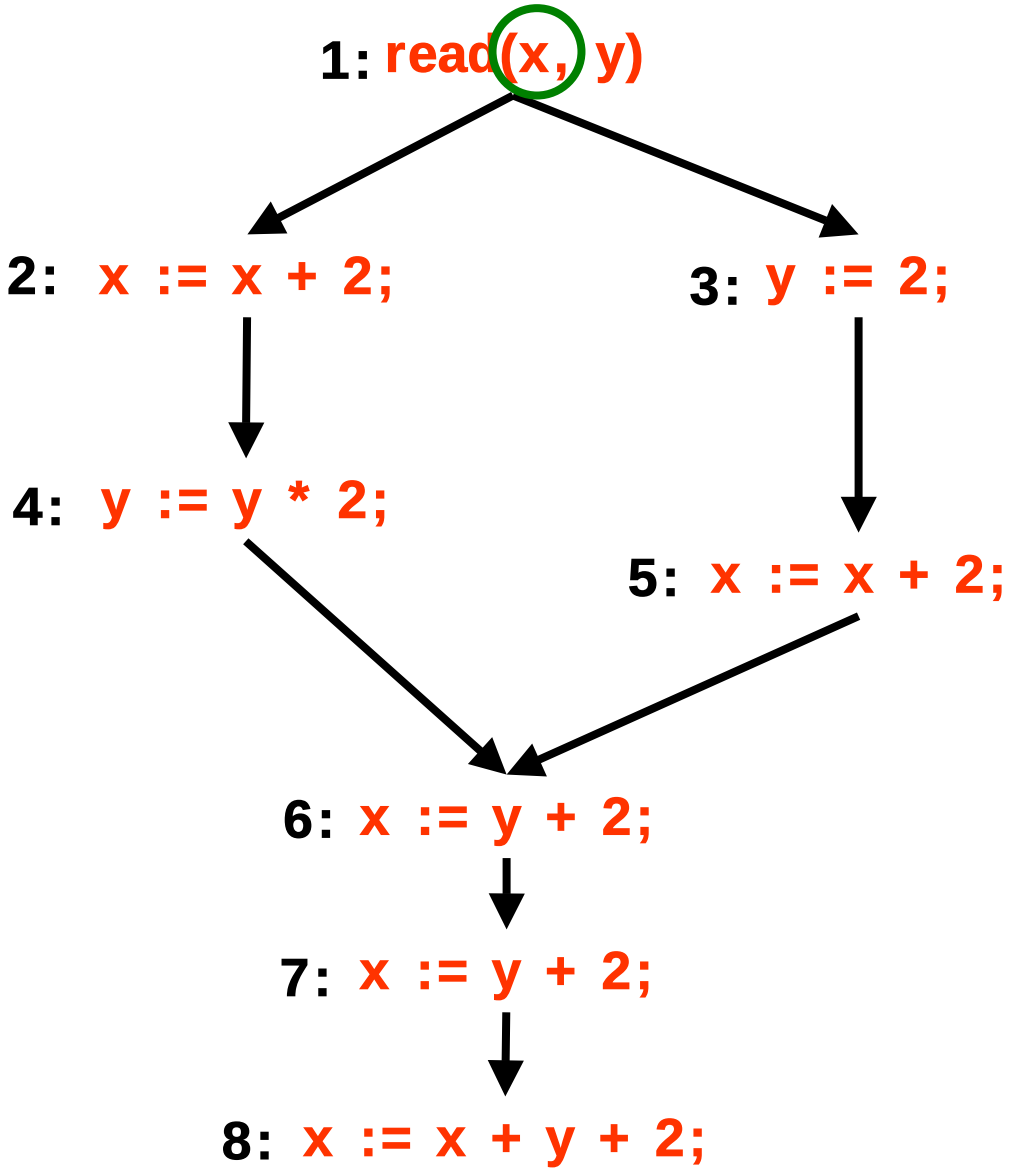
Data-flow Testing

read(x, y)

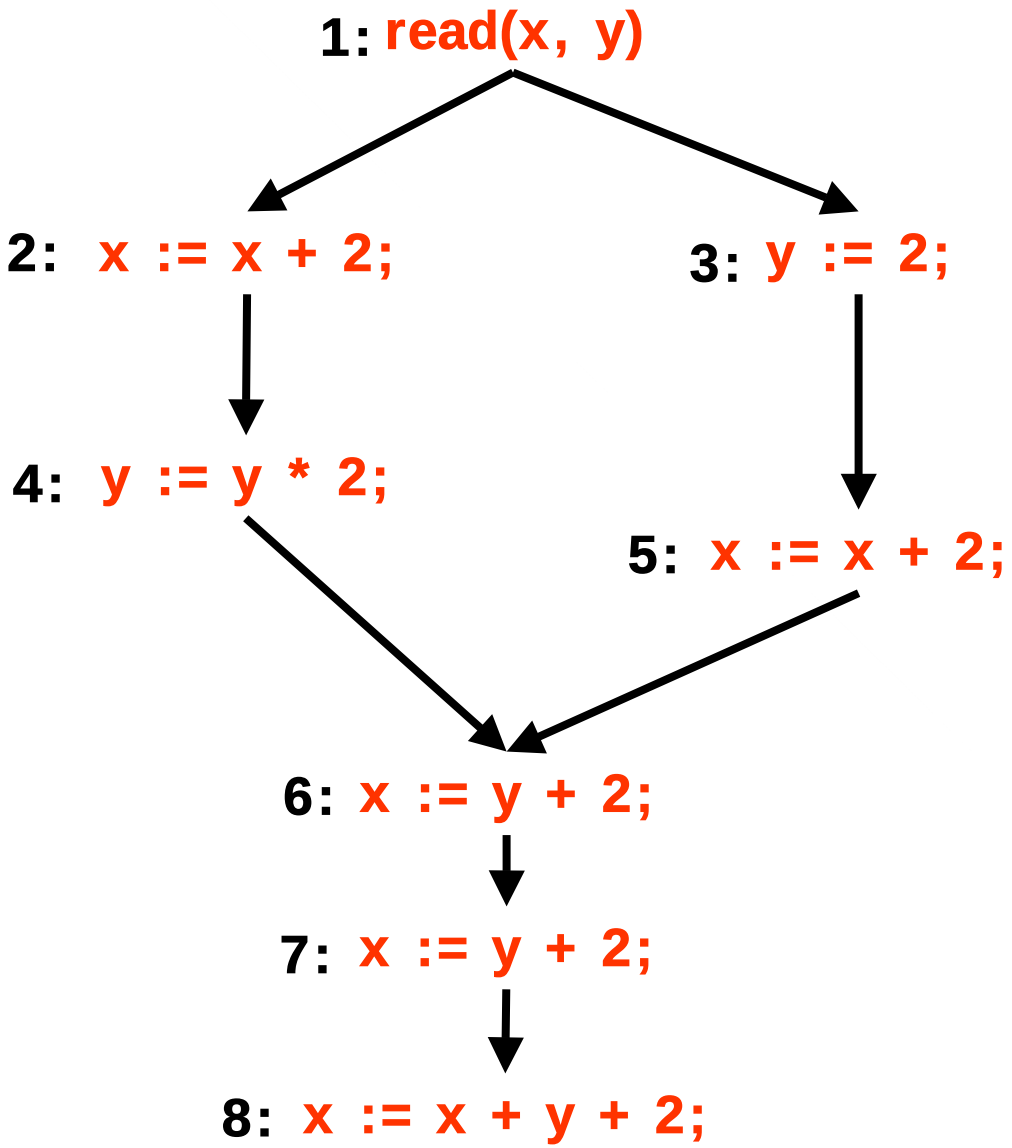
x := x =;

y(x :=;)]T1166.91 j-586857 T

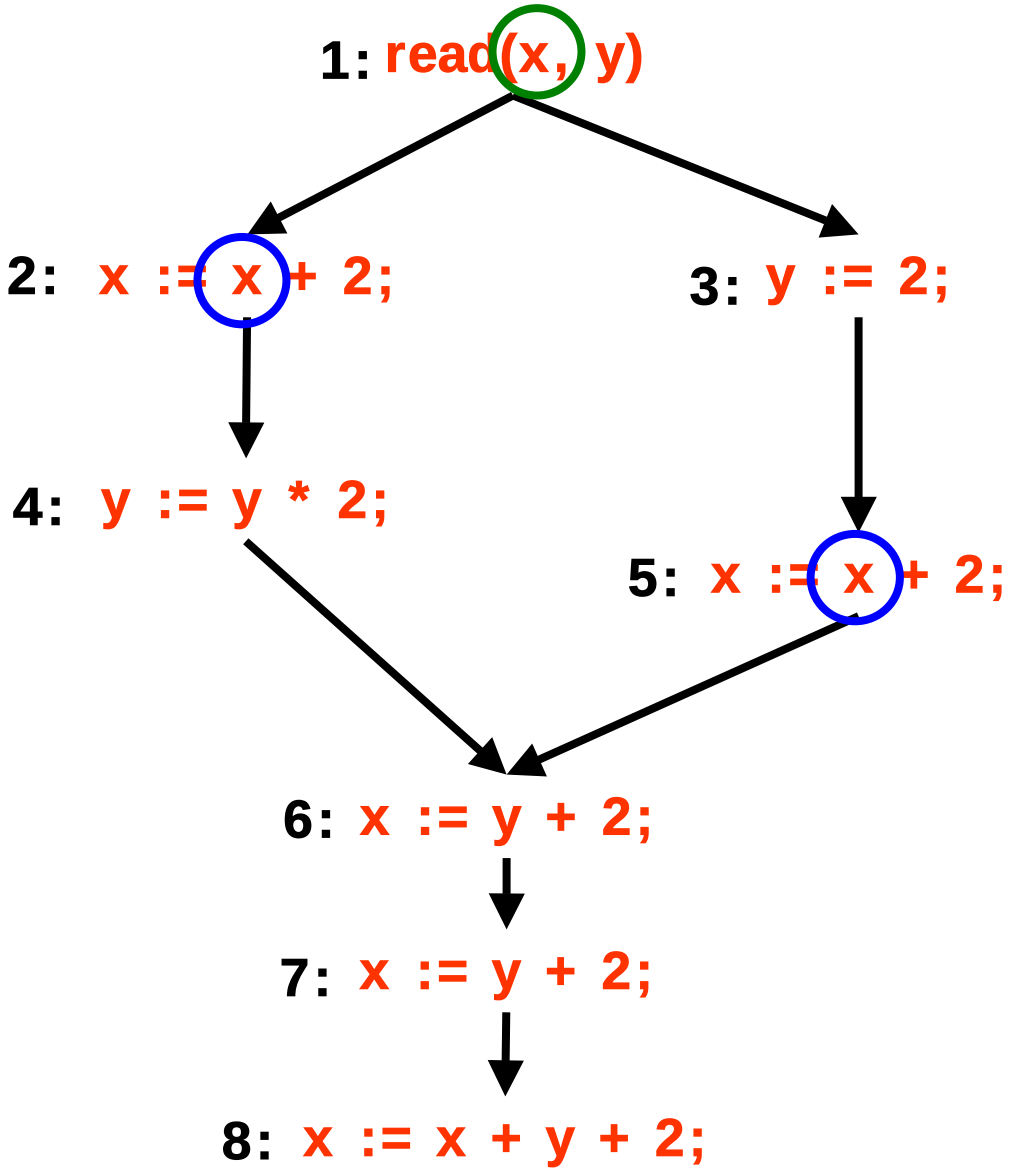
Data-flow Testing



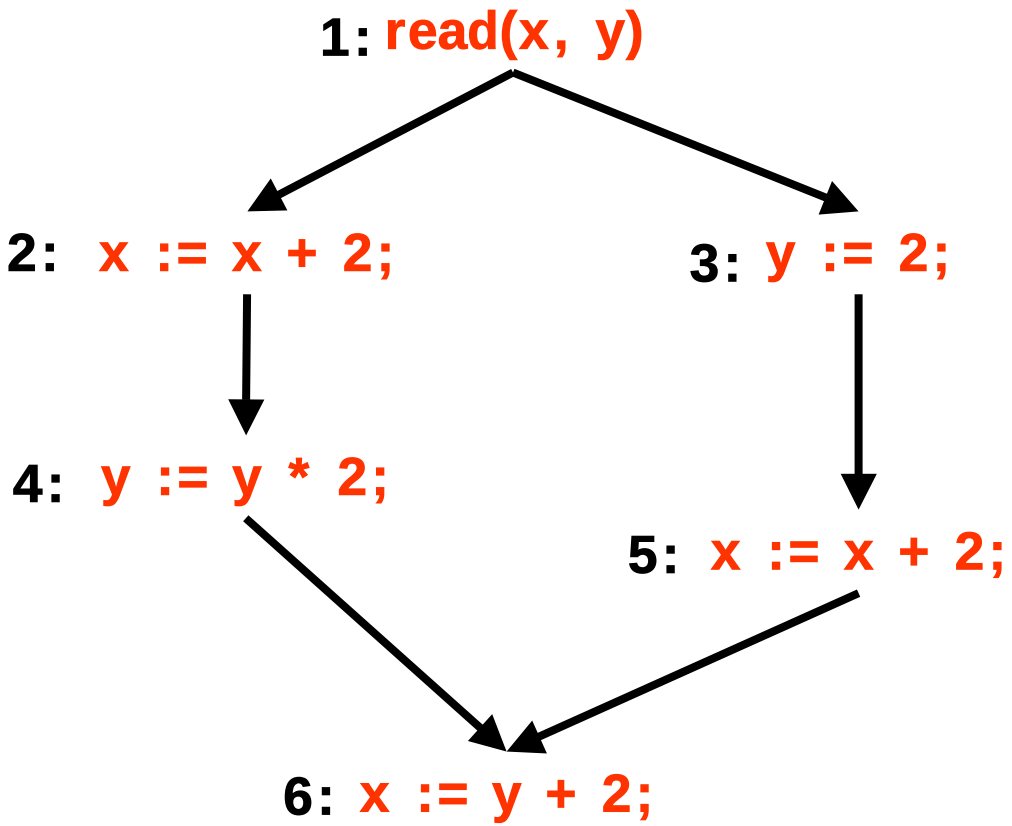
Data-flow Testing



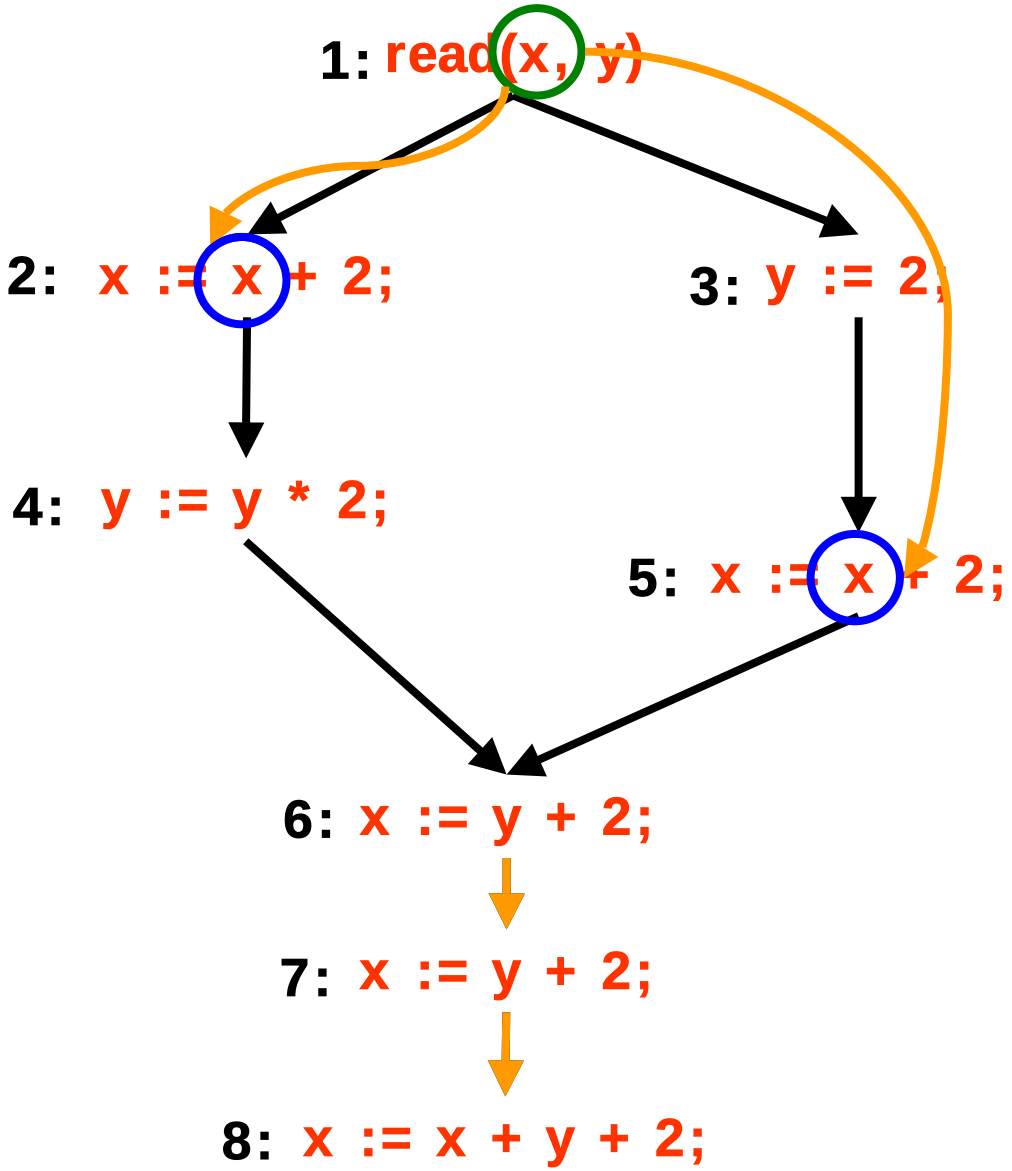
Data-flow Testing



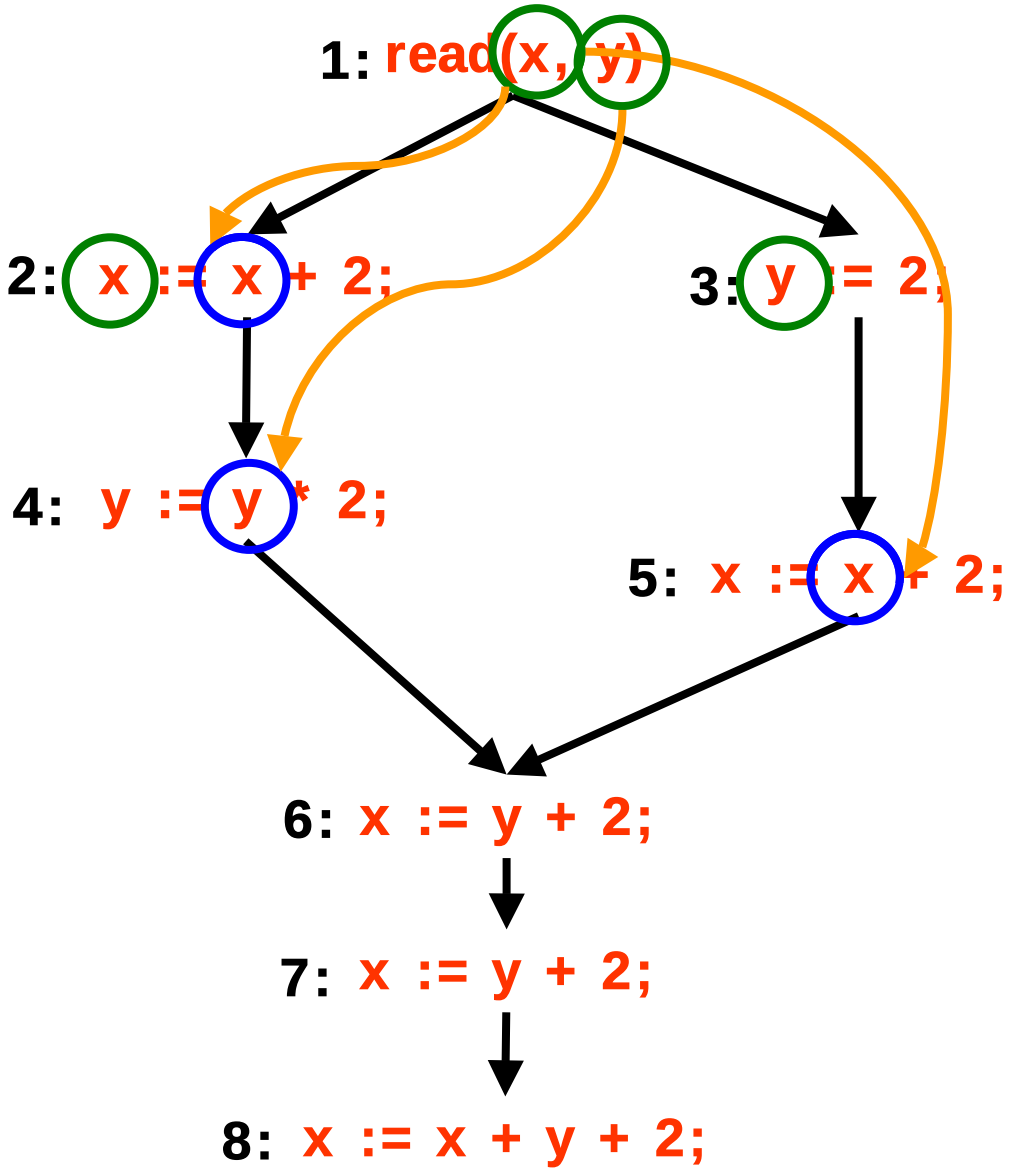
Data-flow Testing



Data-flow Testing



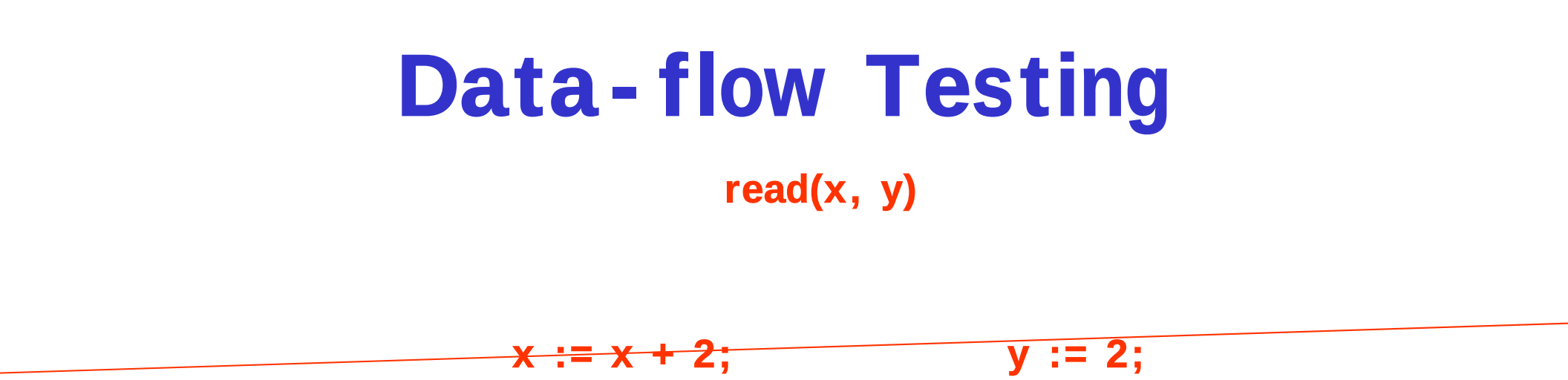
Data-flow Testing



Data-flow Testing

read(x, y)

Data-flow Testing



Data-flow Testing

read(x, y)

Data-flow Testing

read(x, y)

x := x + 2;

yTw[]T1146691 Tj5868557 T

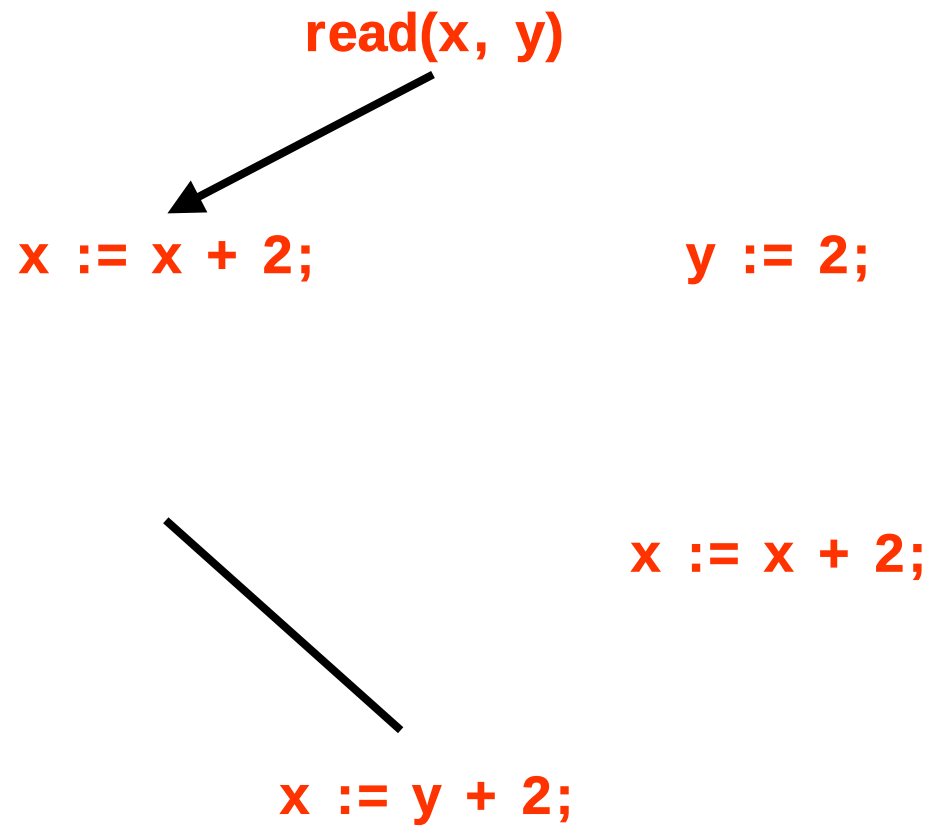
Data-flow Testing

read(x, y)

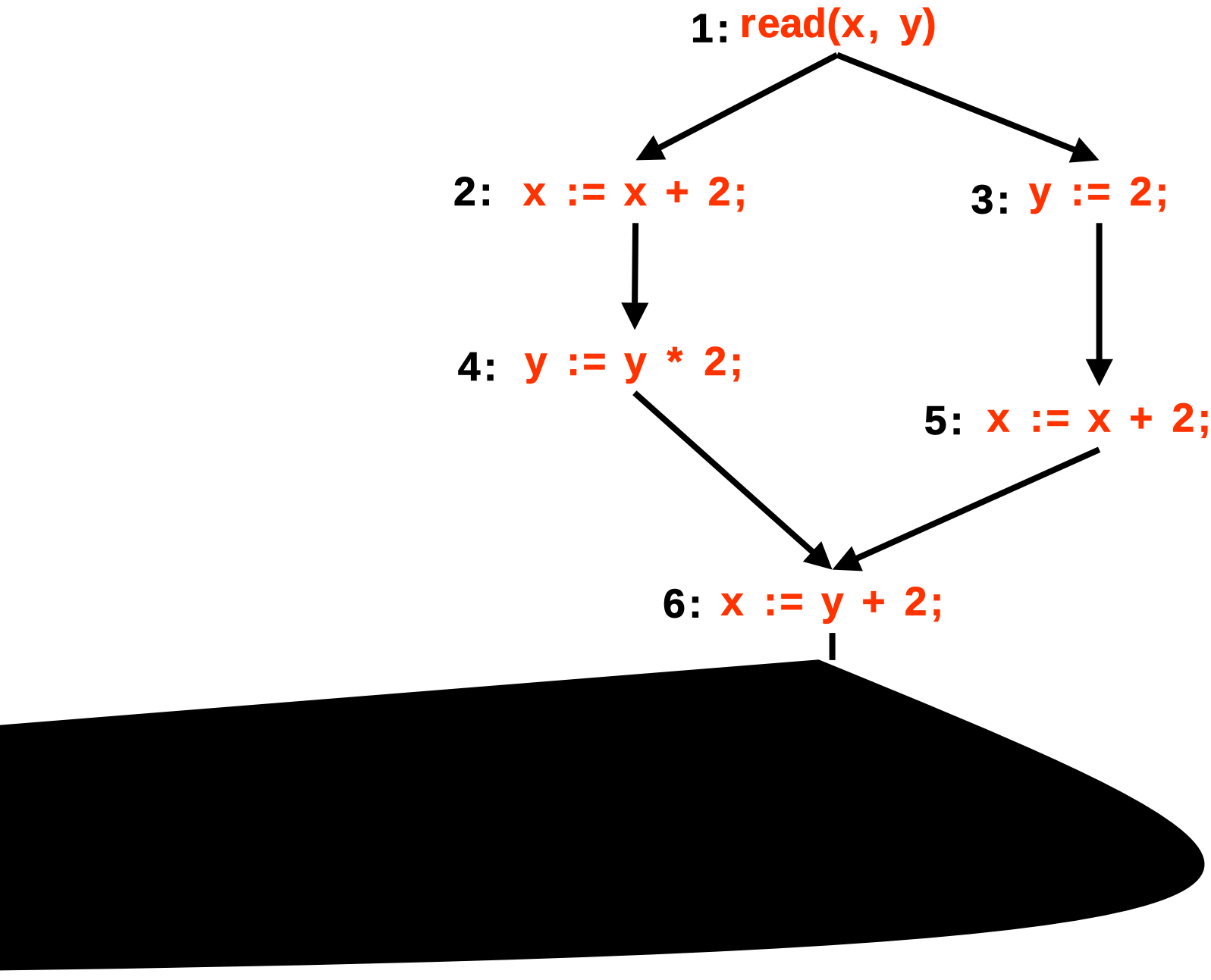
x := x + 2;

yTw[]T1146691 Tj5868557 T

Data-flow Testing



Data-flow Testing



Data-flow Testing

read(x, y)

Data-flow Testing

read(x, y)

x := x + 2;

y := 2;

All Definitions Criterion

- A set P of execution

All Definitions Criterion

- **A set P of execution paths satisfies the all-definitions criterion iff**
  - **for all definition occurrences of a variable x such that**
    - there is a use of x, which [illegible] reachble x from m-n) at n) [illegible] definition m-n
    waths p inP ossuh shat

All Definitions Criterion

- **A set P of execution paths satisfies the all-definitions criterion iff**
  - **for all definition occurrences of a variable x such that**
    - there is a use of x,

All Uses Criterion

read(x, y, z)

x := x + 2;

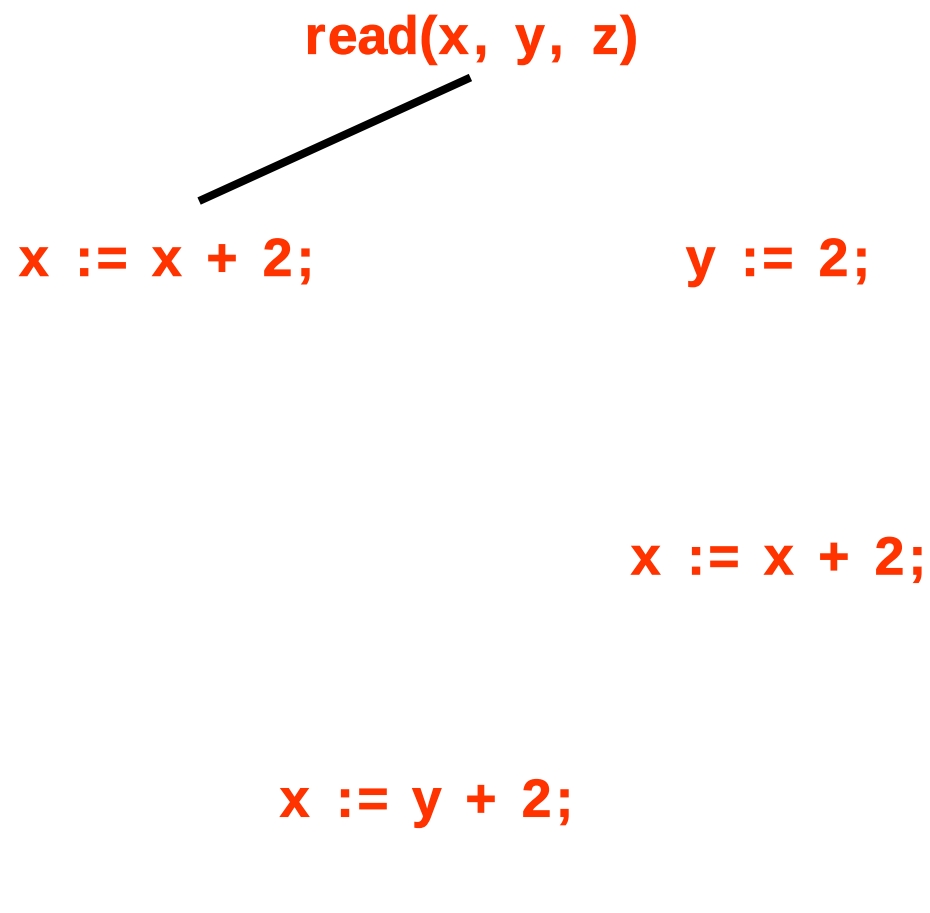
y := 2;

x := x + 2;

x := y + 2;

All Uses Criterion

All Uses Criterion



All Uses Criterion

read(x, y, z)

x := x + 2;

y := 2;

x := x + 2;

x := y + 2;x 21365.2057.85BTCriterion

All Uses Criterion

read(x, y, z)

y := 2;

2;

All DU-paths criterion

- **A set P of execution paths satisfies the all-DU paths criterion iff**
  - **for all definitions of a variable x and all paths q through which that definition reaches a use of x,**
  - **there is at least one path p in P such that**
    - q is a subpath of p and q is cycle-free

An Applicable Family of Data Flow Testing Criteria

- **Assumptions about the program**
  - **No**
    - goto statements
    - with
    - variant records
    - Functions having ‘var’ parameters
      - By reference
    - Procedural or functional parameters
    - Conformant arrays
  - **Every boolean expression that determines the flow of control has at least one occurrence of a variable or a call to the function ‘eof’ or ‘eoln’**

Program Structure

- Program consists of ‘blocks’
-

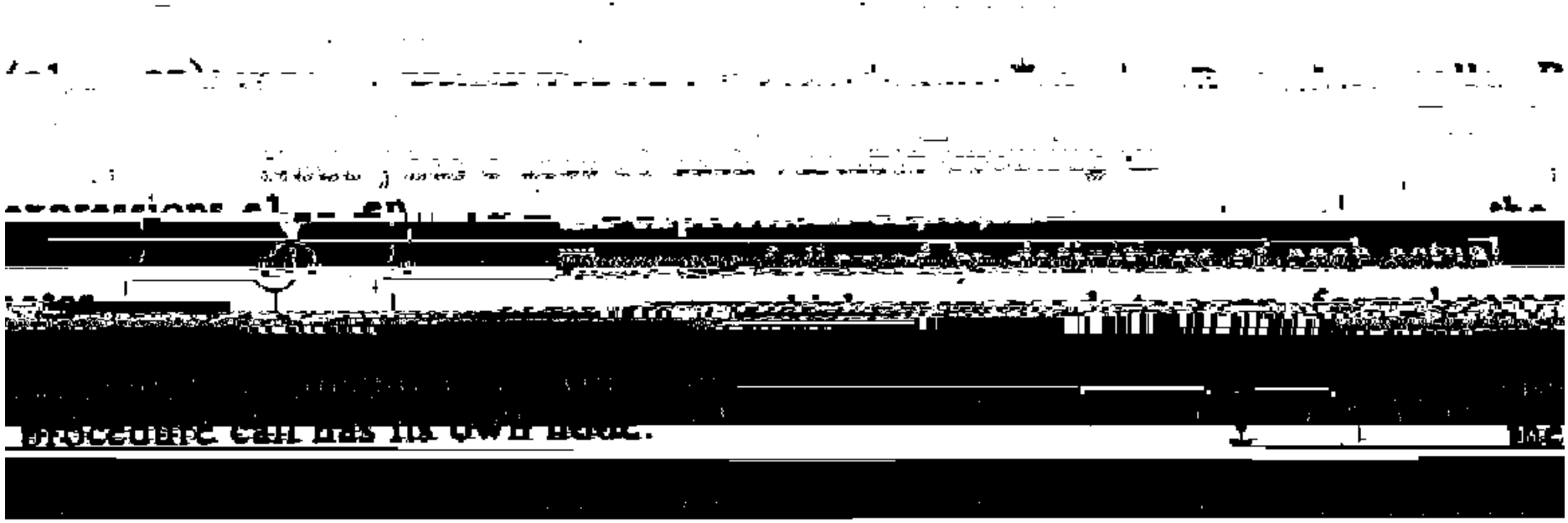
Classifying each variable occurrence

- Definition
  - Value is stored in a memory location
- Use
  - Value is fetched from a memory location
- Undefinition
  - Value and location becomes unbound
- C-use
  - Use in a computation or output statement
  - Associated with each node
- P-use
  - Use in a predicate
  - Associated with each edge

Simple Statements

Simple Statements

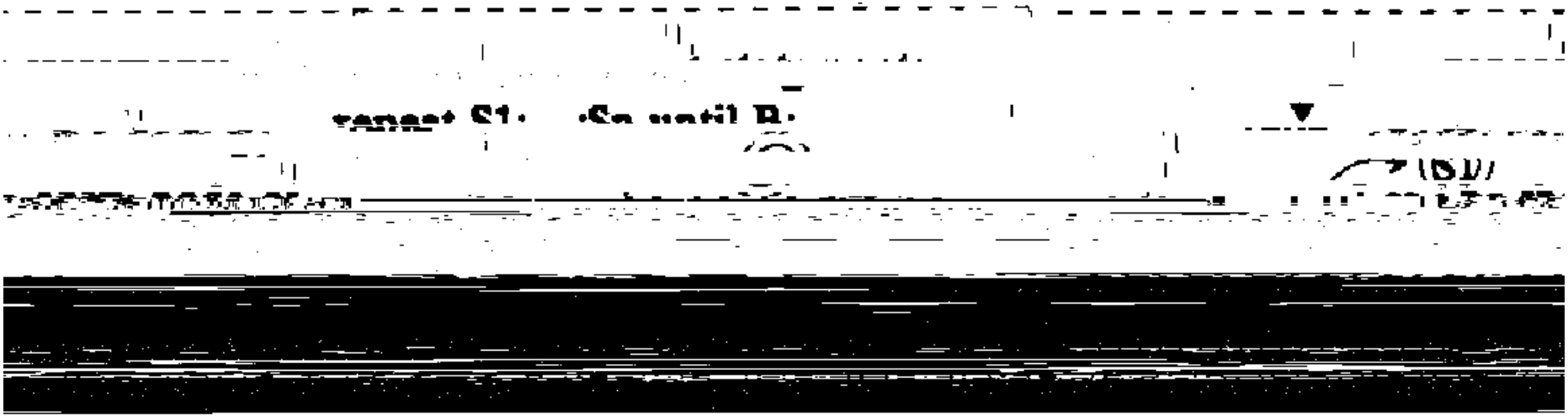
Simple Statements



Repetitive Statements

Repetitive Statements

Repetitive Statements



Conditional Statements

Conditional Statements

Arrays

- **Arr variableysArrays**

Pointers

- Impossible to determine statically the

Records & Files

Restricted Programs Class

- **Satisfying the following properties**
  - **NSUP**
    - No-syntactic-undefined-p-use Property
      - For every p-use of a variable x on aunedge (i,j), iun P, there is some path from the start node to edge (i,j), which contains a global definition of x
  - **NSL**
    - Non-straight-line property
      - P has at least one conditional or repetitiven statement
        - » At least one node iunP's flow-graph has moren thaunone successor
        - » At least one variable has a p-use in P

Def-use graph

- Obtained from the flow graph
- Associate with each node the sets
  - C-use(i)

Definitions for def-use graph

All-DU-paths criterion

- If variable x has a global definition i[ode i, the a(All-DU-path)]TJ*rg0 Tc0.09

Other DF testing criteria

- All-p-uses
- All-c-uses
- All-p-uses/some-c-uses
- All-c-uses/some-p-uses

Definitions of DF criteria

“includes”

Includes relationship

Applicability

- It may be the case that no test set for

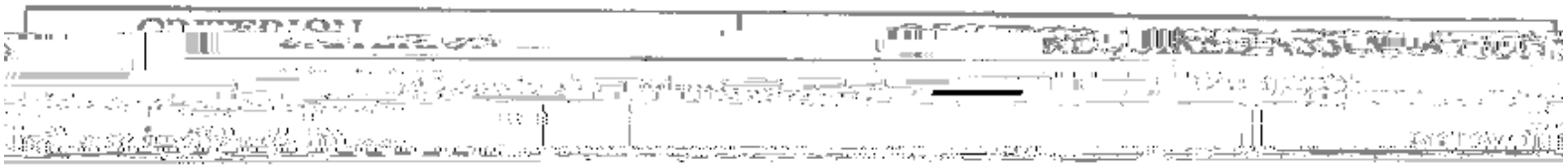
Recall Definition

-

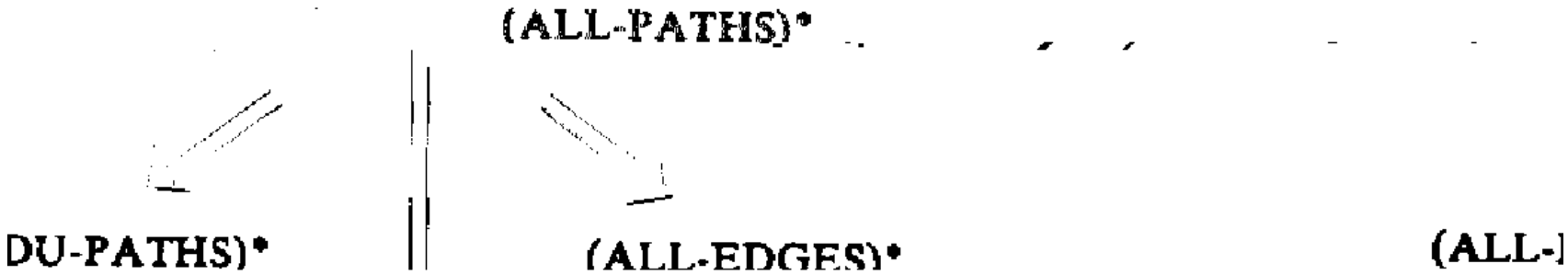
Equivalently

- **fdcu(x,i) =**
  - **{j ∈ dcu(x,i) | the association (i,j,k) is executable}**
- **fdpu(x,i) =**
  - **{(j,k) ∈ dpu(x,i) | the association (i,(j,k),x) is executable}**
- **Intuitively**
  - new criterion C* for each DF criterion C
  - By selecting the required associations from fdcu(x,i) and fdpu(x,i) instead of from dcu(x,i) and dpu(x,i)

Feasible Data-flow Criteria (FDF)



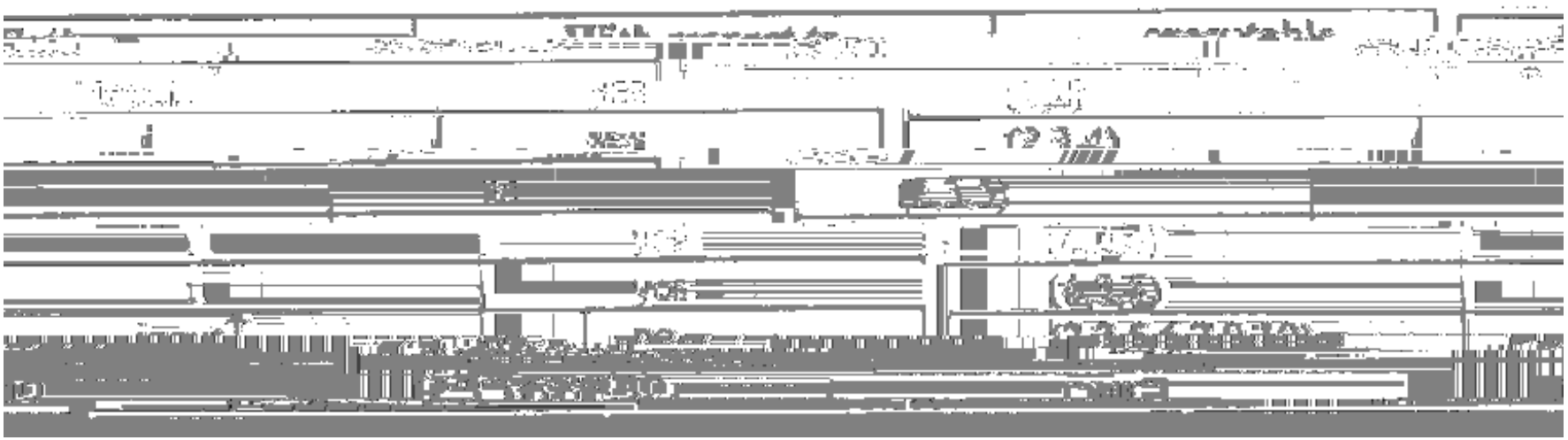
Includes Relationships



Why the different relationships

Example

The Program's DU-paths



Why the different relationships

Let x = X (any integer)
And y = Y < 0

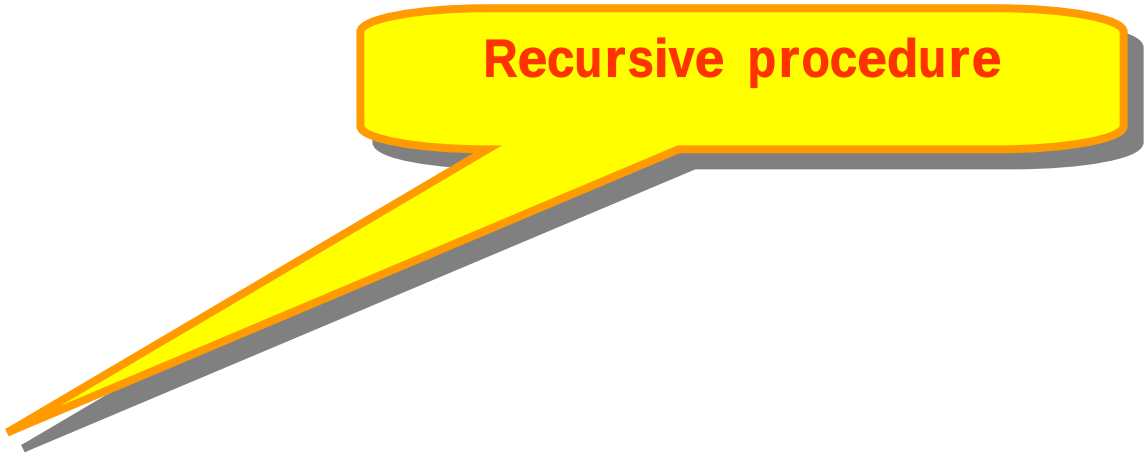
Path executed is
{1,2,3,4,3,4,
3,5,6,7,9,10}

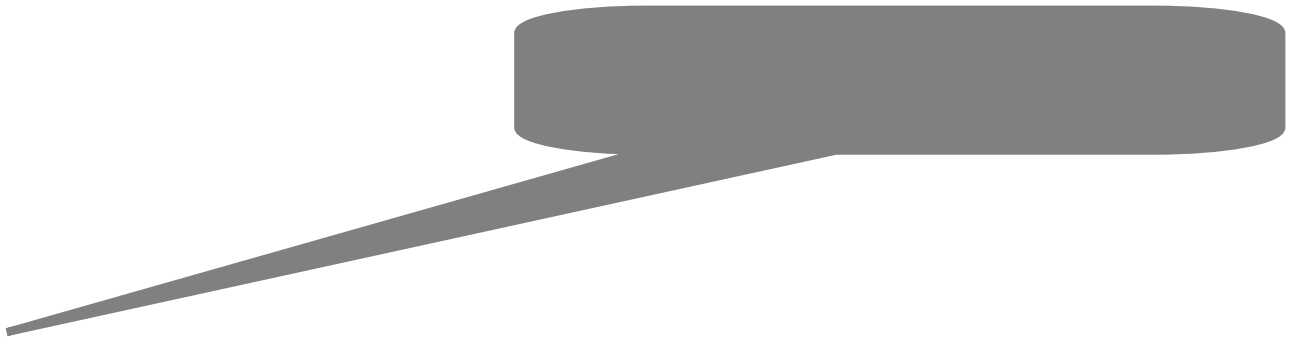
Are all DU-paths
shown earlier covered?
YES

But the associations (2, (6,8), y) and (2,8,x)
are not!
And they are executable by a test case that causes the execution of {1,2,3,4,3,4,3,5,6,8,9,10}
Hence (all-du-paths)*

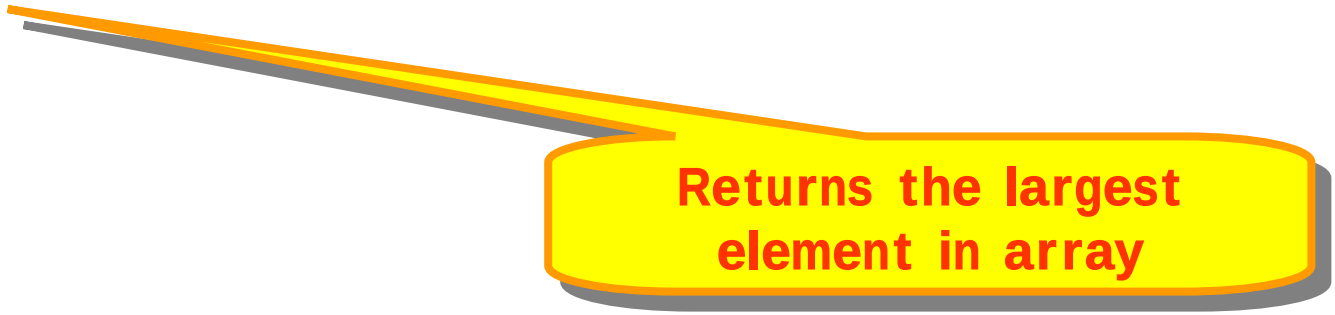
Interprocedural DF Testing

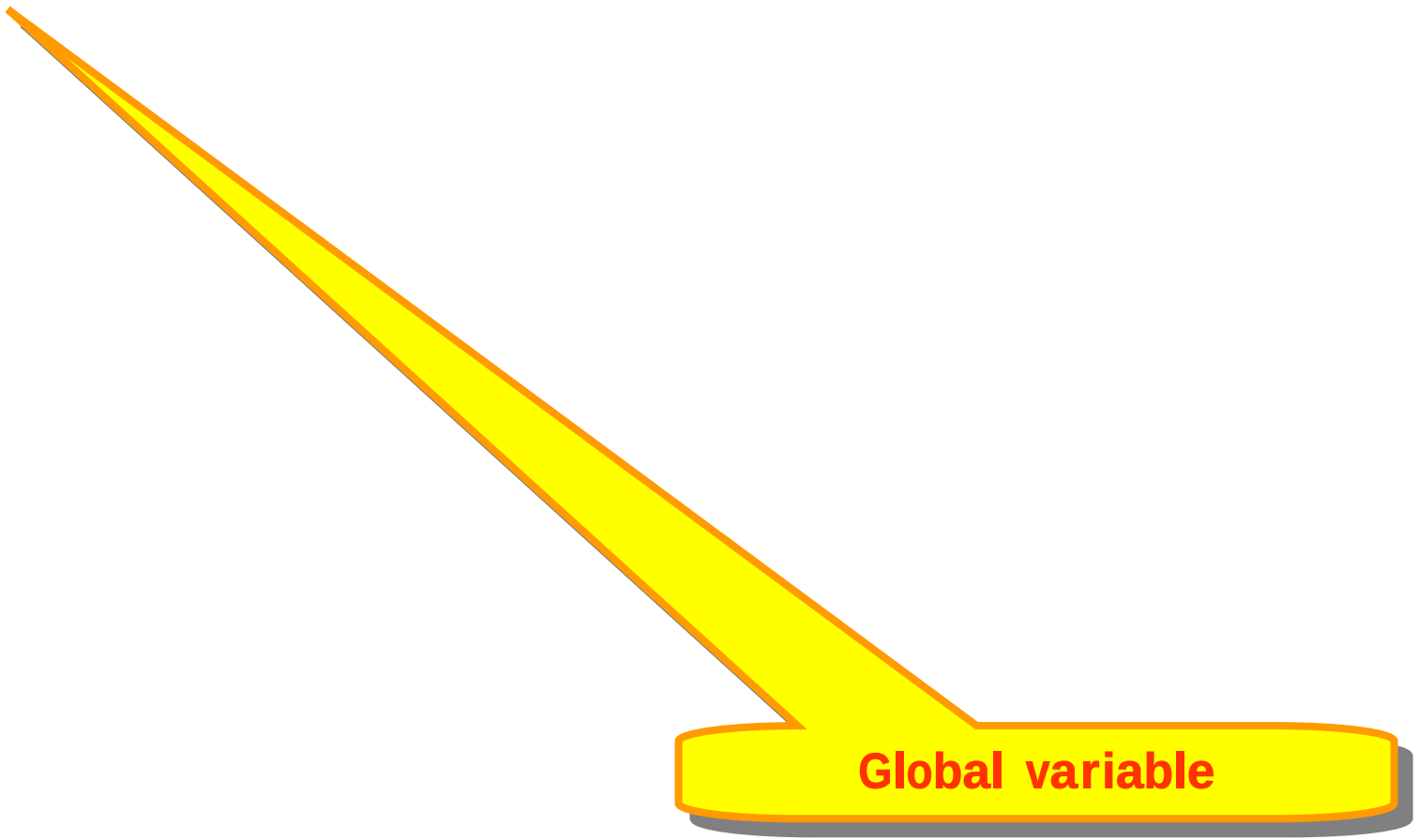
- Most DF testing methodologies deal with dependencies that exist within a procedure (i.e., *intra*procedural)
- Data dependencies also exist among procedures
- Requires analysis of the flow of data across procedure boundaries
- Calls and Returns
- Direct dependencies (single call/return)
- Indirect dependencies (multiple calls/returns)


Recursive procedure

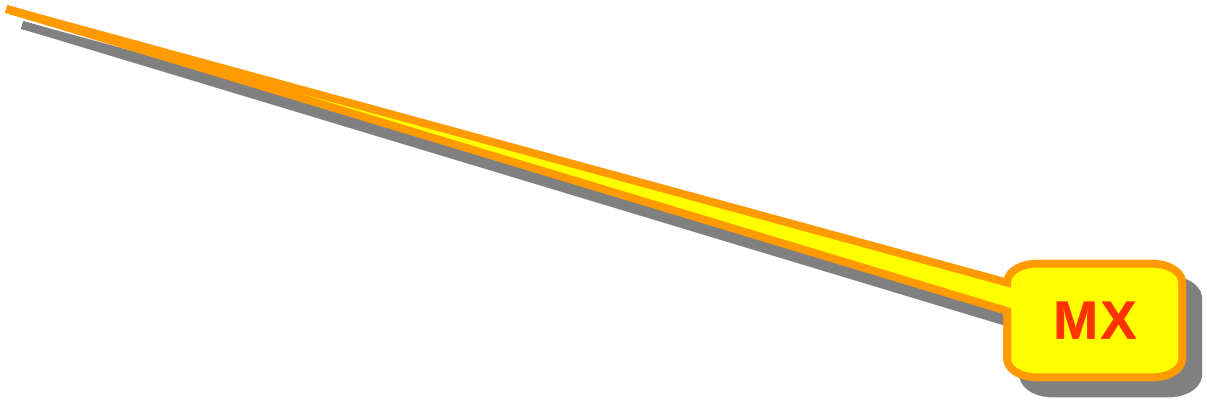


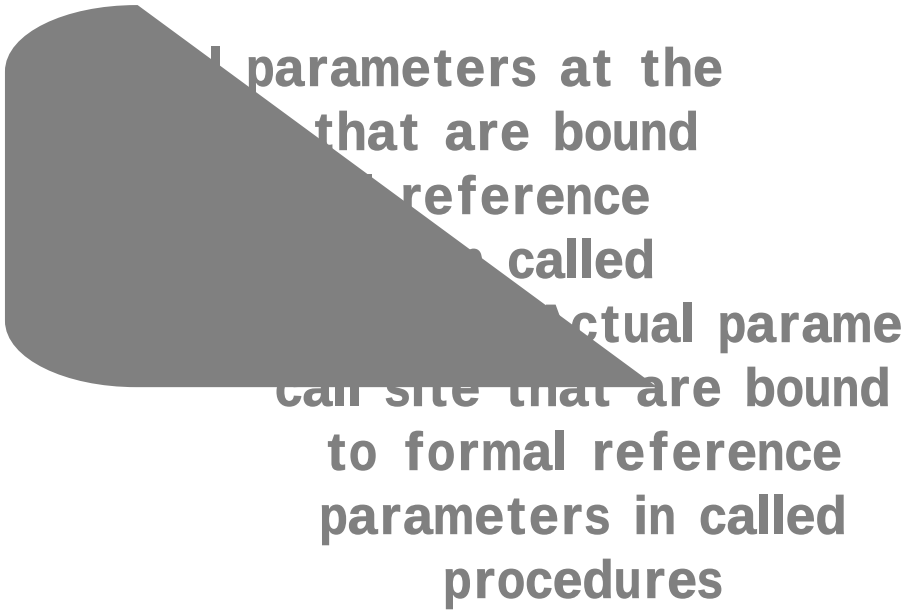

last element of array

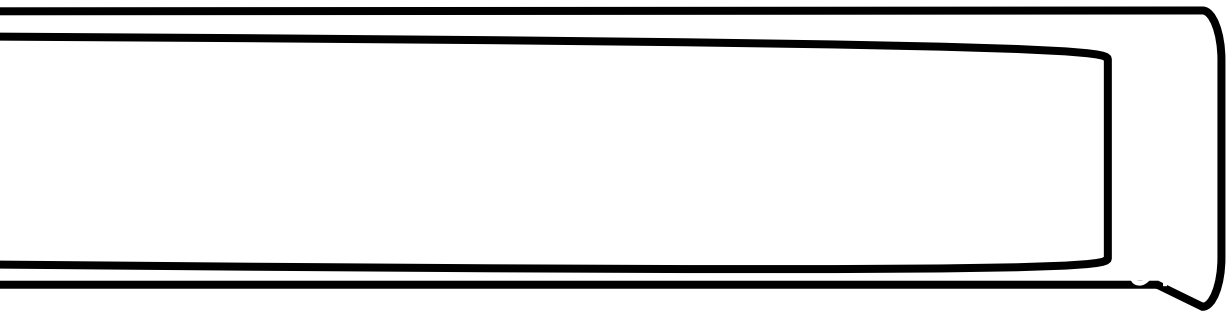

Returns the largest
element in array


Global variable

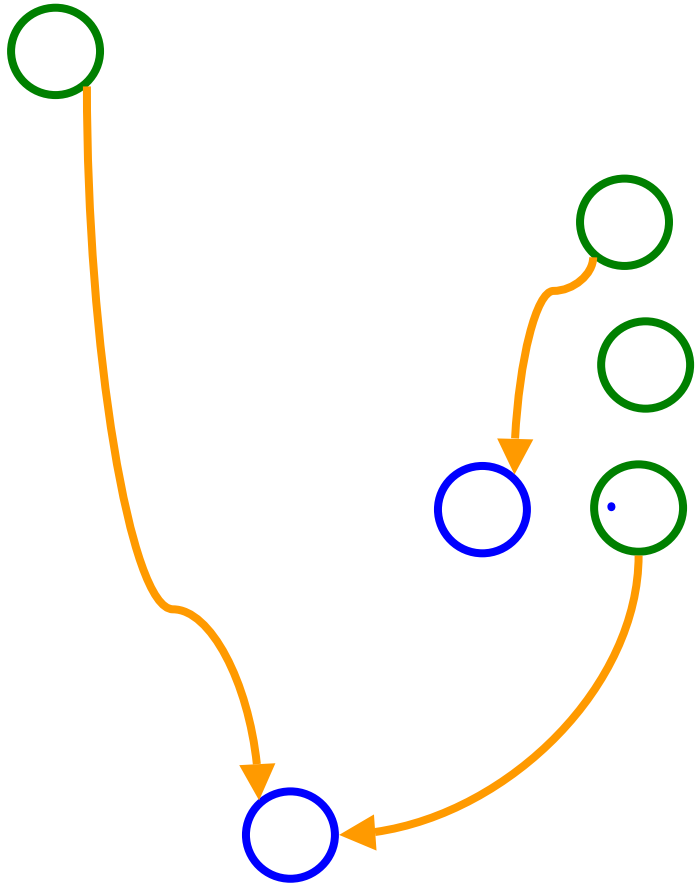
Lets consider only reference parameters that reach across procedure boundaries


MX

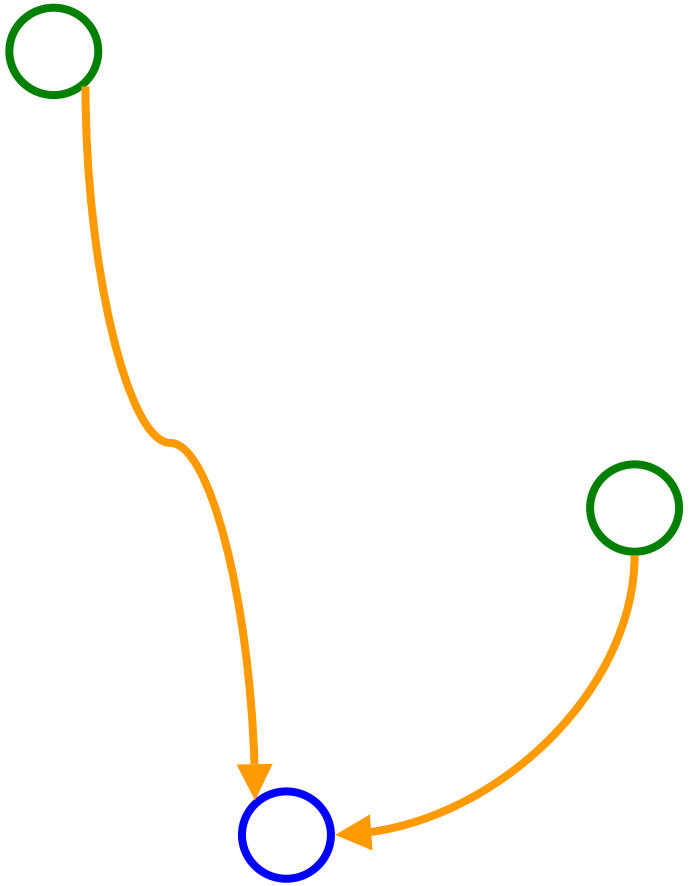

parameters at the
that are bound
reference
called
ctual parame
call site that are bound
to formal reference
parameters in called
procedures



The Def-uses



A test case



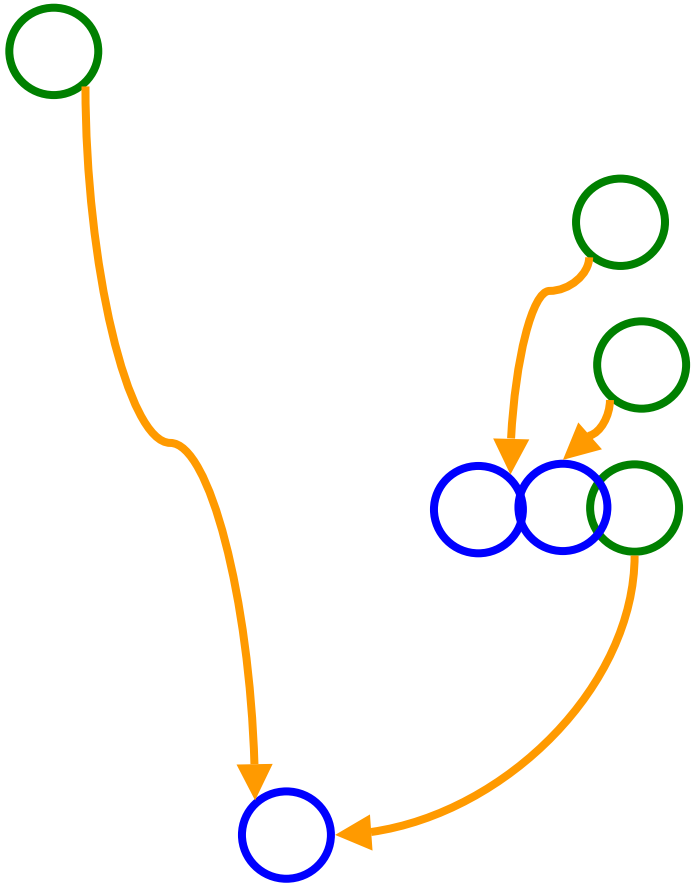
A test case

S = {3,5,1,6}

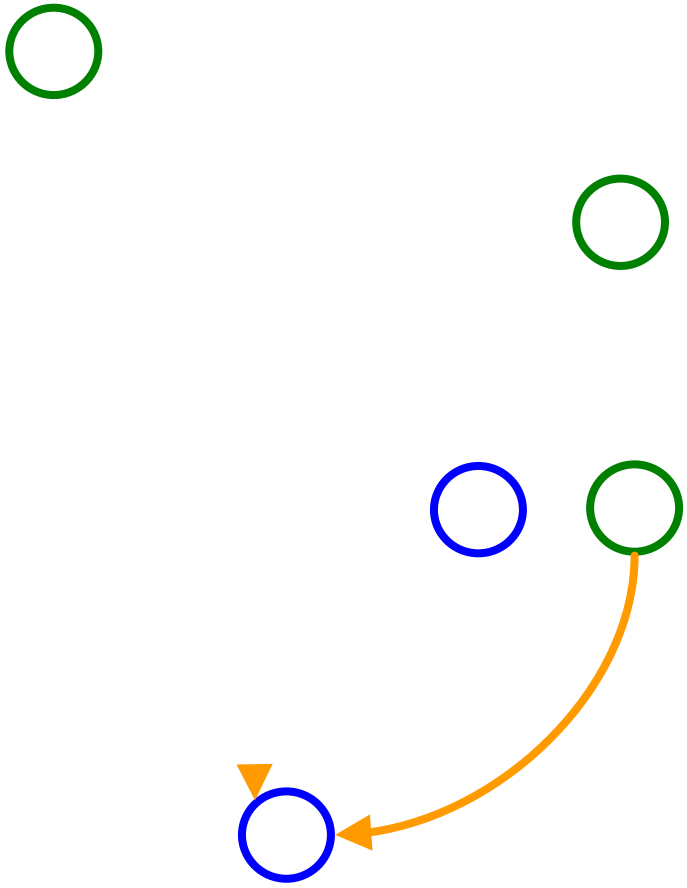
F = 1

L = 4

Execute and check



A test case



Any missed

Any missed def-uses?