

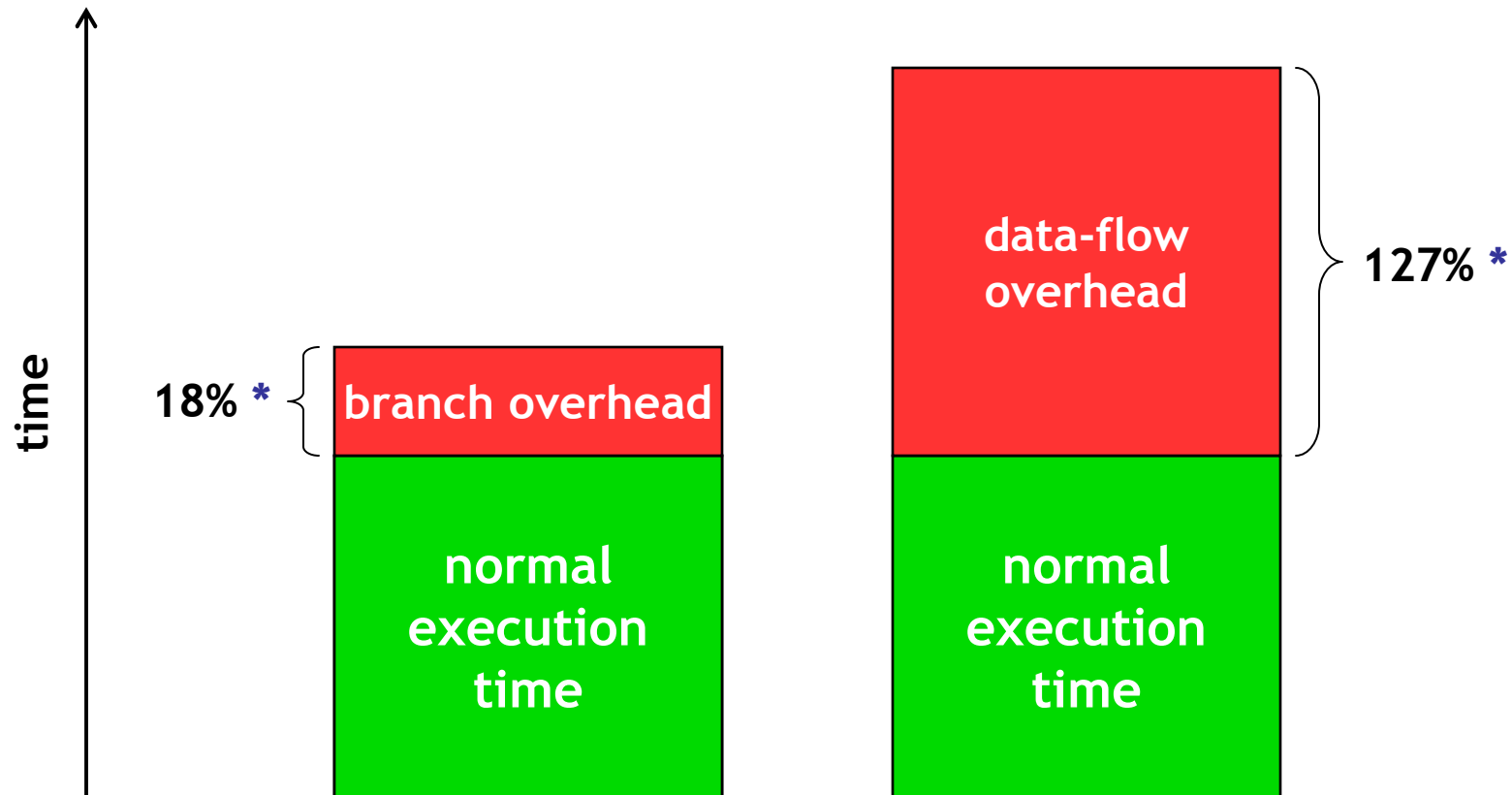
Efficiently Monitoring Data-Flow Test Coverage*

Raul Santelices

Mary Jean Harrold

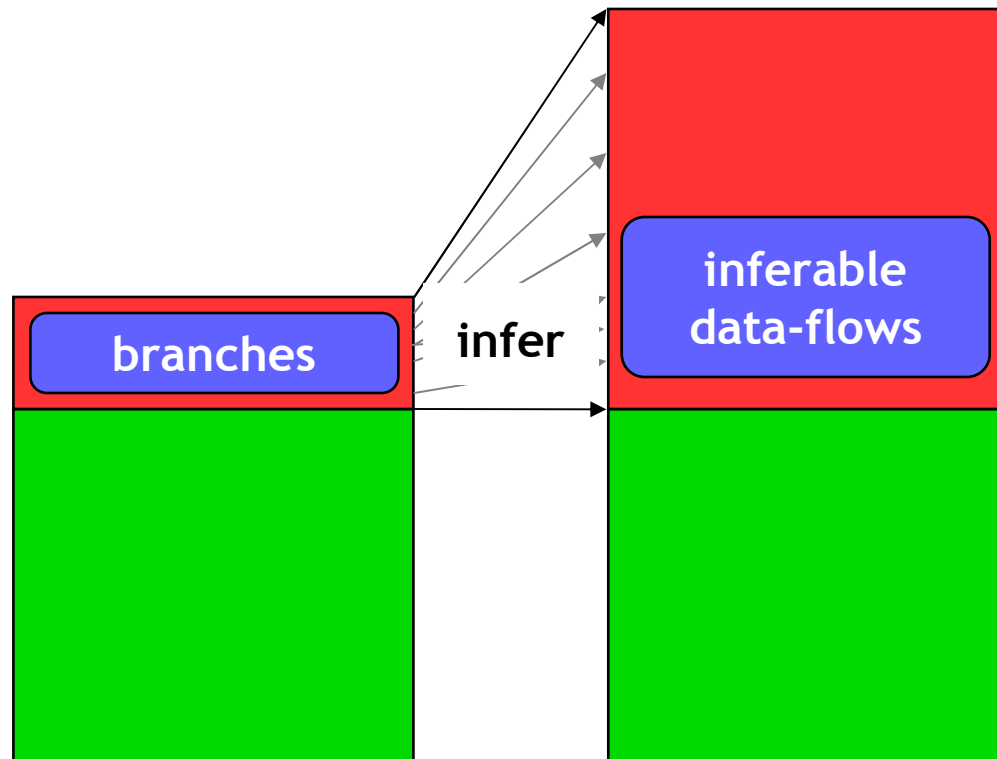
* Supported in part by NSF awards CCF-0541049, CCF-0429117, and CCF-0306372 to Georgia Tech and by Tata Consultancy Services, Ltd.

Problem: Cost of Monitoring



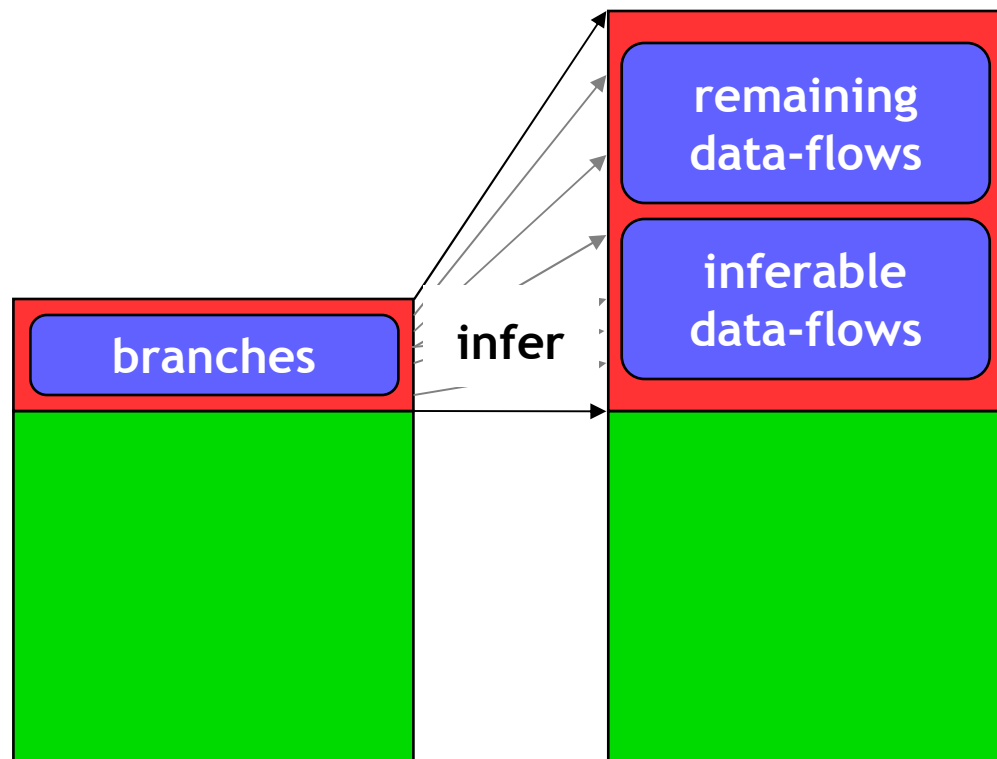
* Averages reported by Misurda et al. (ICSE 2005)

Approach: Branch-based Data-Flow Monitoring



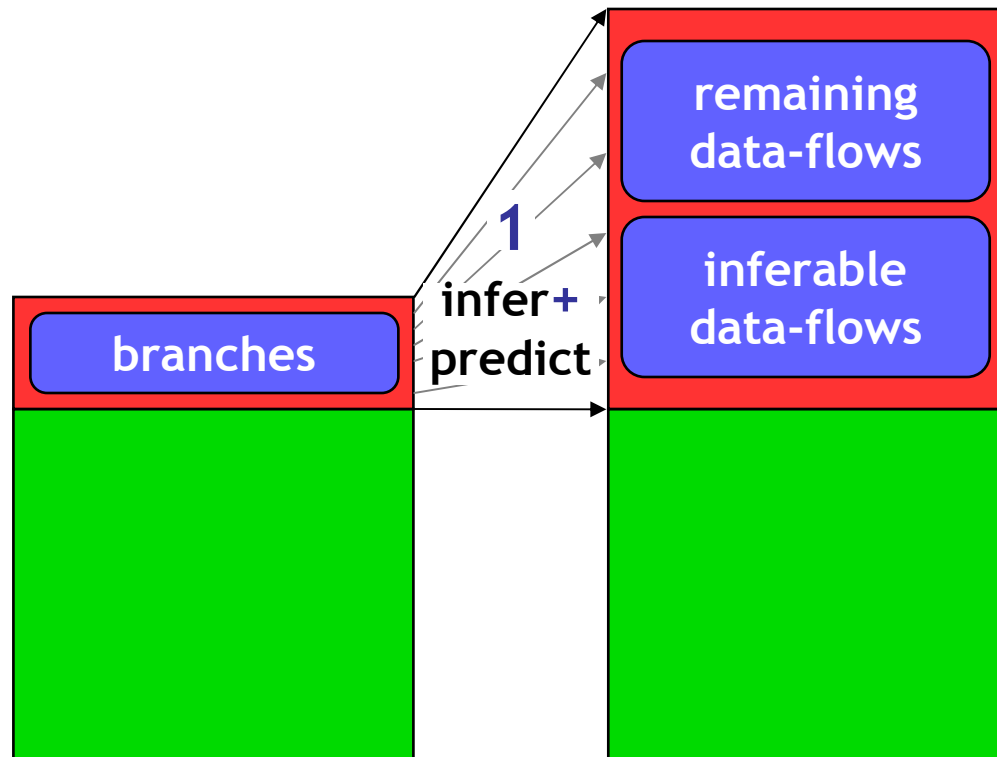
- Infer data-flow coverage (as much as possible)

Approach: Branch-based Data-Flow Monitoring



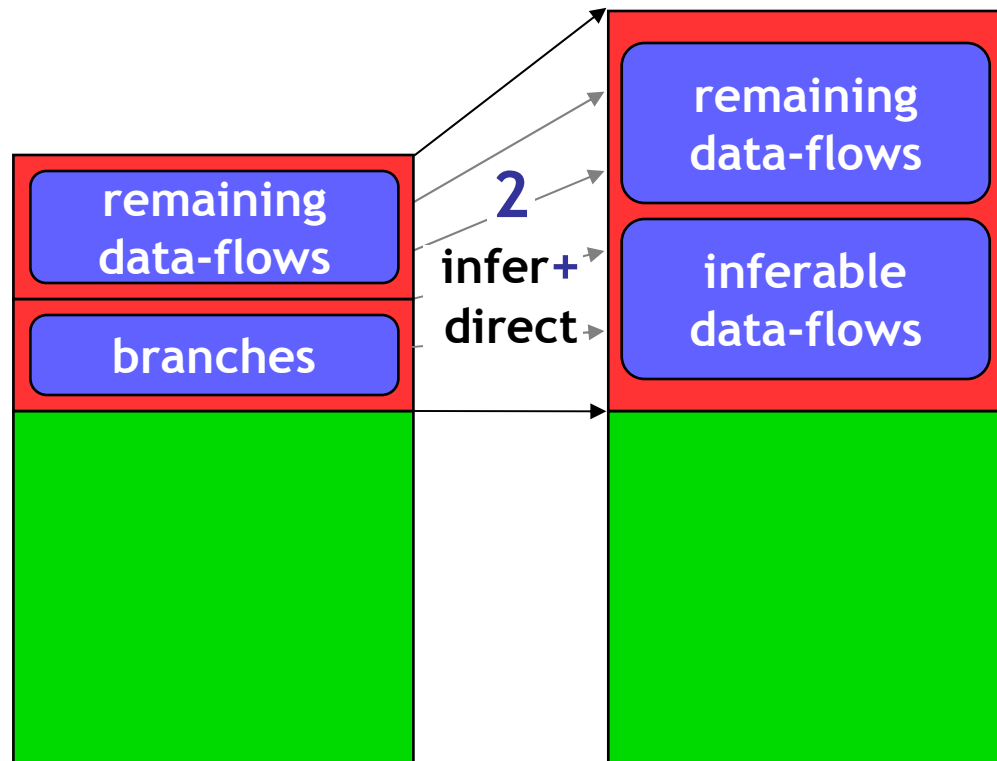
- Infer data-flow coverage (as much as possible)
- Two alternatives for remaining data-flows:

Approach: Branch-based Data-Flow Monitoring



- Infer data-flow coverage (as much as possible)
- Two alternatives for remaining data-flows:
 1. Predict coverage

Approach: Branch-based Data-Flow Monitoring



- Infer data-flow coverage (as much as possible)
- Two alternatives for remaining data-flows:
 1. Predict coverage
 2. Directly monitor coverage (*hybrid monitoring*)

Previous Work

- Same-type entity subsumption
 - Agrawal (POPL '94), Ball and Larus (TOPLAS '94), Bertolino and Marré (TSE '03)
- Different-type entity subsumption
 - Merlo and Antoniol (CASCON '99), Santelices et al. (SOQUA '06)
- Dynamic instrumentation
 - Arnold and Ryder (PLDI '01), Tikir and Hollingsworth (ISSTA '02), Misurda et al. (ICSE '05)

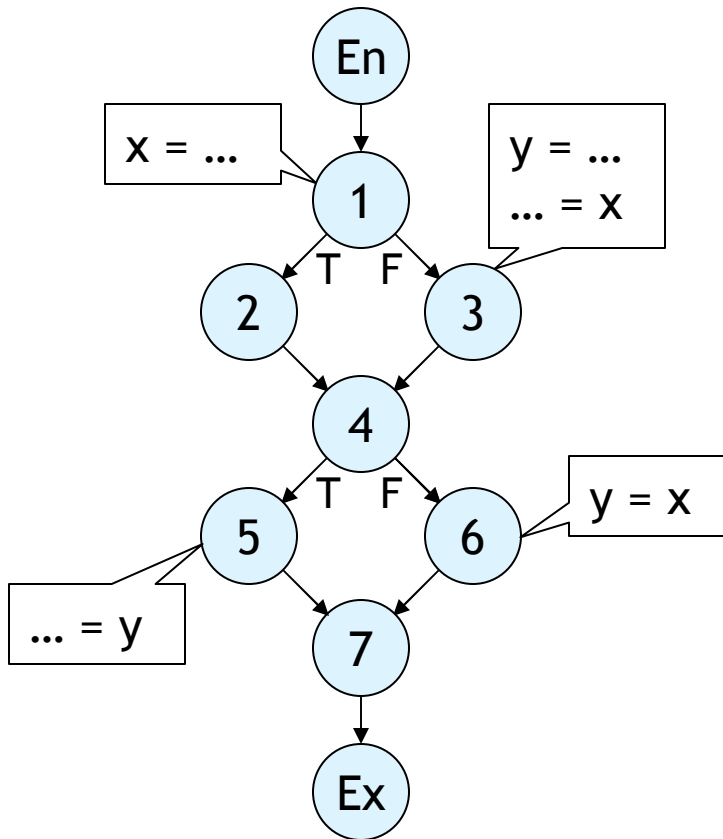
Outline

▶ Background

- Inferability analysis: what
- Inferability analysis: how
- Study
- Conclusion

Program Representation

Control-Flow Graph (CFG)

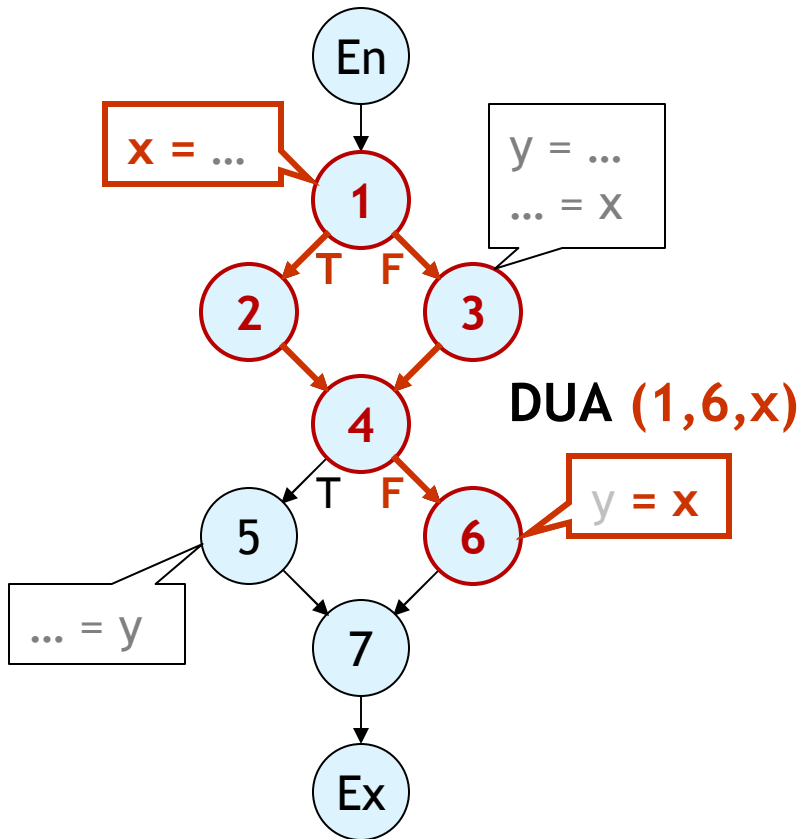


x = ... **definition**

... = x **use**

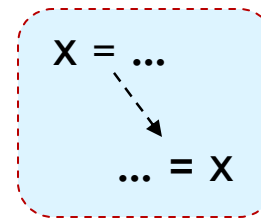
Program Representation

Control-Flow Graph (CFG)



$x = \dots$ definition

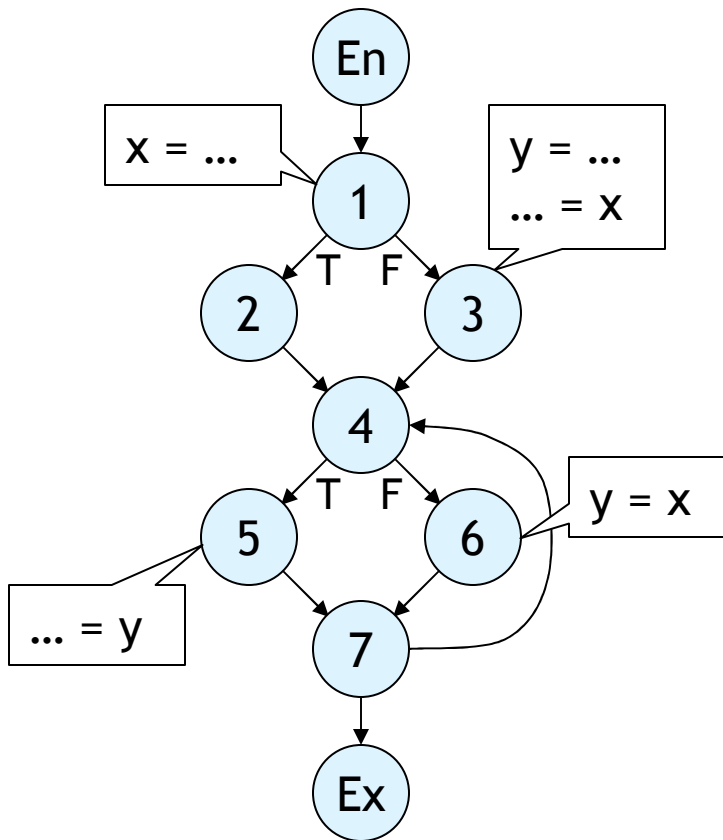
$\dots = x$ use



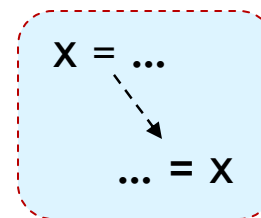
definition-use association (DUA)

data-flow entity

Traditional DUA Monitoring



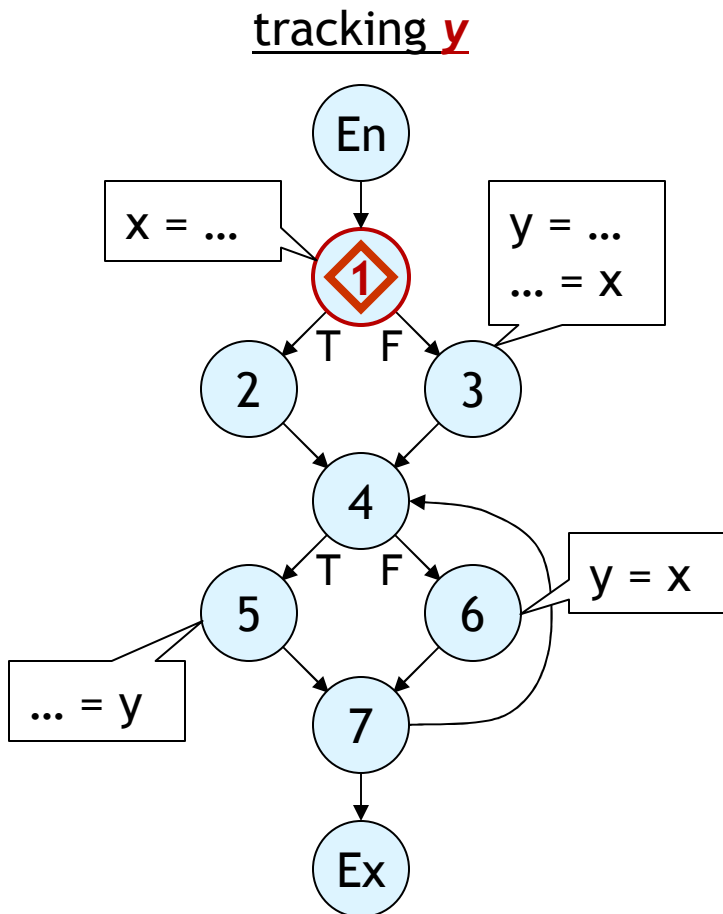
- 1) At definition: record **active definition**
- 2) At use: retrieve **active definition** and mark DUA



definition-use
association (DUA)

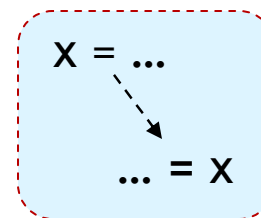
data-flow entity

Traditional DUA Monitoring



- 1) At definition: record **active definition**
- 2) At use: retrieve **active definition** and mark DUA

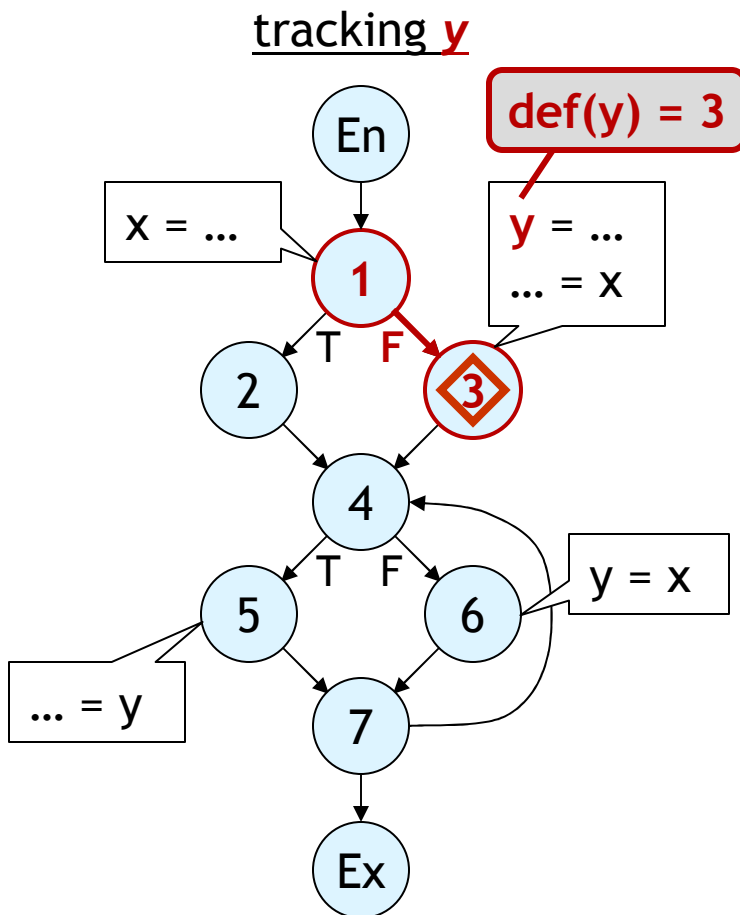
$\text{def}(y) =$



definition-use
association (DUA)

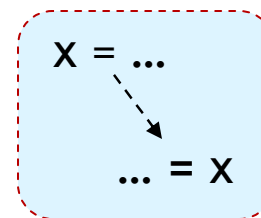
data-flow entity

Traditional DUA Monitoring



- 1) At definition: record **active definition**
- 2) At use: retrieve **active definition** and mark DUA

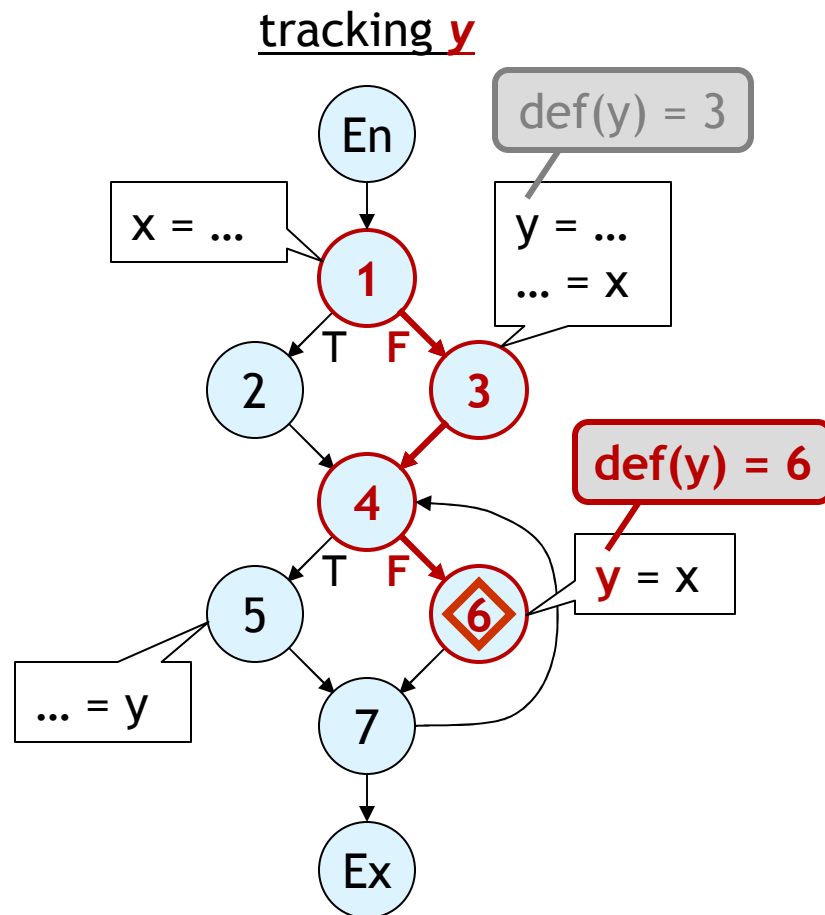
$def(y) = 3$



definition-use
association (DUA)

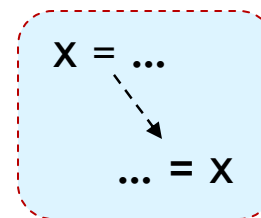
data-flow entity

Traditional DUA Monitoring



- 1) At definition: record **active definition**
- 2) At use: retrieve **active definition** and mark DUA

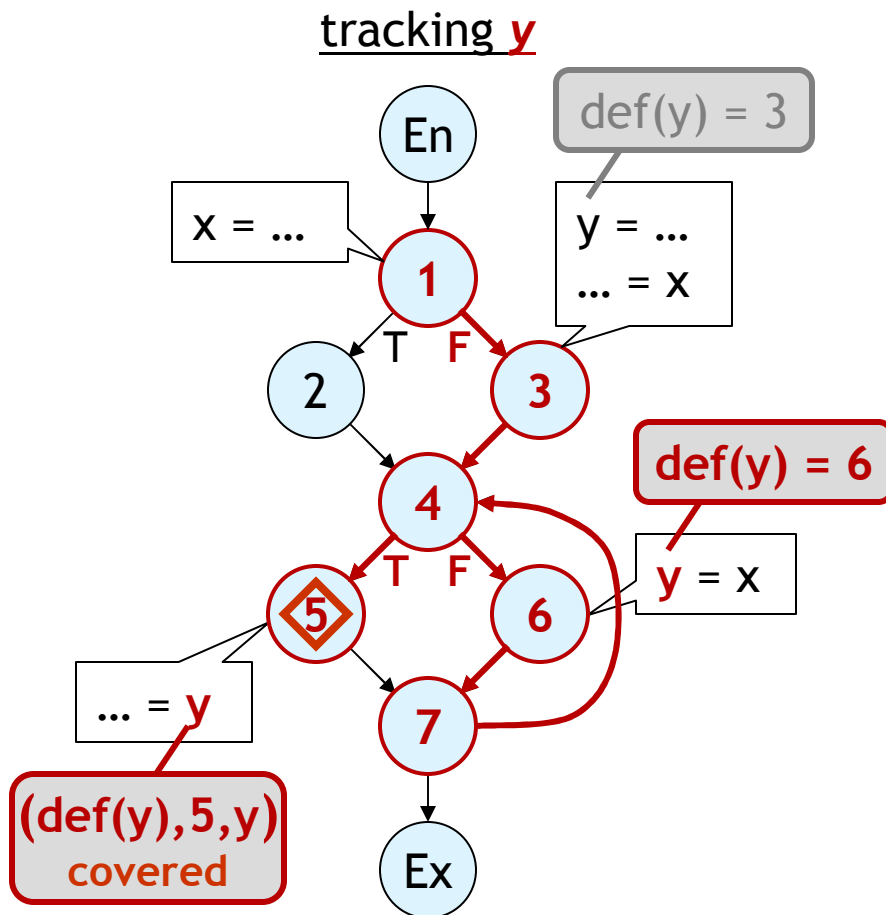
def(y) = ~~3~~ 6



definition-use association (DUA)

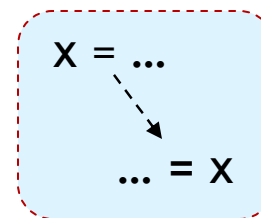
data-flow entity

Traditional DUA Monitoring



- 1) At definition: record **active definition**
- 2) At use: retrieve **active definition** and mark DUA

$\text{def}(y) = \cancel{3} 6$
 $(6, 5, y)$ covered!



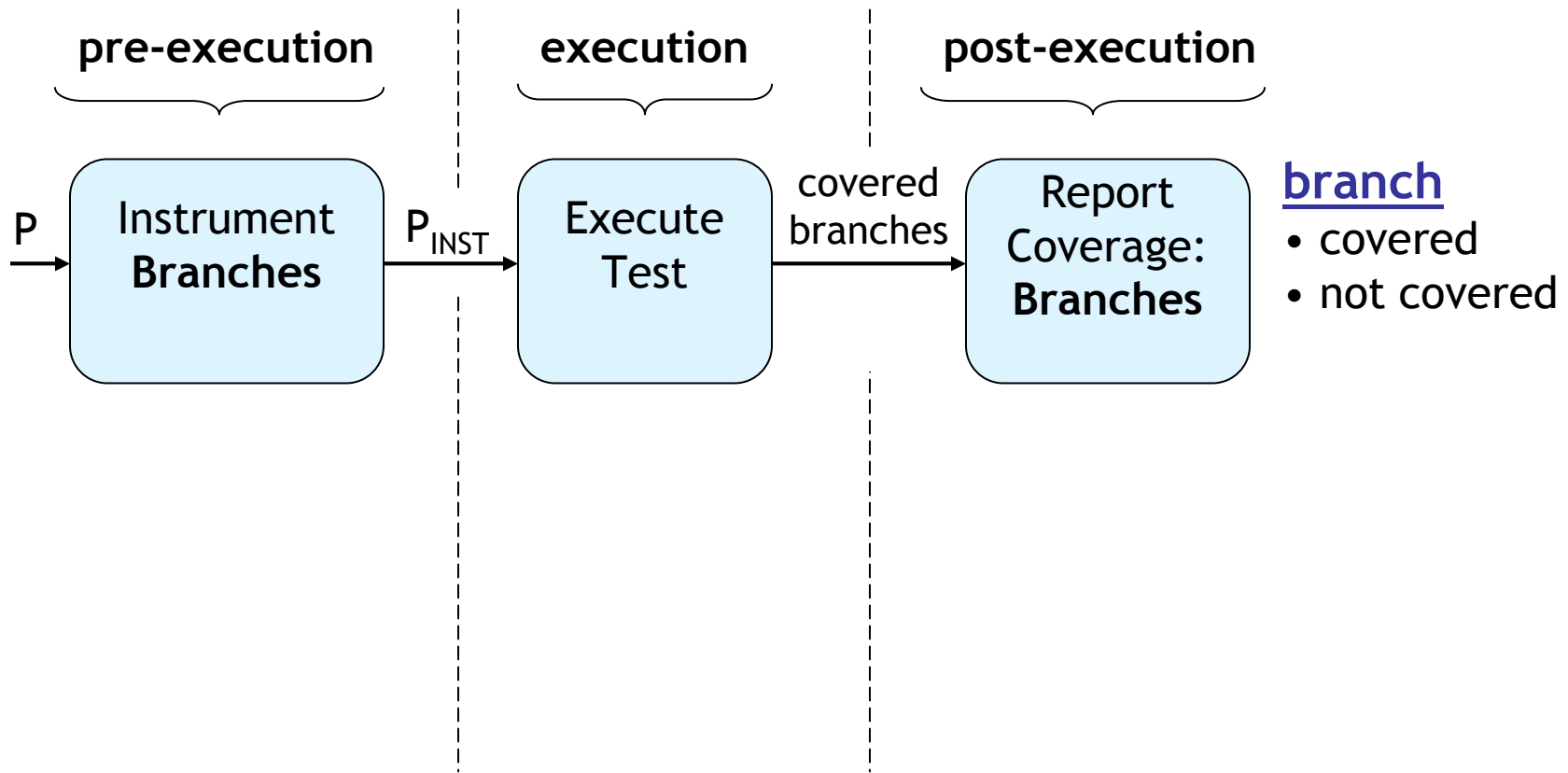
definition-use
association (DUA)

data-flow entity

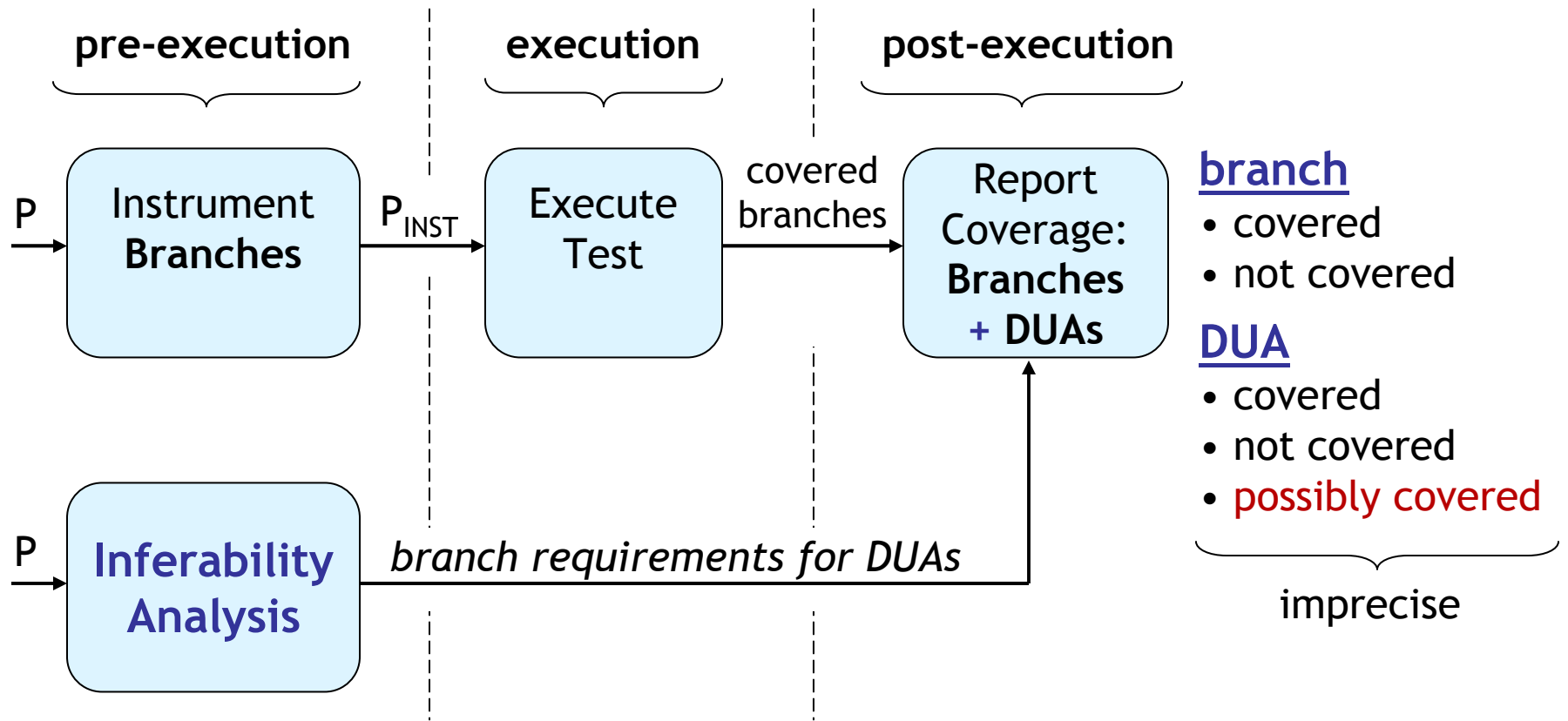
Outline

- Background
- ▶ **Inferability analysis: what**
- Inferability analysis: how
- Study
- Conclusion

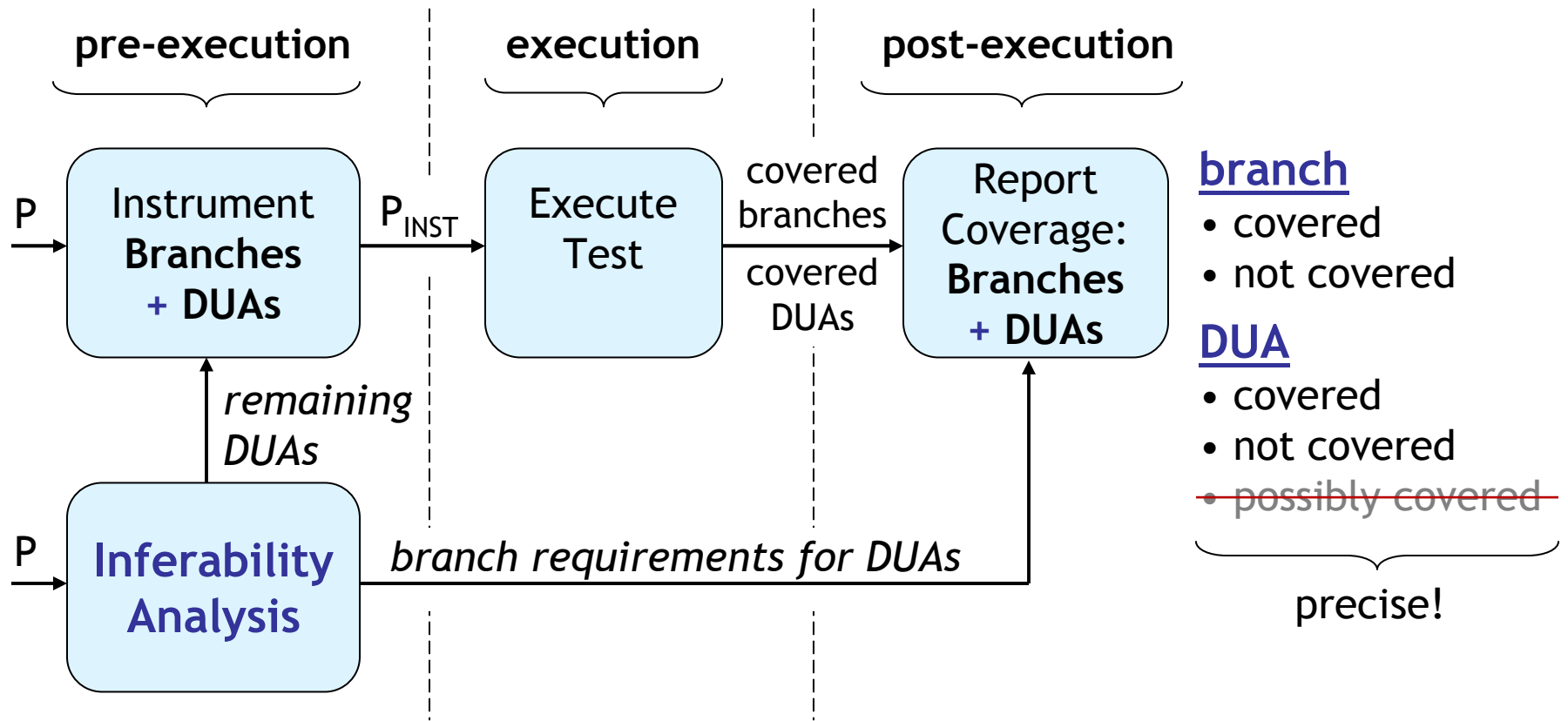
The Big Picture



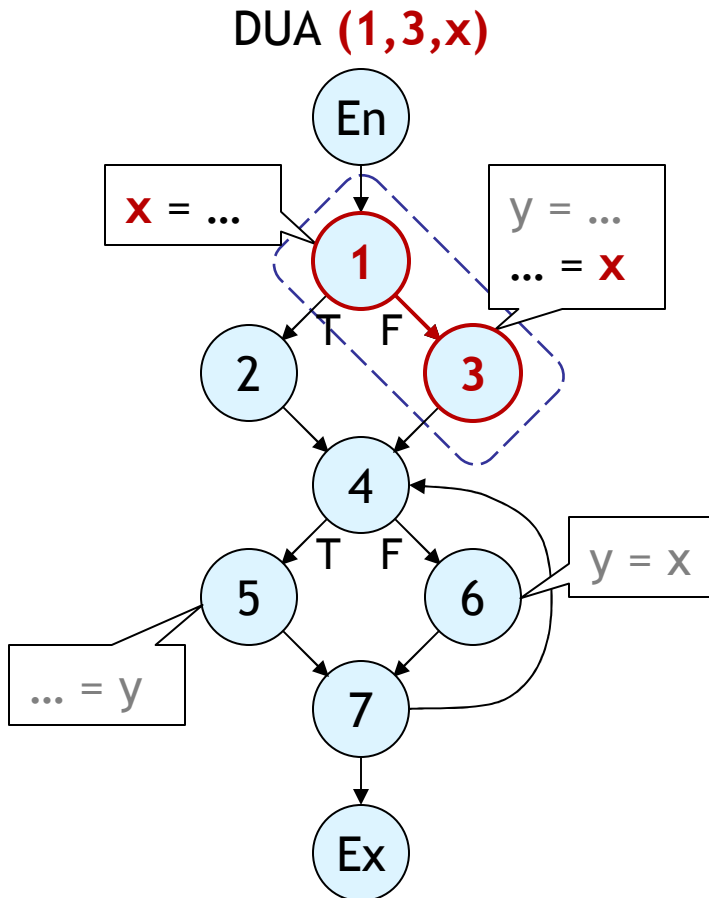
The Big Picture



The Big Picture

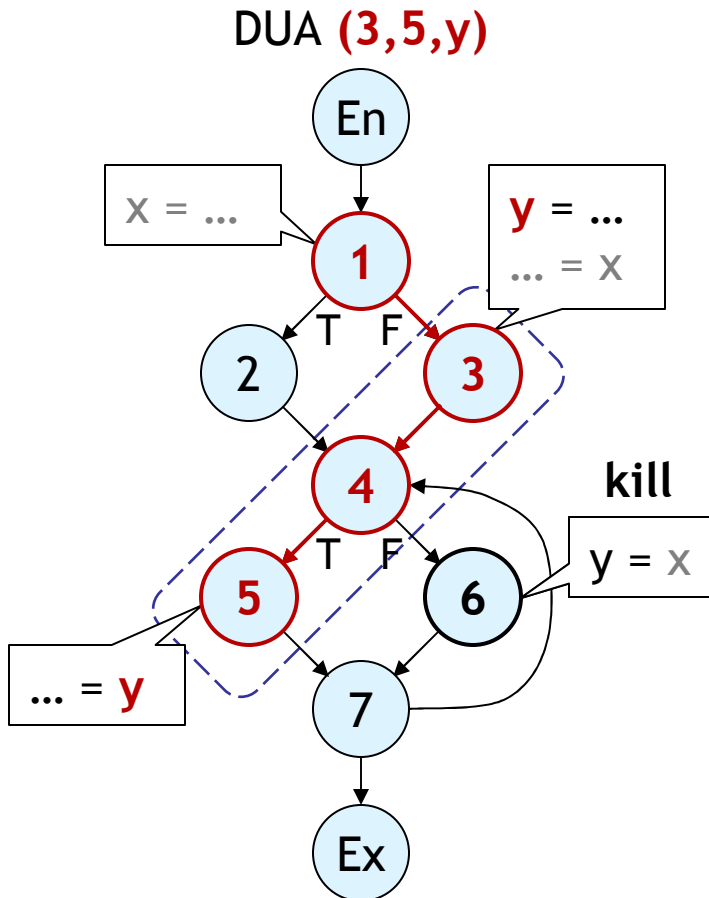


Relating DUAs to Branches



$1F \Rightarrow (1,3,x)$ is covered

Relating DUAs to Branches

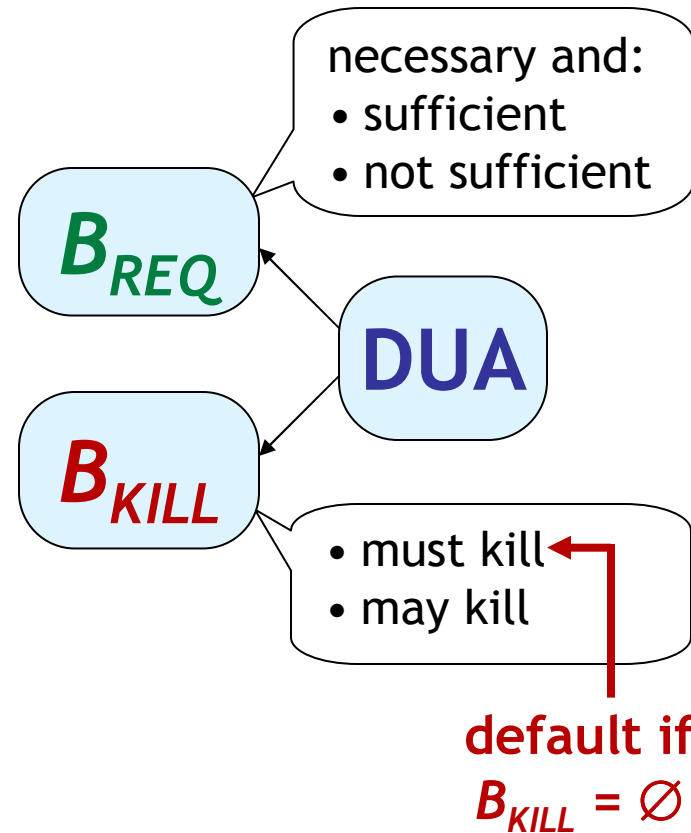
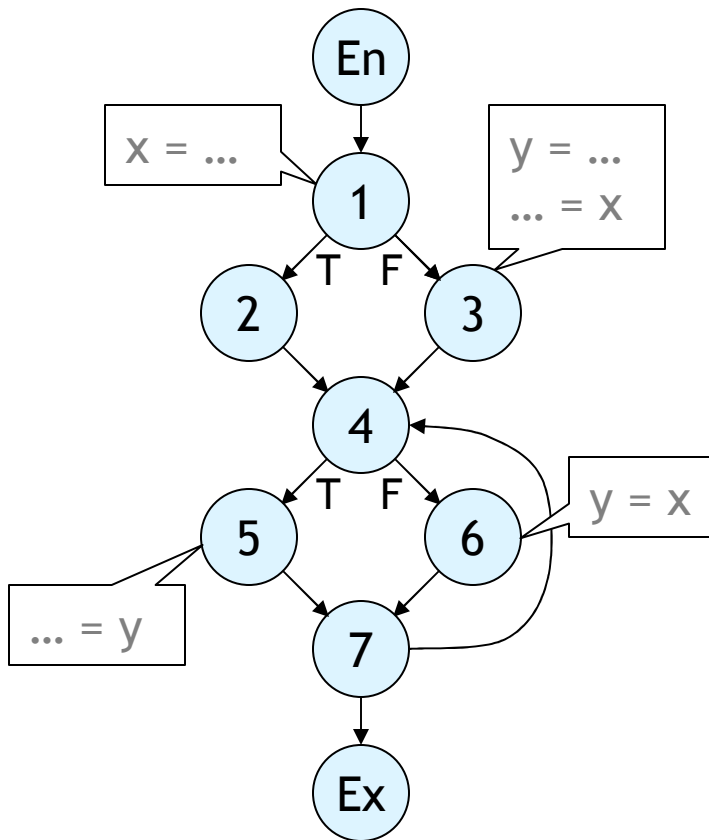


$1F \Rightarrow (1,3,x)$ is covered

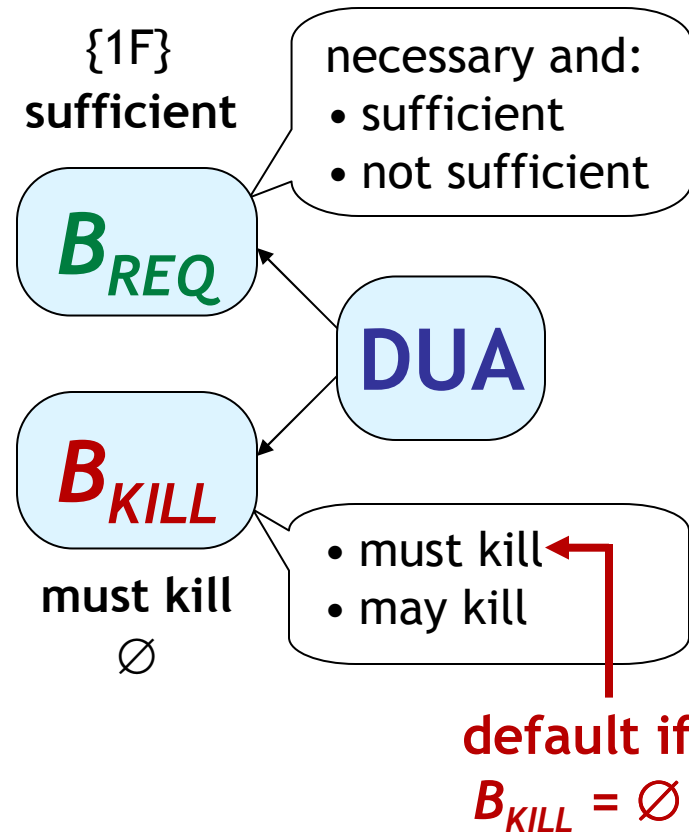
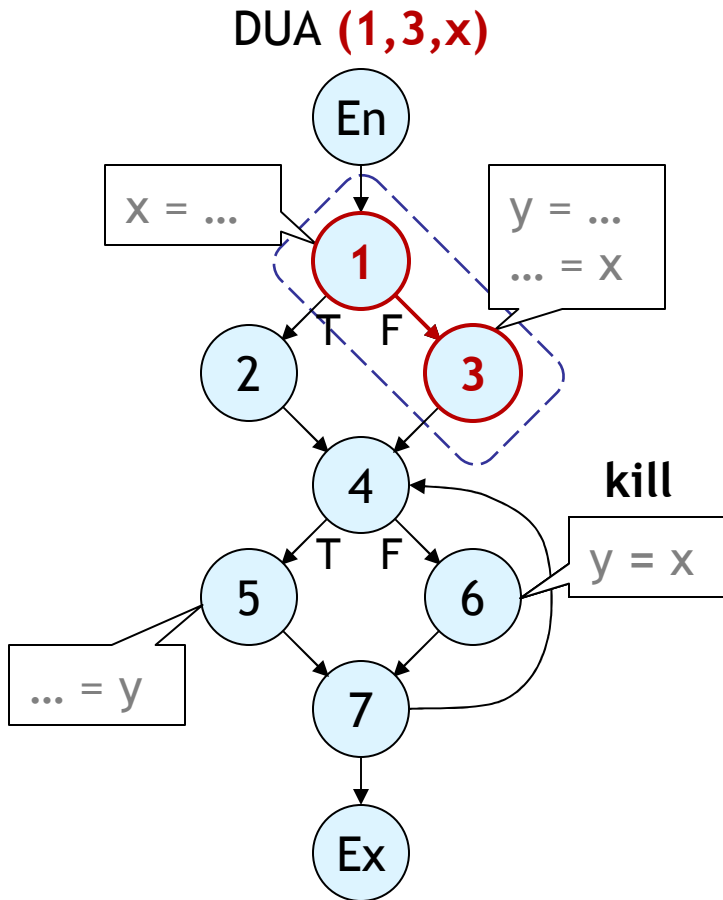
$1F,4T \Rightarrow (3,5,y)$ is covered

- if y is not **killed** (redefined) at node 6

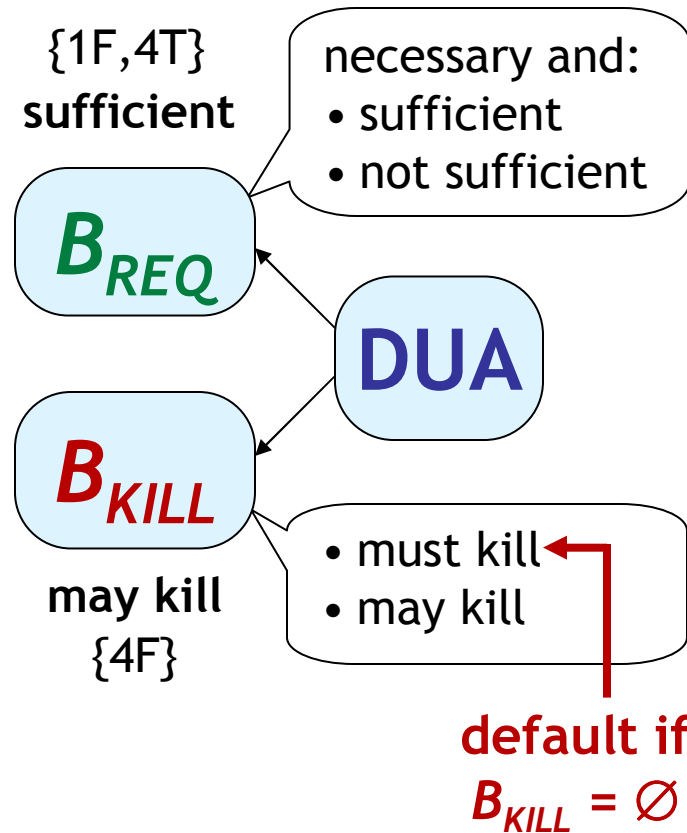
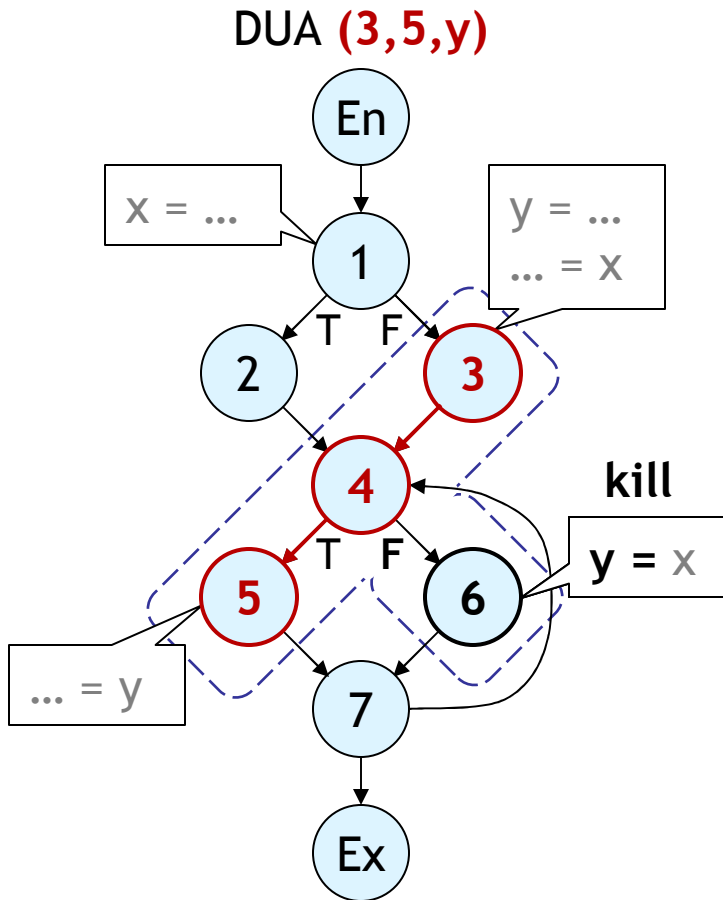
Relating DUAs to Branches



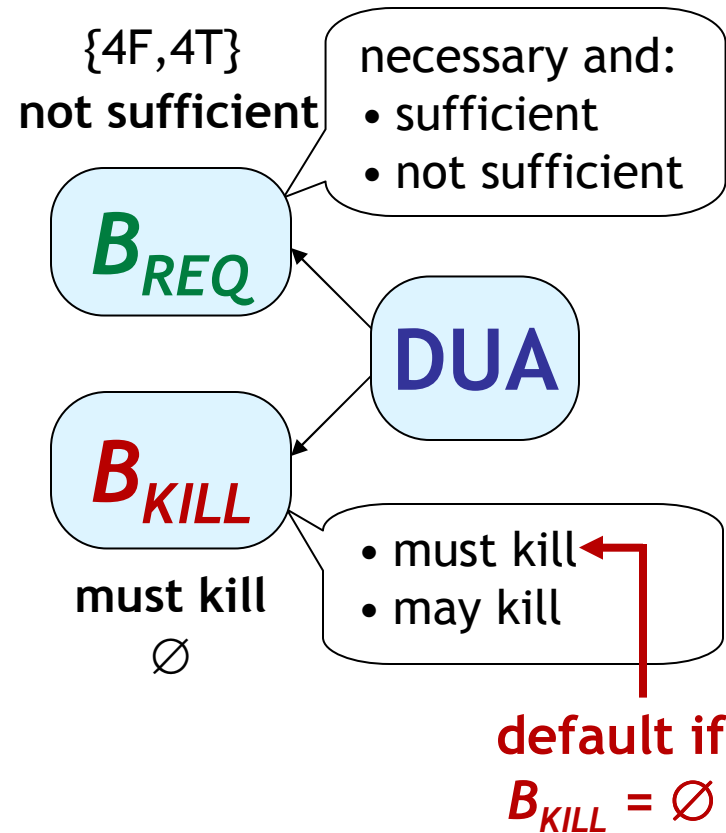
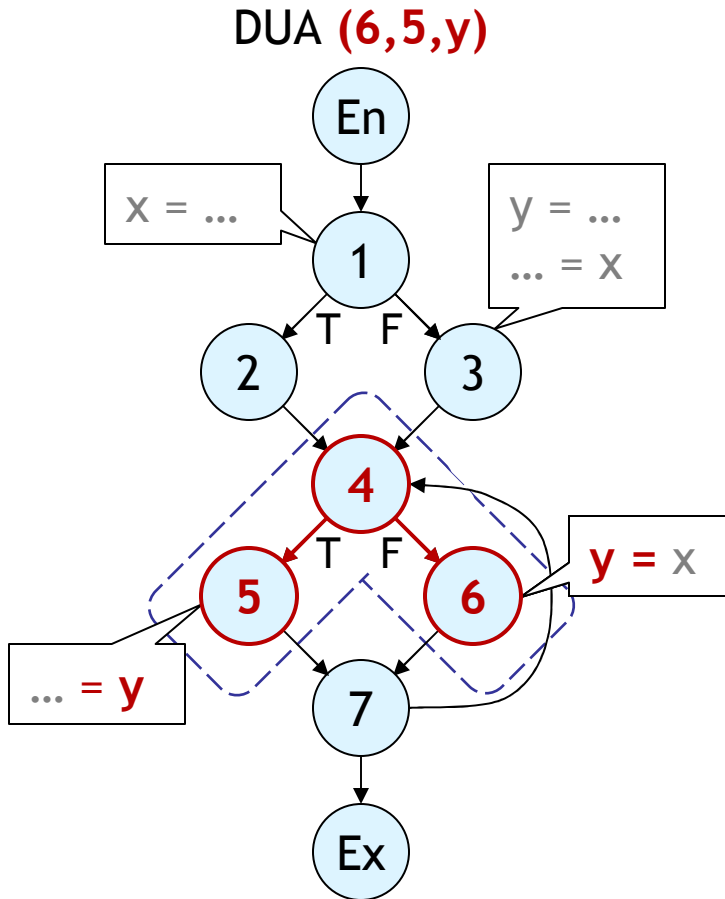
Relating DUAs to Branches



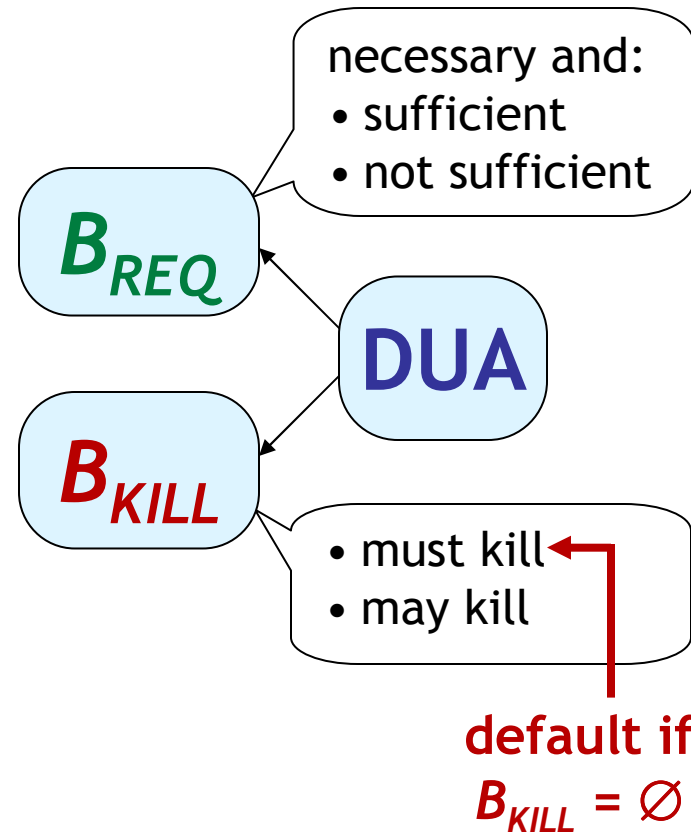
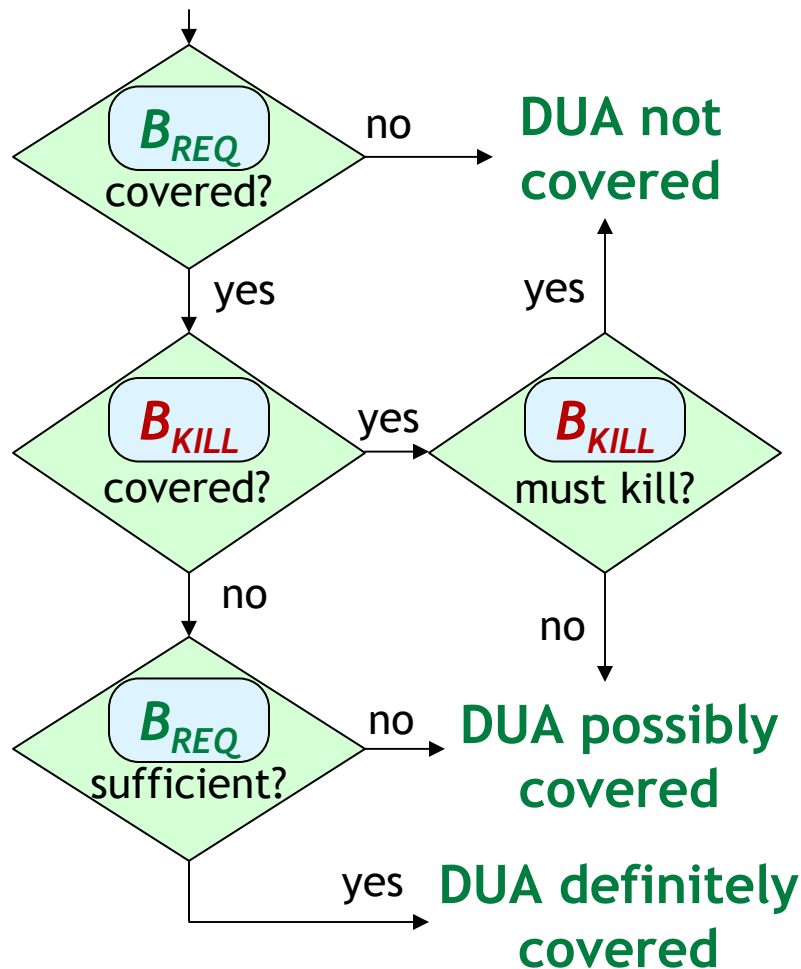
Relating DUAs to Branches



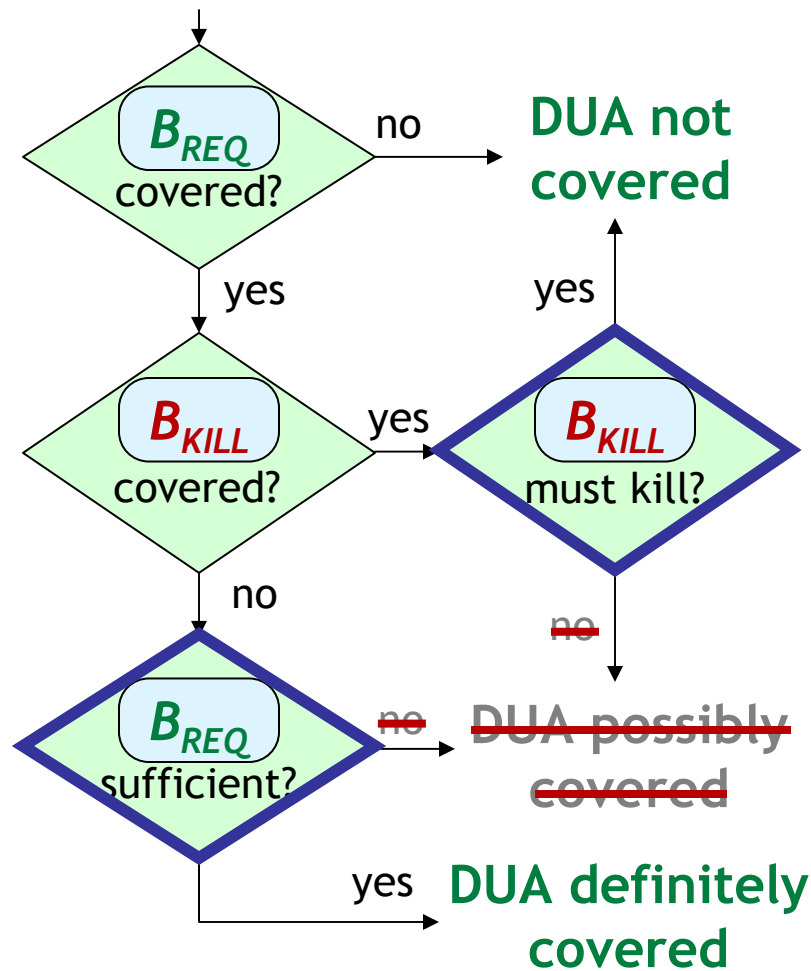
Relating DUAs to Branches



Deciding Coverage of a DUA



Classifying DUAs

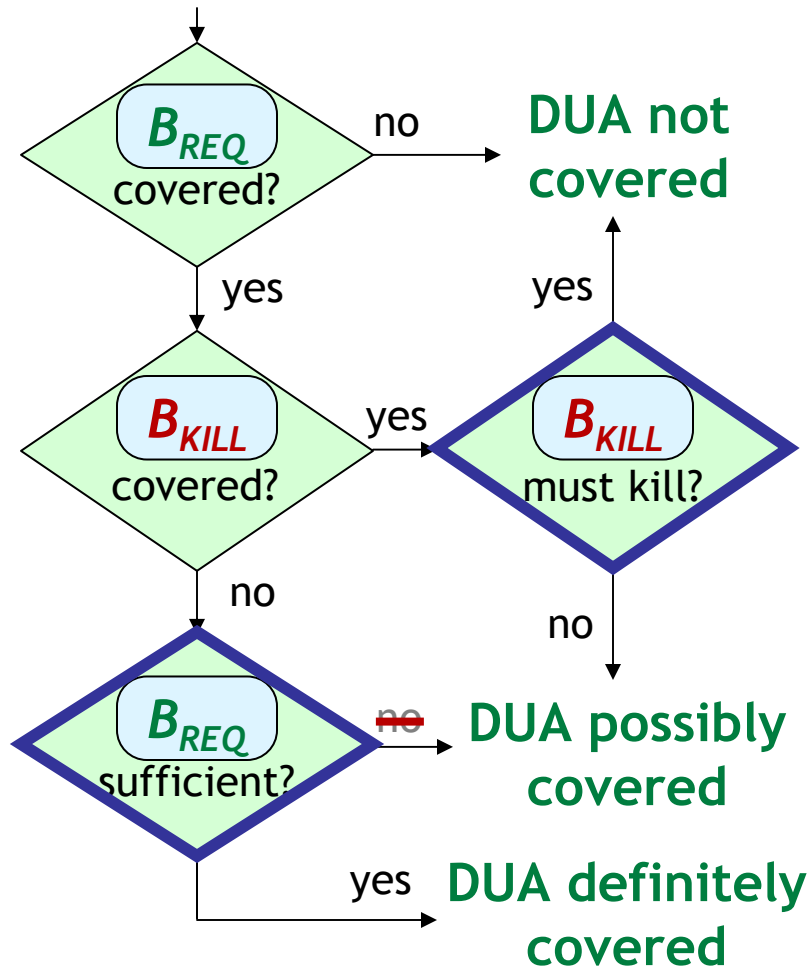


inferable

B_{REQ} sufficient

B_{KILL} must kill

Classifying DUAs



inferable

B_{REQ} sufficient

B_{KILL} must kill

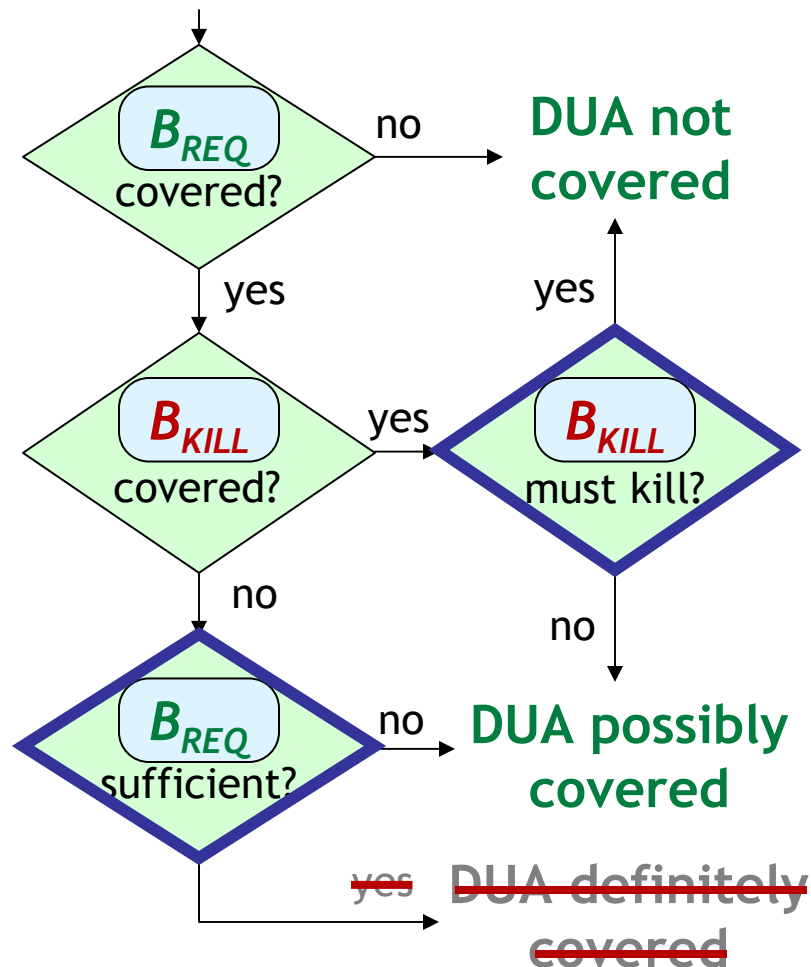
remaining DUAs:

conditionally-inferable

B_{REQ} sufficient

B_{KILL} contains a may-kill

Classifying DUAs



inferable

B_{REQ} sufficient

B_{KILL} must kill

remaining DUAs:

conditionally-inferable

B_{REQ} sufficient

B_{KILL} contains a may-kill

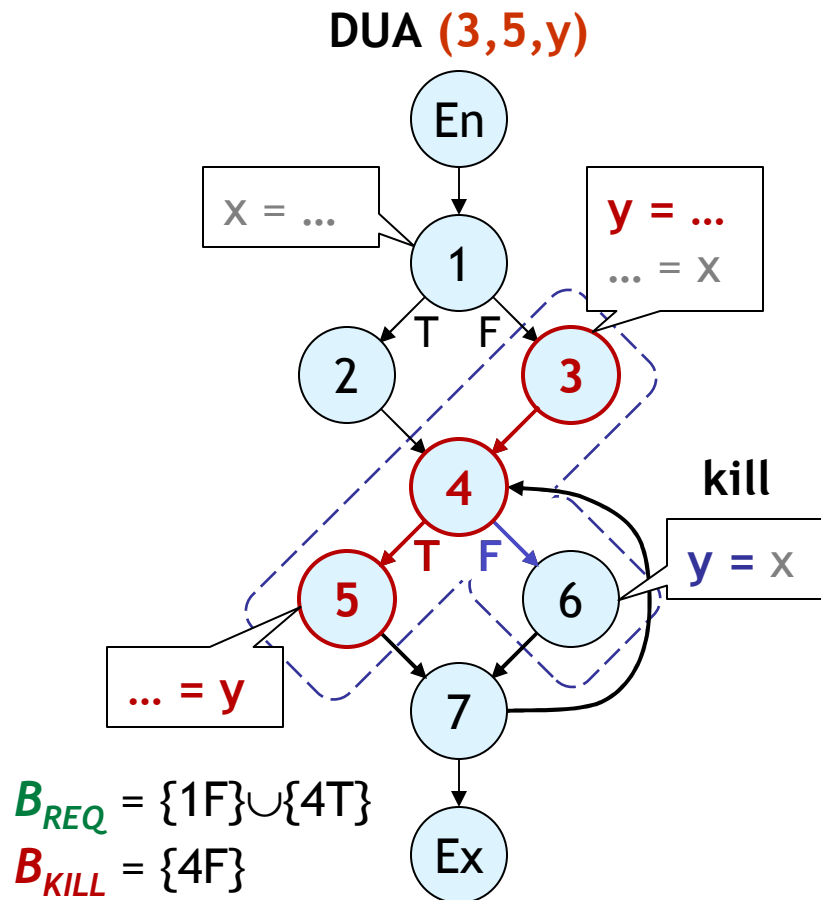
non-inferable

B_{REQ} not sufficient

Outline

- Background
- Inferability analysis: what
- ▶ **Inferability analysis: how**
- Study
- Conclusion

1. Computing B_{REQ} and B_{KILL}

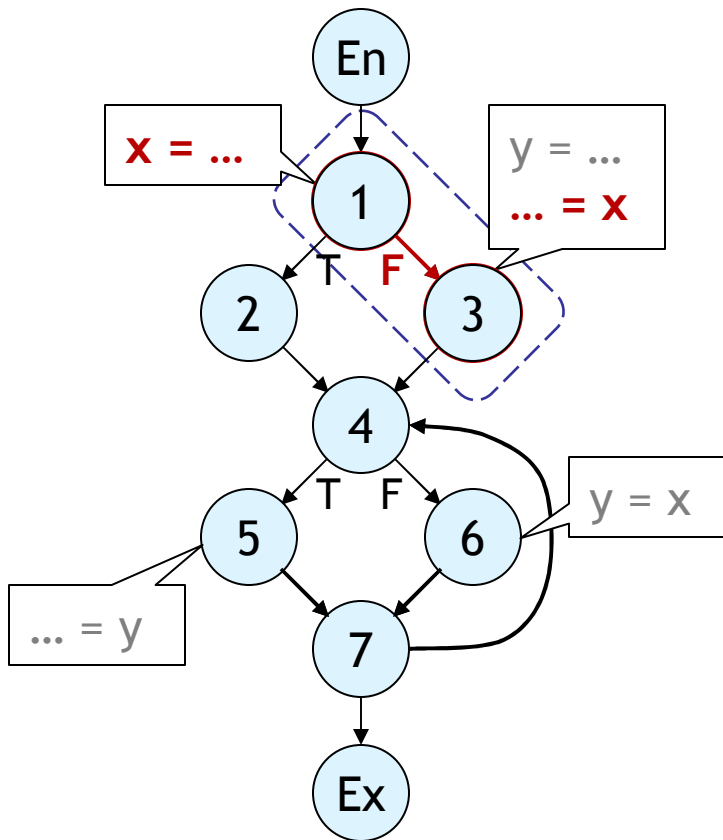


- Control-dependence:
 - node N is control-dependent on branch B if taking B implies N covered
- For DUA (d,u,v) :
 - B_{REQ} = union of control-dependences of d and u
 - B_{KILL} = union of control-dependences of all *kills* of (d,u,v)

2. Sufficiency of B_{REQ}

- Nodes d and u are in **node order** if:
 - (1) there exists no path from u to d ; or
 - (2) d dominates u ; or
 - (3) u postdominates d
- B_{REQ} is **sufficient** if definition and use are in node order

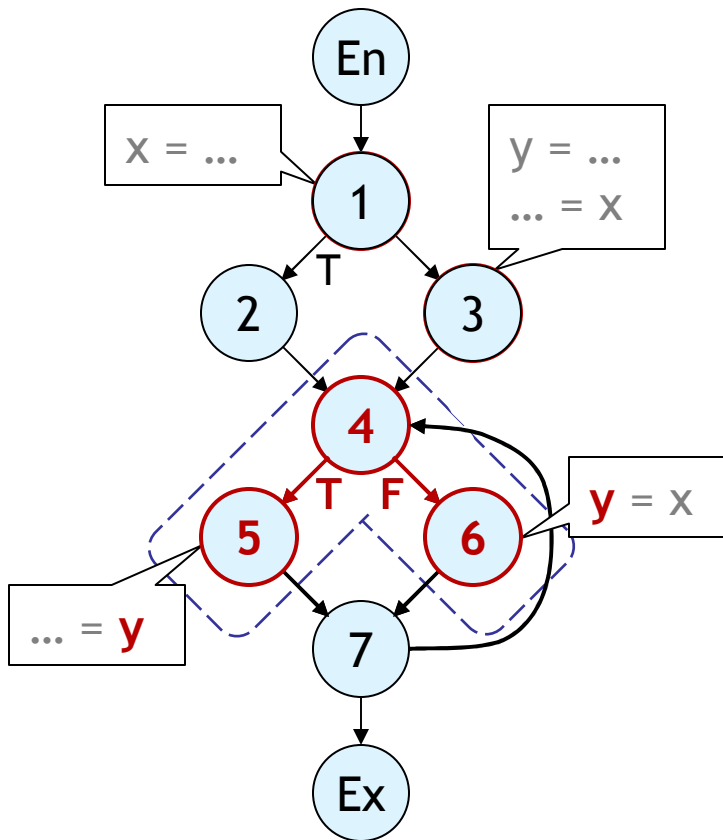
2. Sufficiency of B_{REQ}



$(1, 3, x)$: B_{REQ} is sufficient

- 1 is not reachable from 3
- 1 dominates 3

2. Sufficiency of B_{REQ}



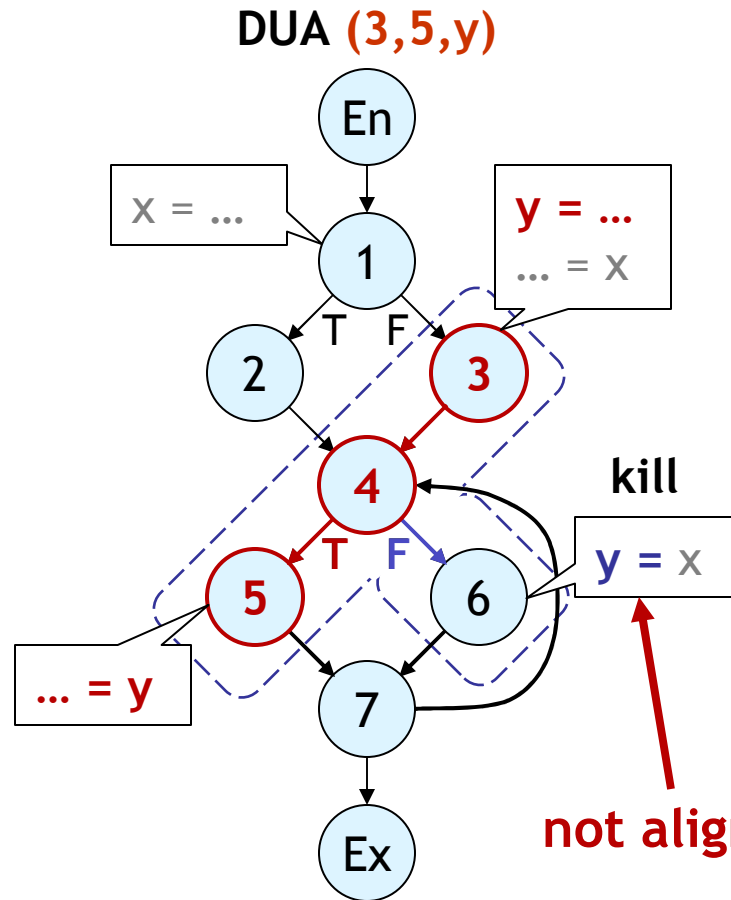
(1,3,x): B_{REQ} is sufficient

- 1 is not reachable from 3
- 1 dominates 3

(6,5,y): B_{REQ} not sufficient

- there is a path from 5 to 6
- 6 does not dominate 5
- 5 does not postdominate 6

3. Must vs. May B_{KILL}



- Kill k is **aligned** with DUA (d,u,v) if:
 - (d,k) and (k,u) are in **node order**, and
 - d is not reachable from u
- B_{KILL} **must** kill if all *kills* aligned

Outline

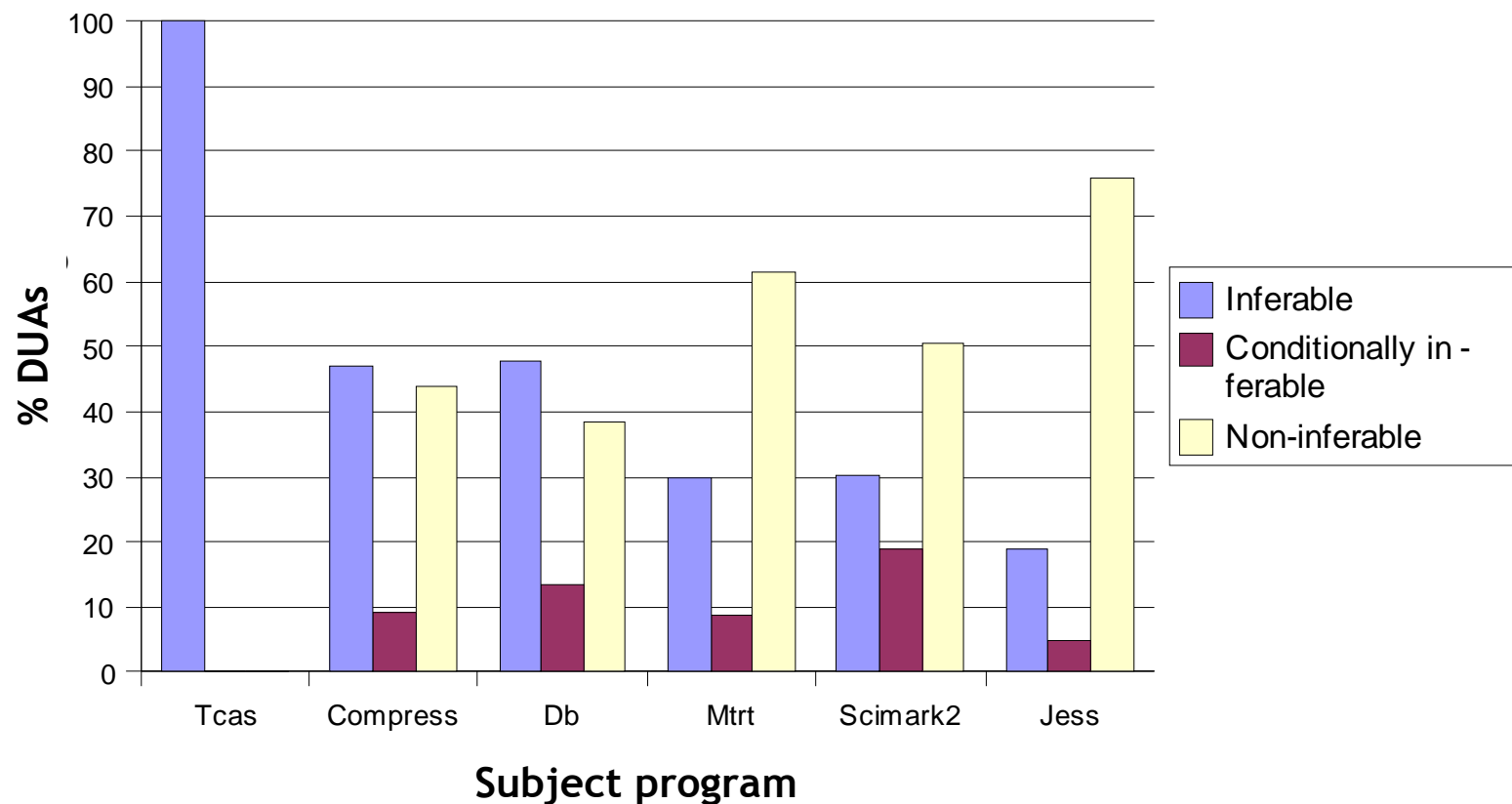
- Background
- Inferability analysis: what
- Inferability analysis: how
- ▶ **Study**
- Conclusion

Study and Tool

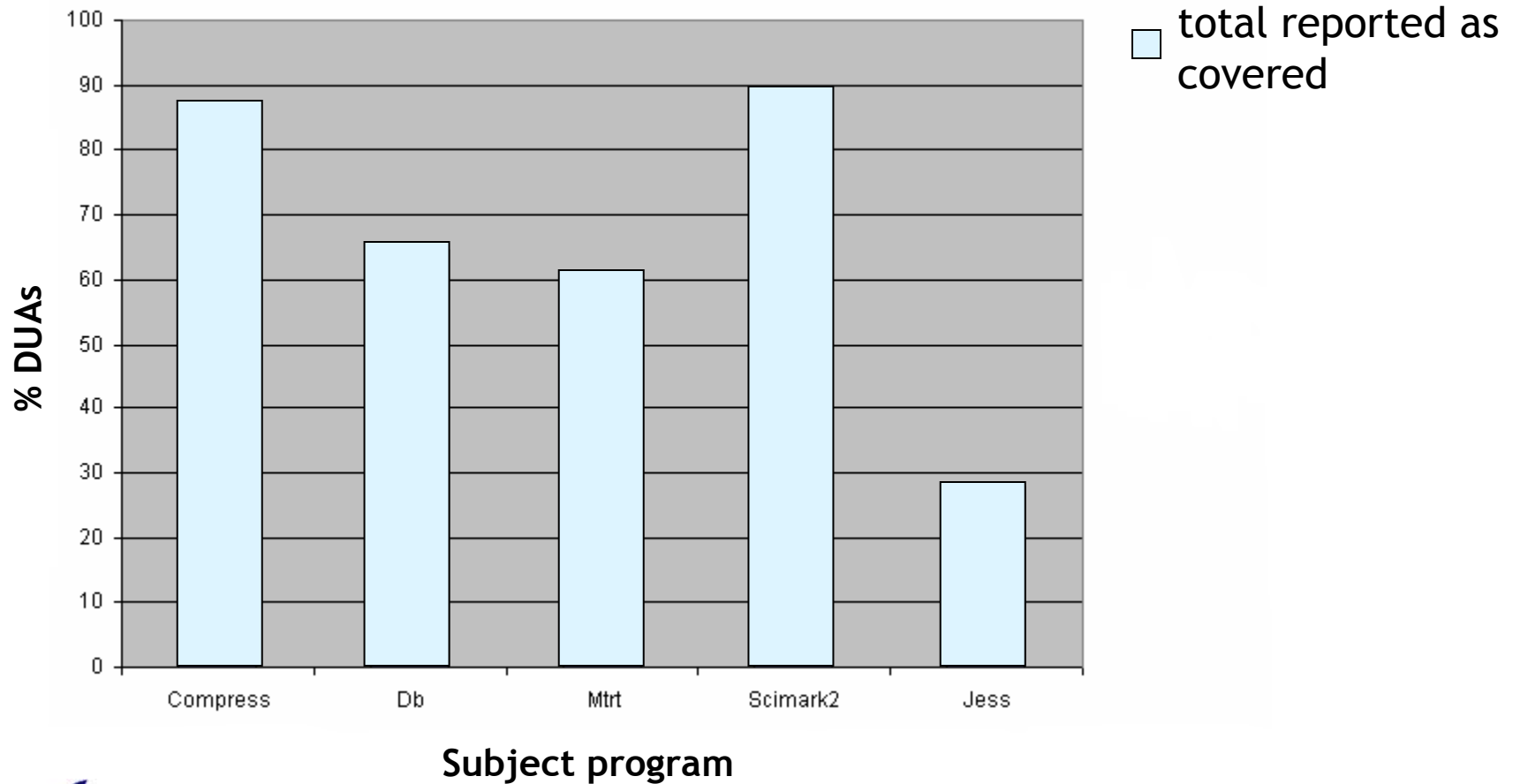
- Implemented tool: **DUA-Forensics**
 - Java, interprocedural, local variables
- Subjects:

Subject	Lines of Code	# DUAs
Tcas	150	136
Compress	587	445
Db	663	526
Mtrt	1242	101
Scimark2	1805	1054
Jess	6365	12927

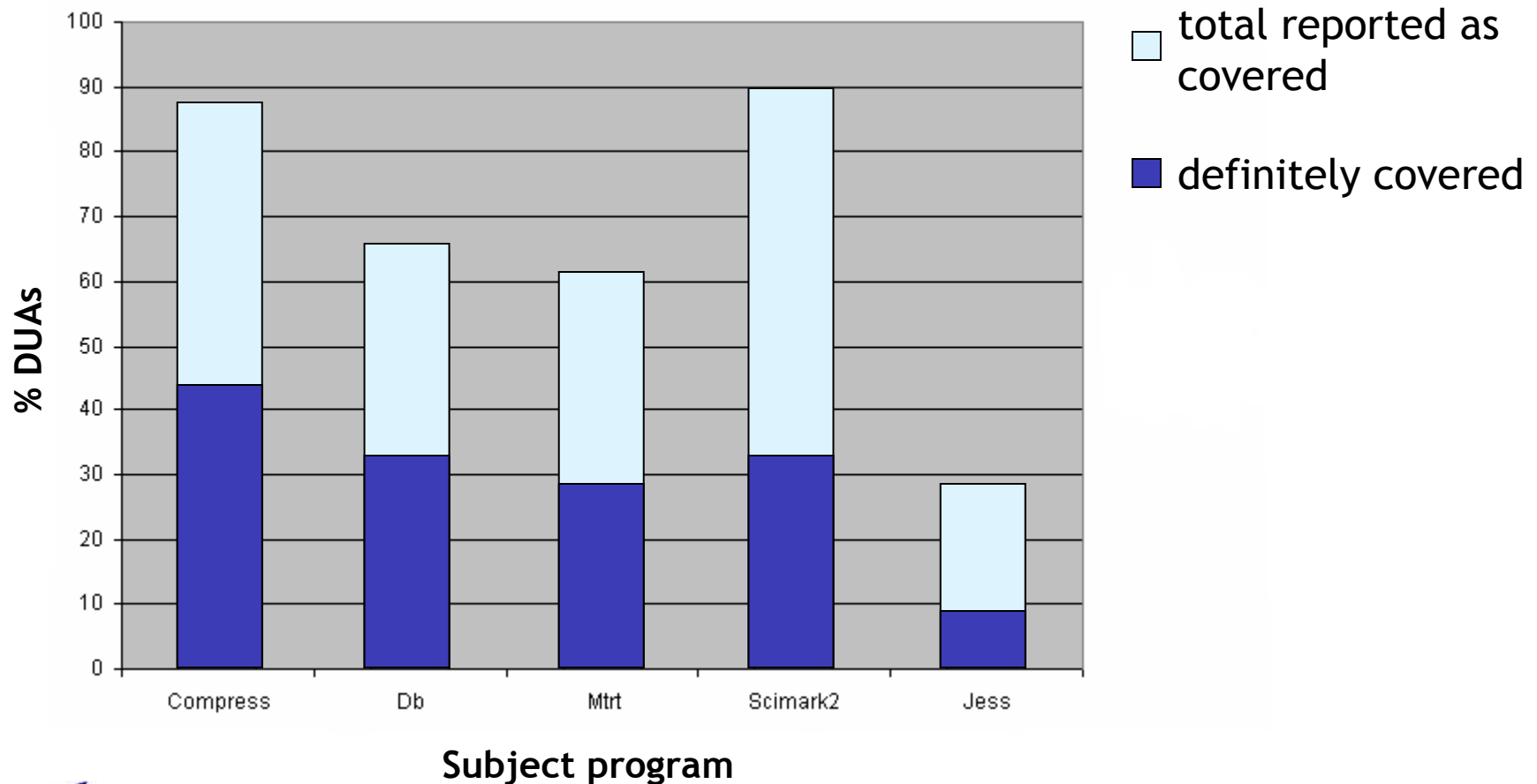
Study 1: Distribution of DUA Inferability Types



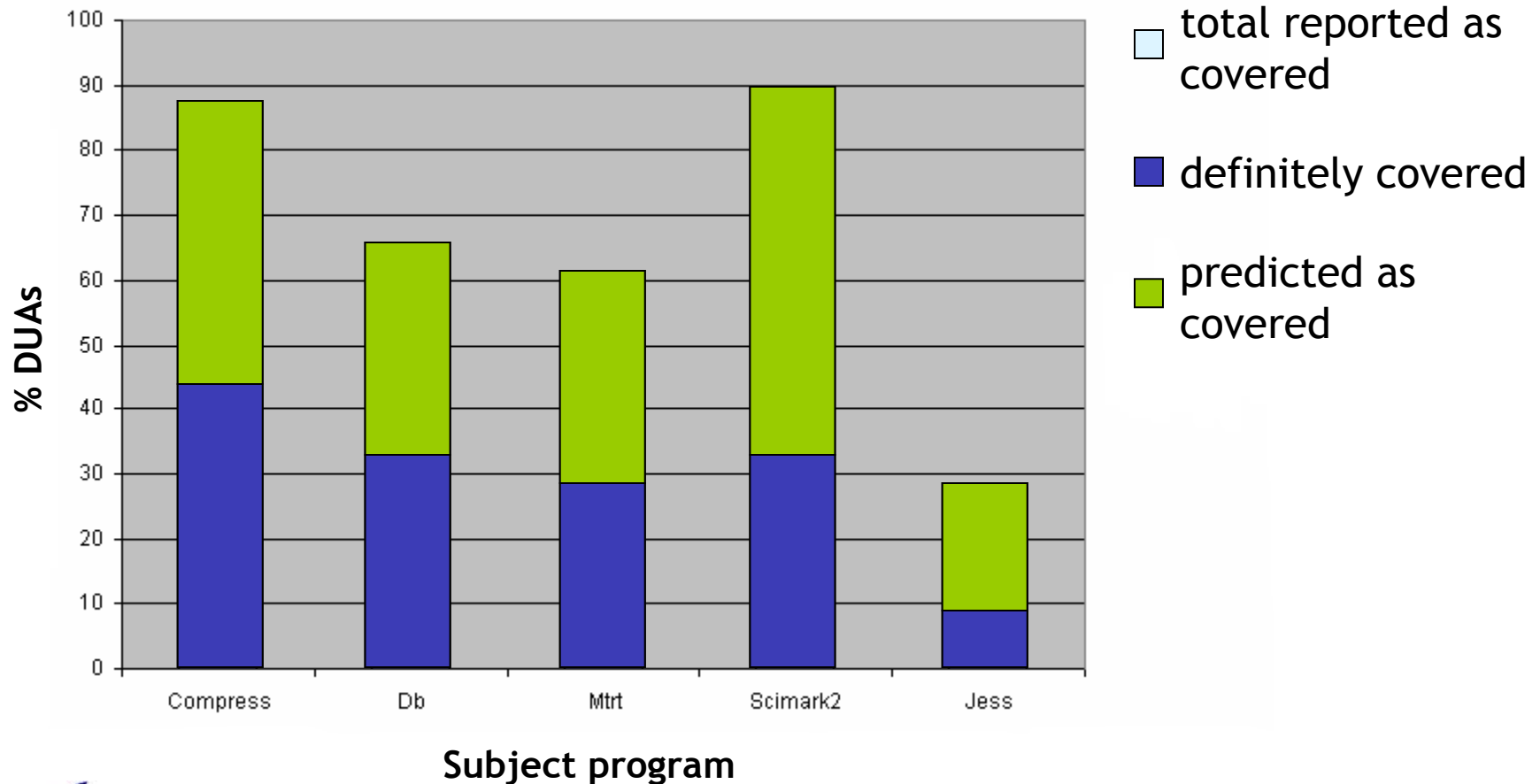
Study 2: Precision of Branch-based DUA Coverage Report



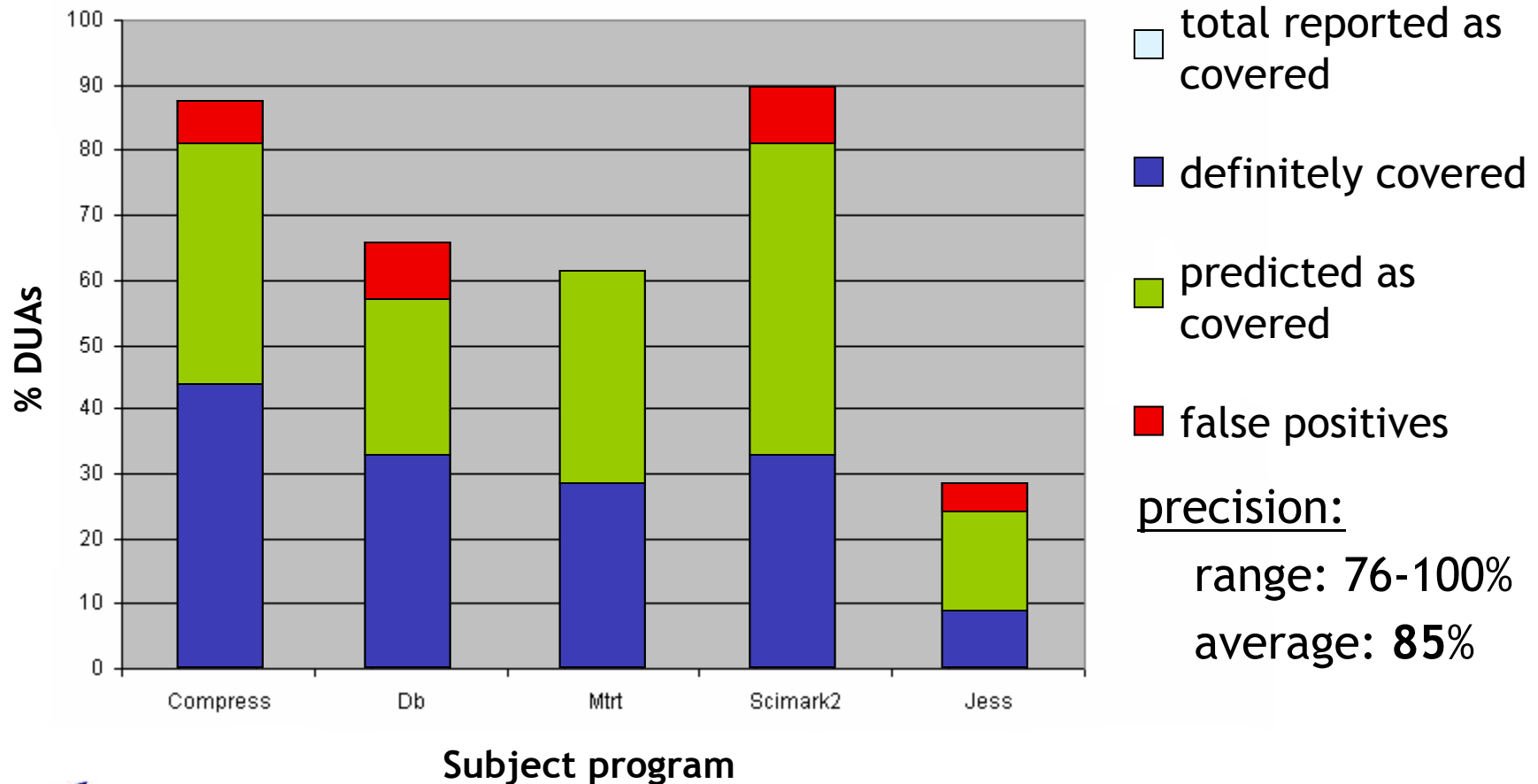
Study 2: Precision of Branch-based DUA Coverage Report



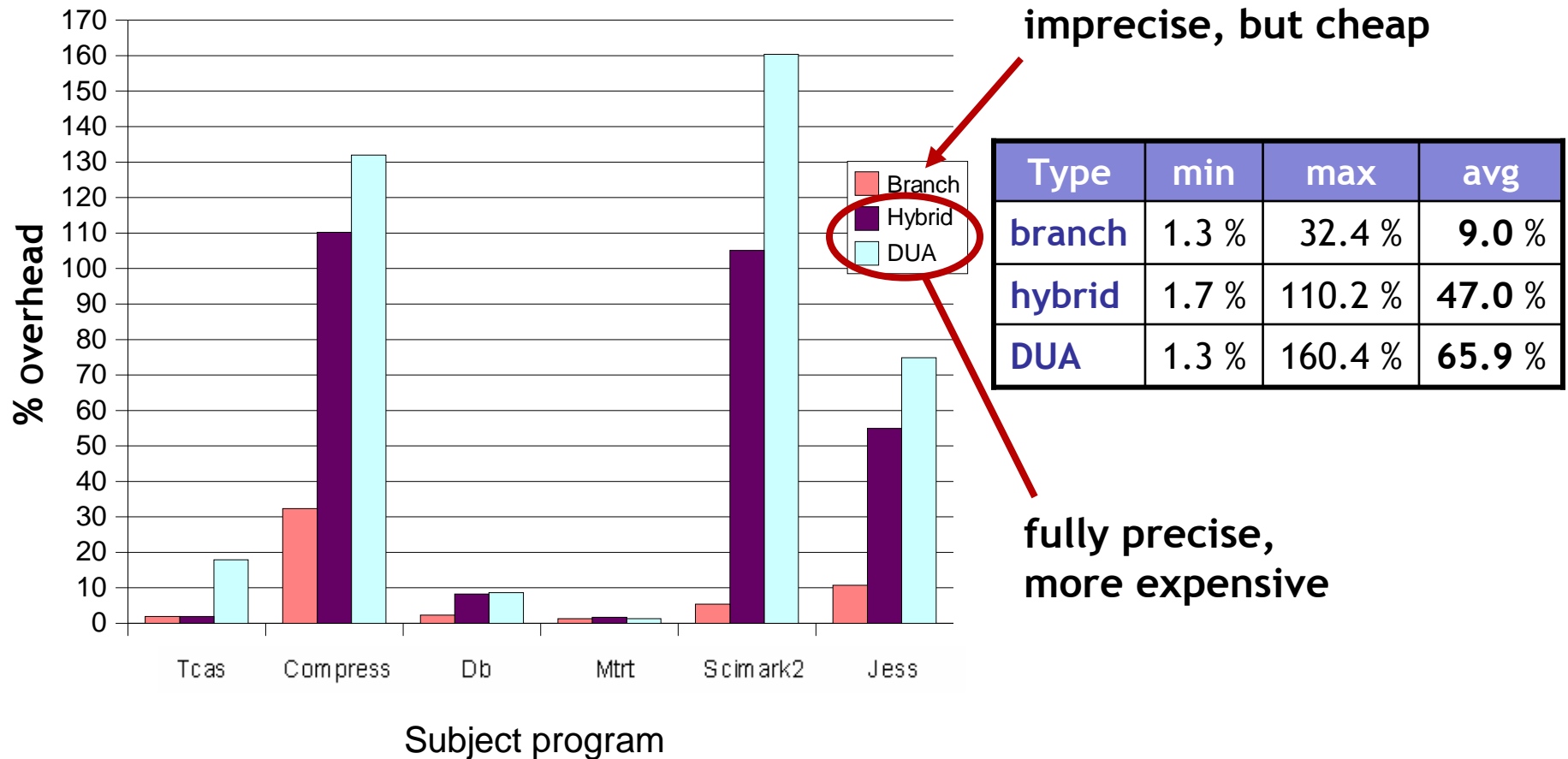
Study 2: Precision of Branch-based DUA Coverage Report



Study 2: Precision of Branch-based DUA Coverage Report



Study 3: Branch and DUA Monitoring Overheads



Outline

- Background
- Inferability analysis: what
- Inferability analysis: how
- Study
- ▶ **Conclusion**

Conclusion

- Inferability analysis and studies:
 - Monitor DUAs using only branches, with good estimate accuracy
 - Hybrid mix for precise DUA monitoring, more efficient than traditional approach
 - Support decision on branch-based vs. precise DUA monitoring

Future Work

- Study complications of aliasing
- Infer DUAs from **acyclic paths** instead of branches
- Infer DUAs or other flows for debugging, security
- Inferability types distribution as a data-flow complexity metric

Questions?