Secure Virtualization with Formal Methods

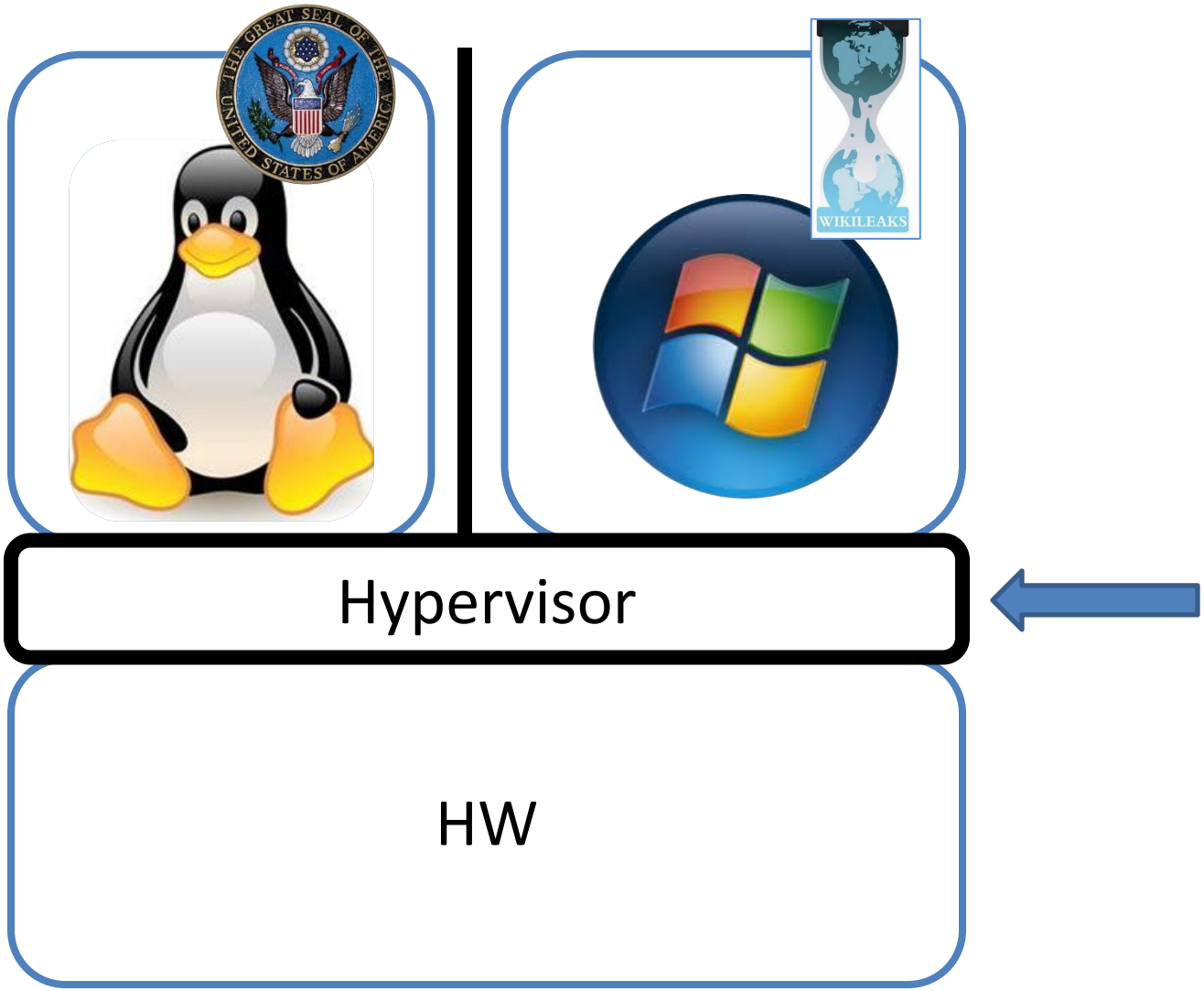
Cynthia Sturton, Rohit Sinha,
Petros Maniatis,
Sanjit A. Seshia, David Wagner

Spring 2013

Cloud Computing: Infrastructure as a Service


NETFLIX
Newsweek
SAN FRANCISCO STATE UNIVERSITY
LinkedIn
amazon.com
The Washington Post
reddit
WIKILEAKS
THE TIMES
THE GREAT SEAL OF THE UNITED STATES OF AMERICA
twitter
zynga


THE GREAT SEAL OF THE UNITED STATES OF AMERICA
E PLURIBUS UNUM
WIKILEAKS
amaz


THE GREAT SEAL OF THE UNITED STATES OF AMERICA
WIKILEAKS
Hypervisor
HW

Vision: Trustworthy Virtualization

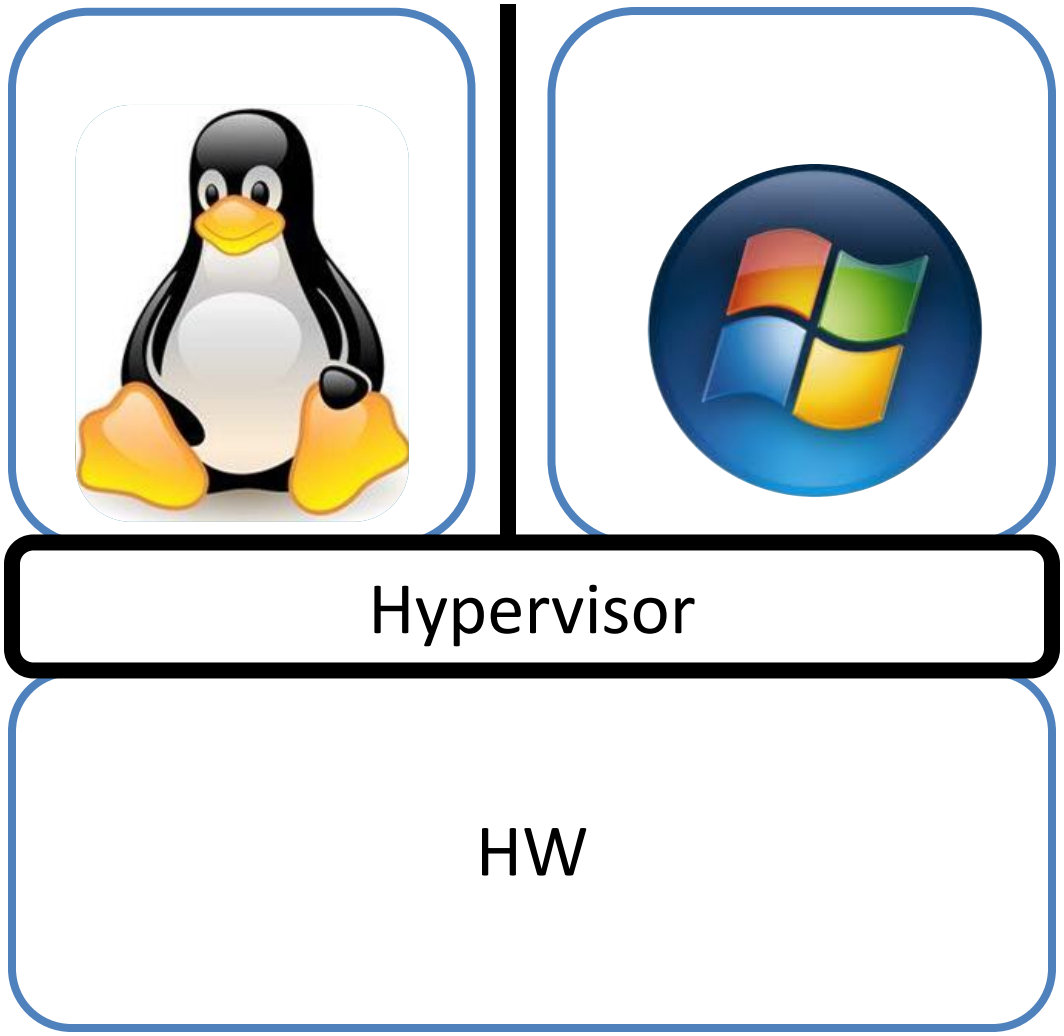
Trustworthy Virtualization

- Large
  - ~150k LOC
- Complex
  - Management of virtual memory
  - Protection of system resources
- Bugs have been found

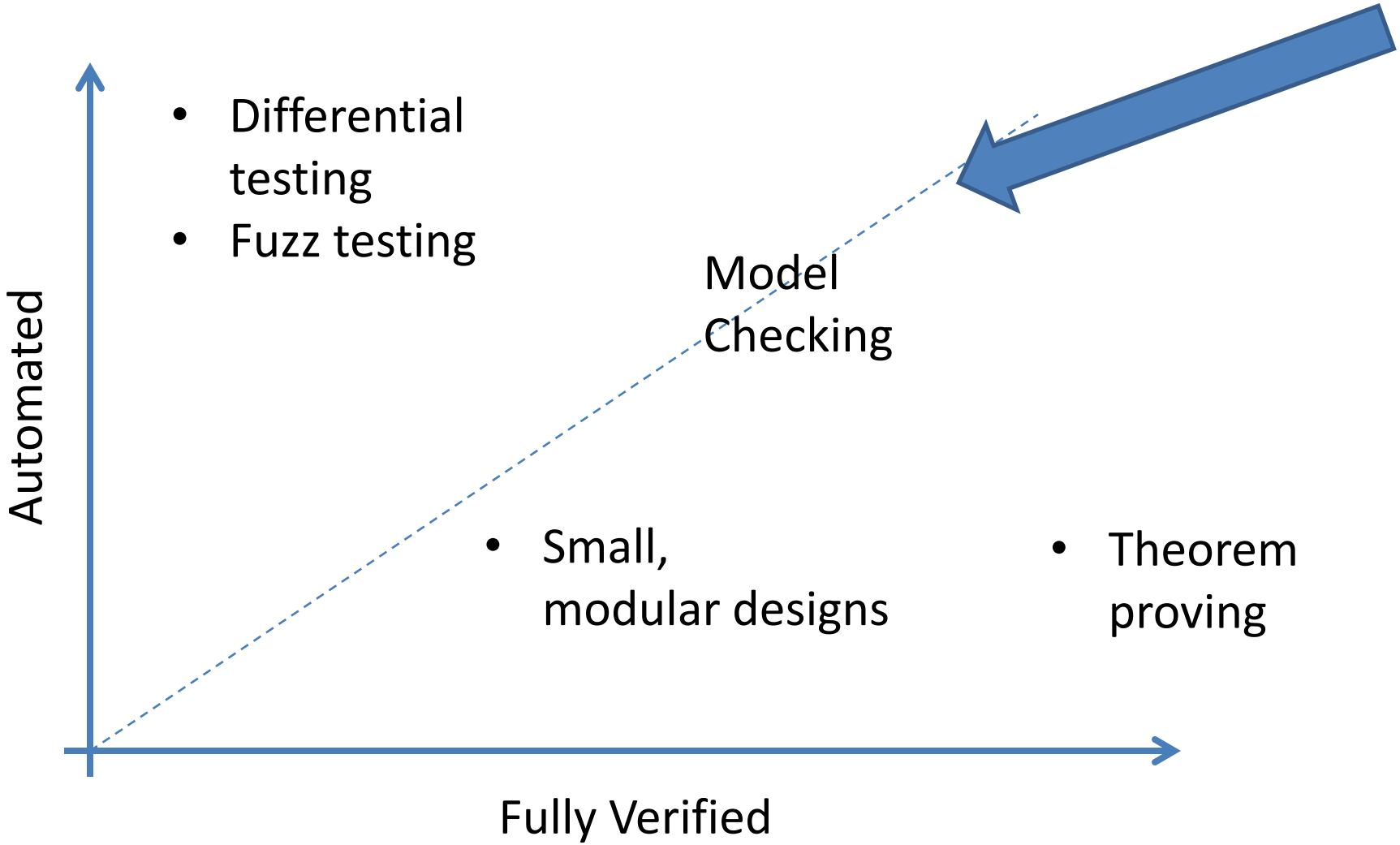
Outline: Trustworthy Virtualization

- Security properties
- Challenges to verification
- Verifying large data structures
- Validating the model

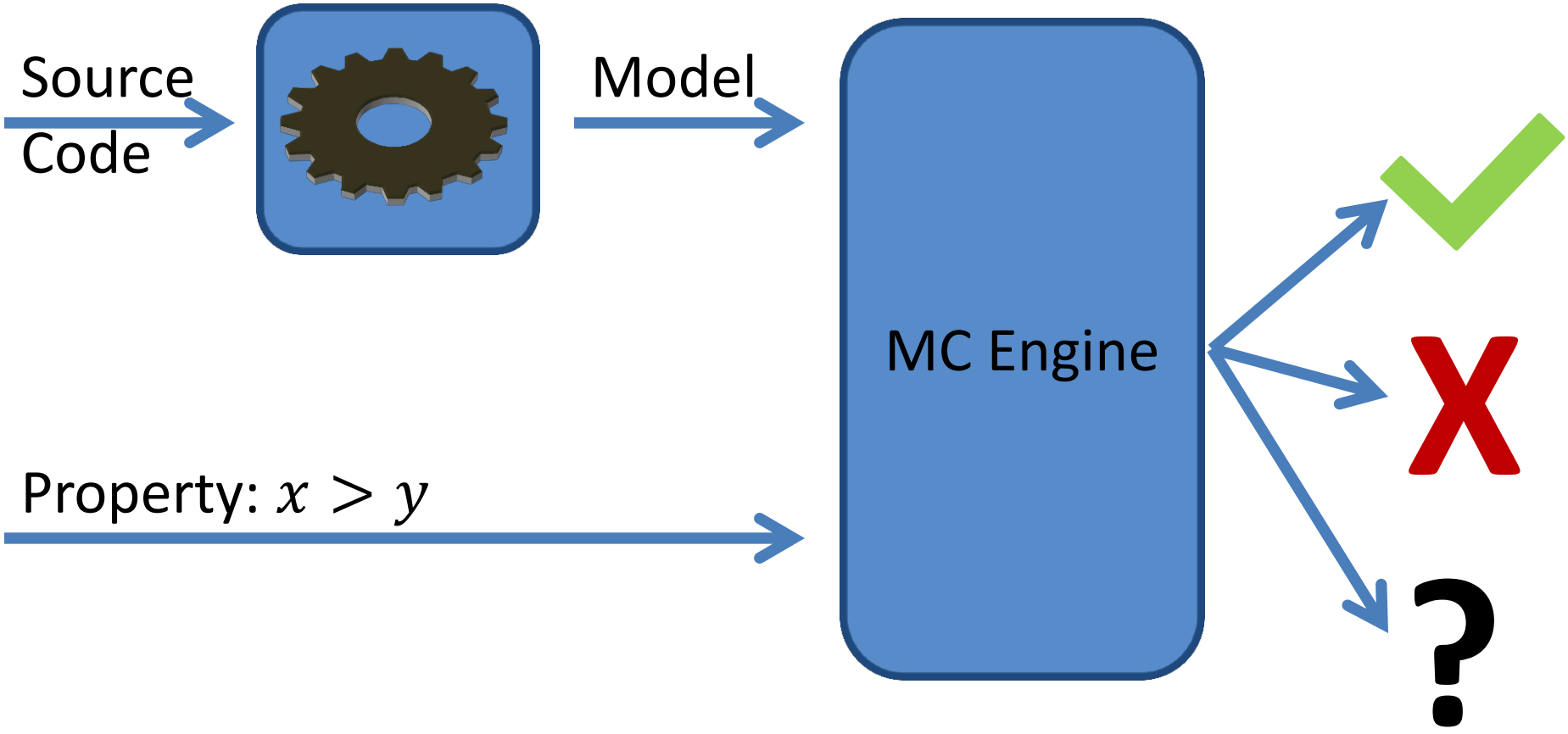
Security Properties



Software Verification



Model Checking



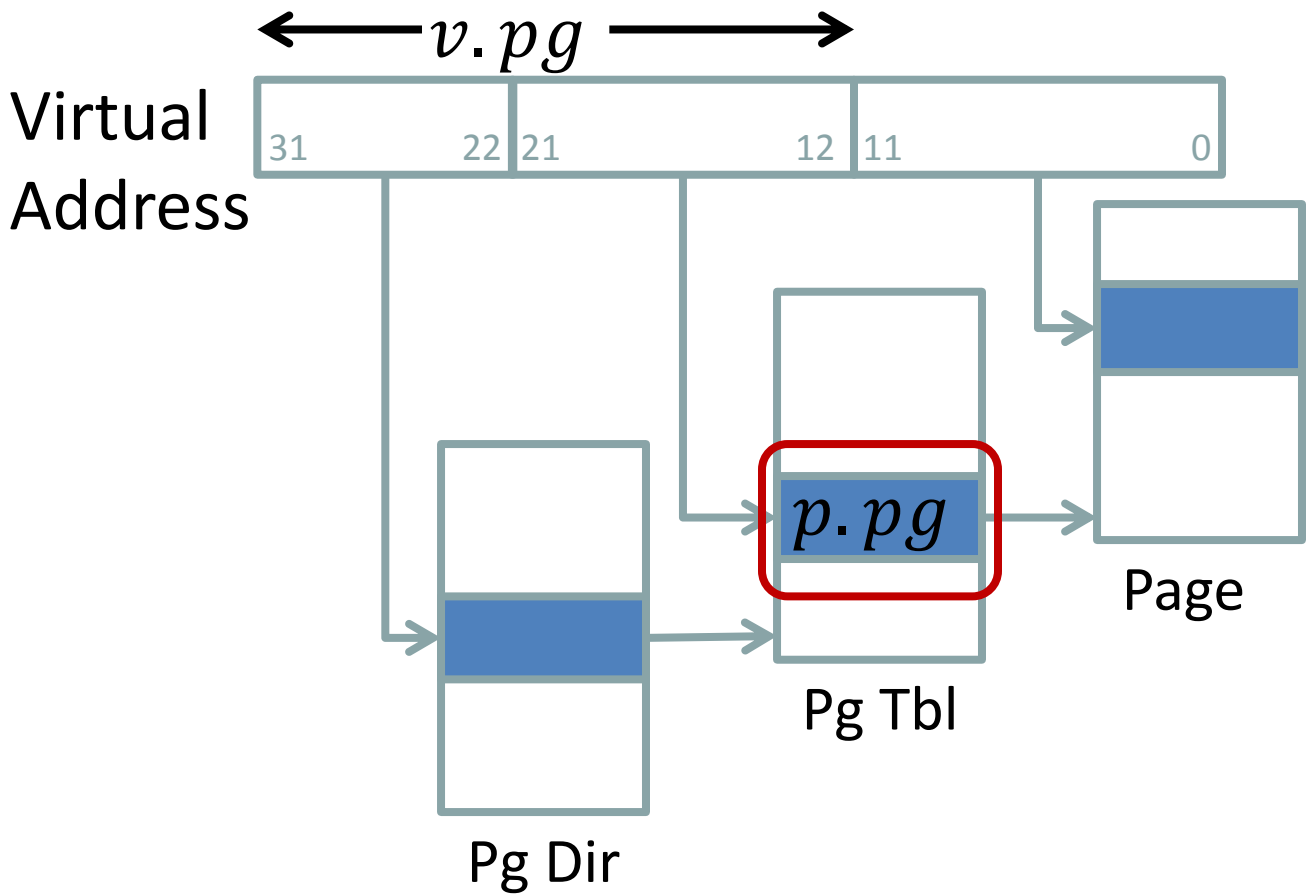
Security Property: Memory Isolation

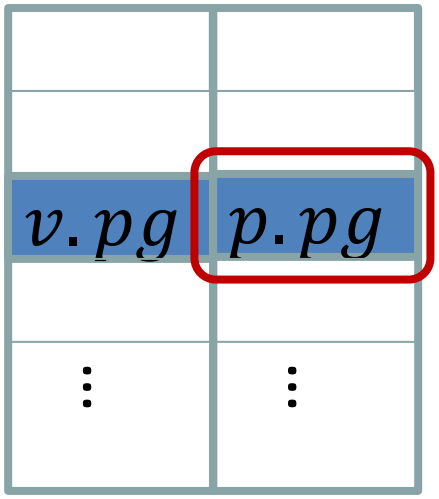
- Guest A can not access Guest B's memory
- Analyze management of virtual memory
- Caveats:
  - Memory isolation ≠ Isolation
  - Assume memory safety

Example Property

Page Table Walk

TLB Look Up



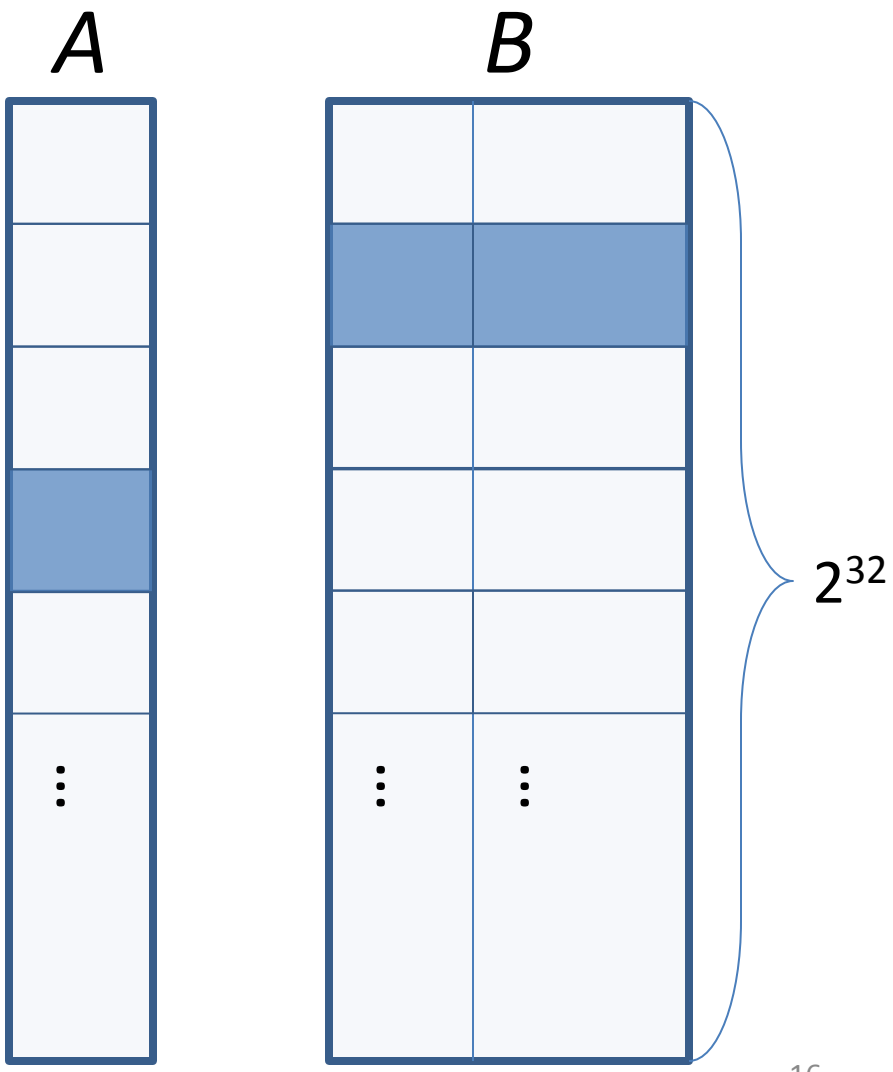


S^2W

Verification of Large Data Structures

Small and Short Worlds (S^2W)

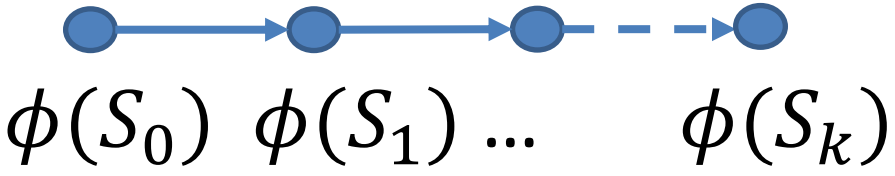
1. Small World:
   - Use a scaled down model
   - Abstraction makes the state space manageable



Small and Short Worlds (S^2W)

1. Small World:
   - Use a scaled down model
2. Short World:
   - Bounded model checking (BMC)
   - Bound := reachability diameter

Verify:
ϕ is an invariant



Reachability Diameter

Every state reachable in k steps is also reachable in $k-1$ steps →

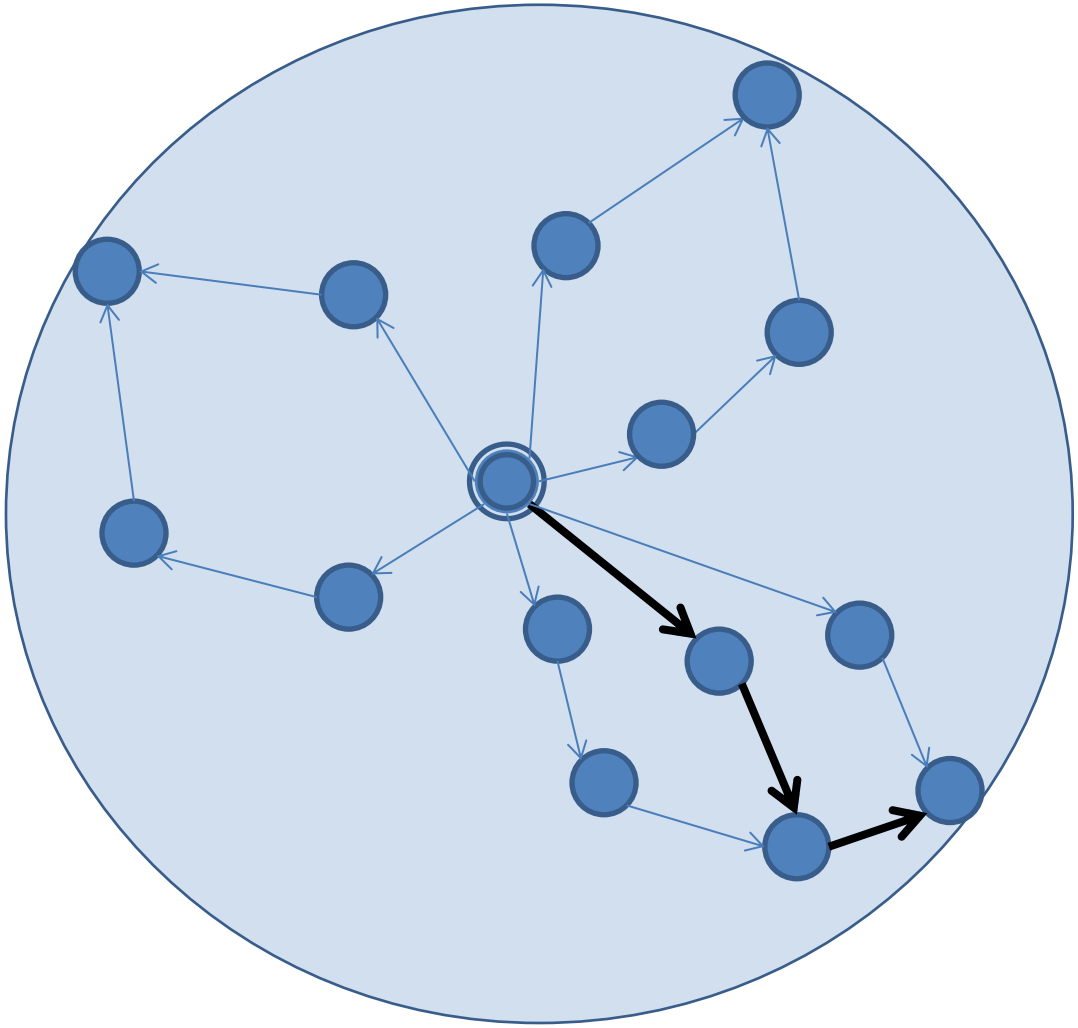
Every reachable state is reachable in $k-1$ steps

Reachability Diameter

Every state reachable in 3 steps is also reachable in 2 steps →

Every reachable state is reachable in 2 steps

Reachability Diameter



Reachability Diameter

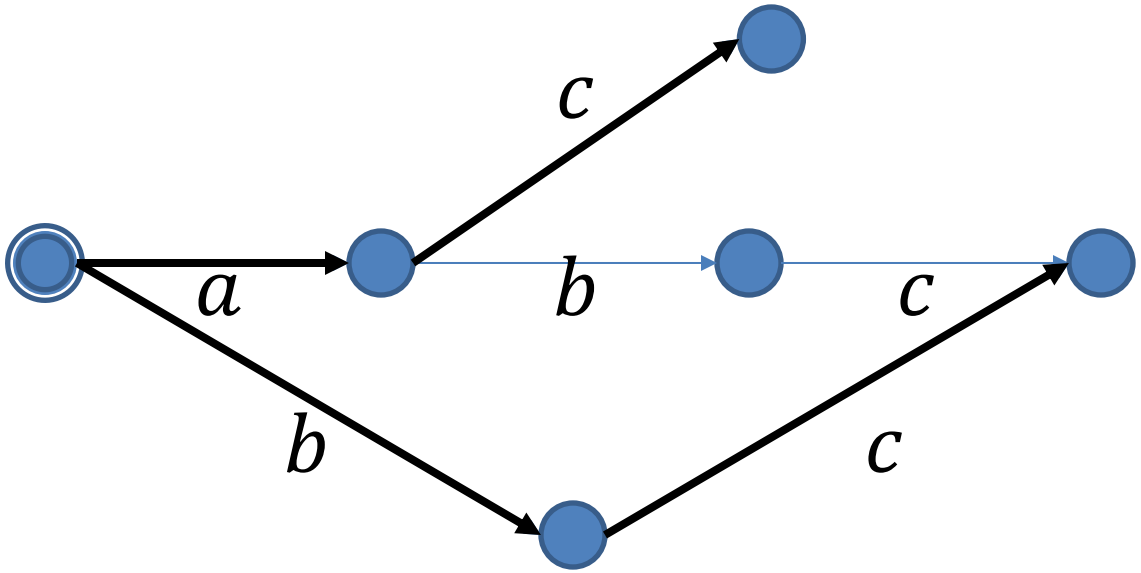
Every state reachable in 3 steps is also reachable in 2 steps →

Every reachable state is reachable in 2 steps

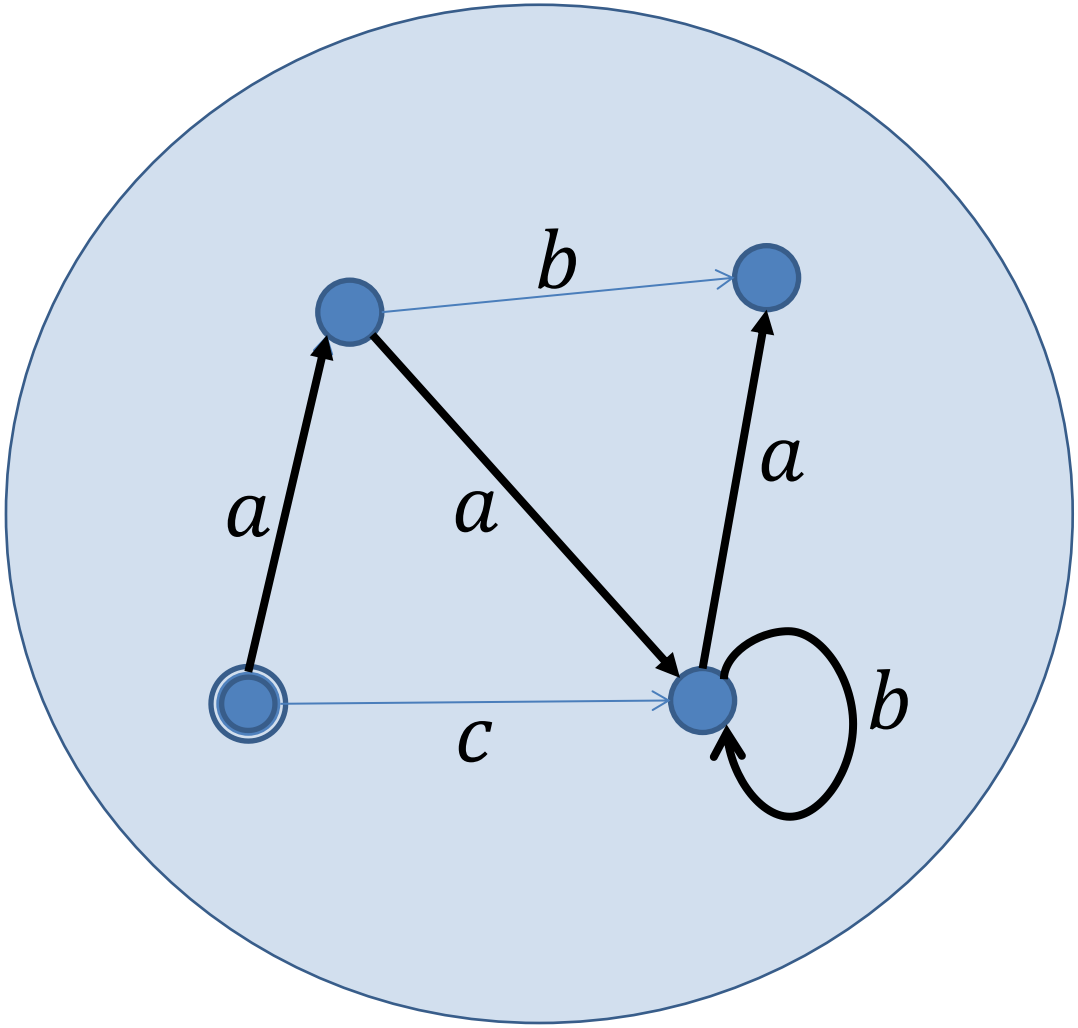
For all input traces of length 3,
there exists a 2-input trace with the same end state

Heuristic: Sub-sequence

a, c
b, c



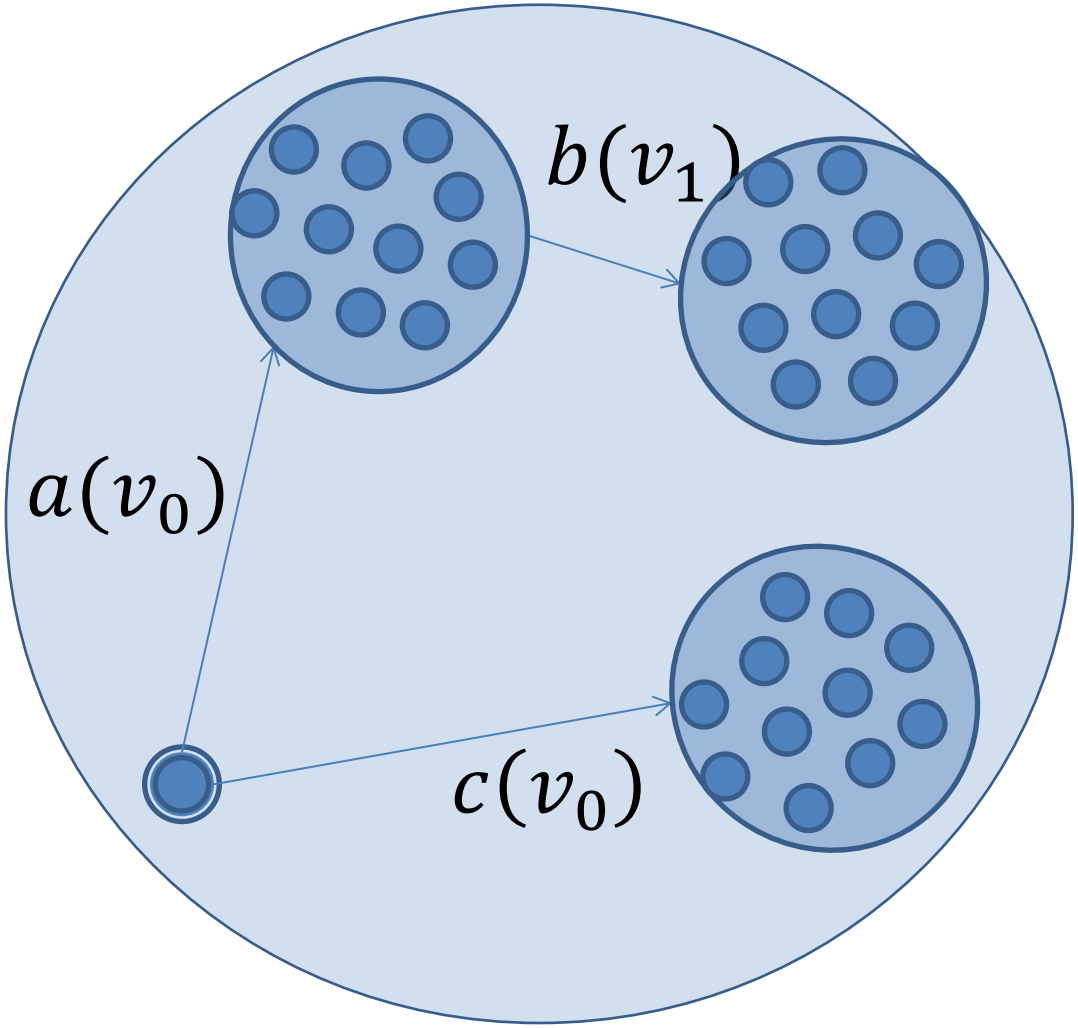
Heuristic: Gadgets



a, a, a
a, a, b

...

Heuristic: Gadgets



Evaluation

Is S^2W practical for real world applications?

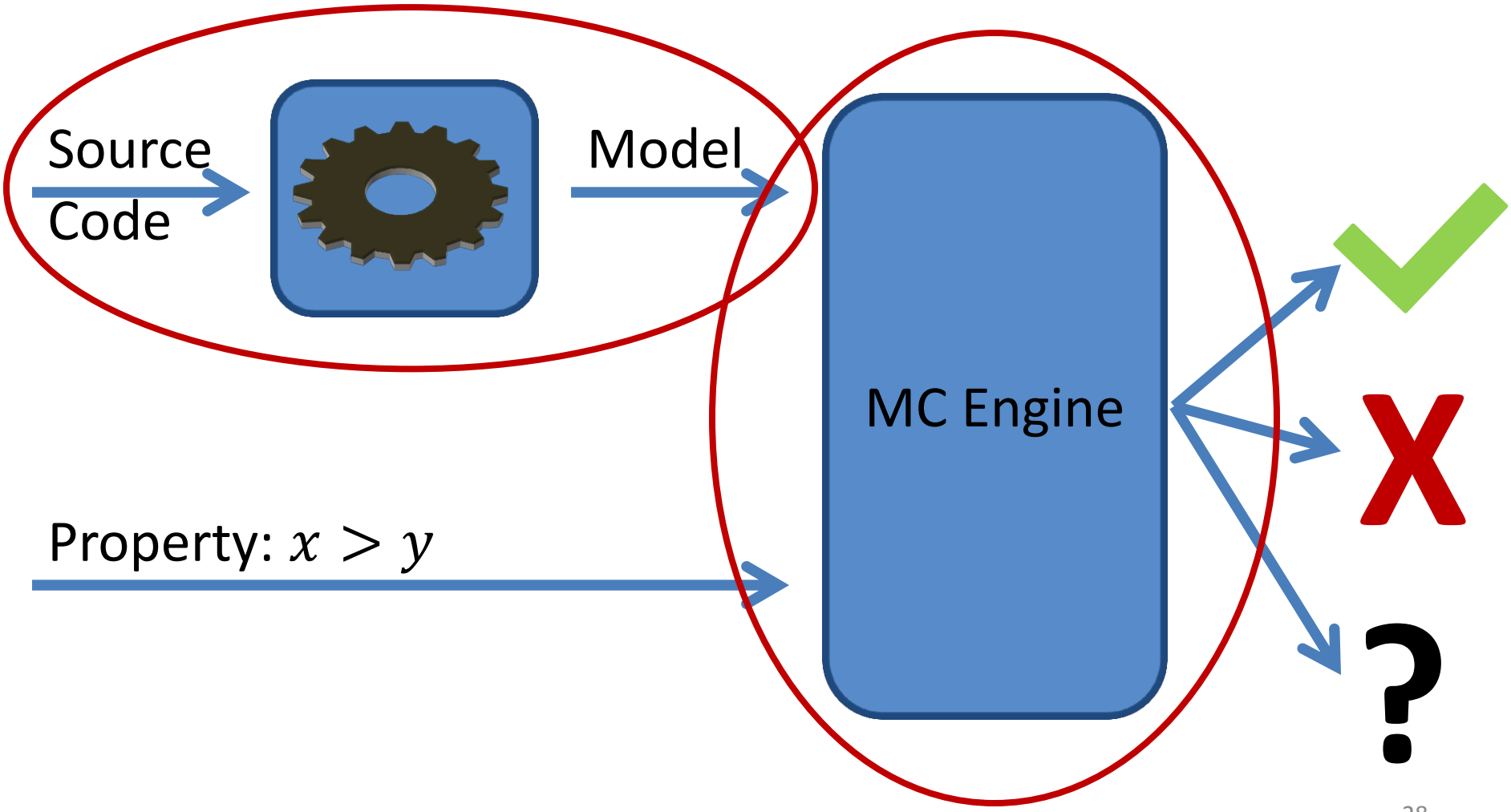
Case studies

- Bochs' address translation
  - Translation via TLB = translation via page tables
- Content Addressable Memory-based Cache
  - If present, cached data = data in memory
- Shadow Page Tables
  - Guest/host isolation

Case Studies

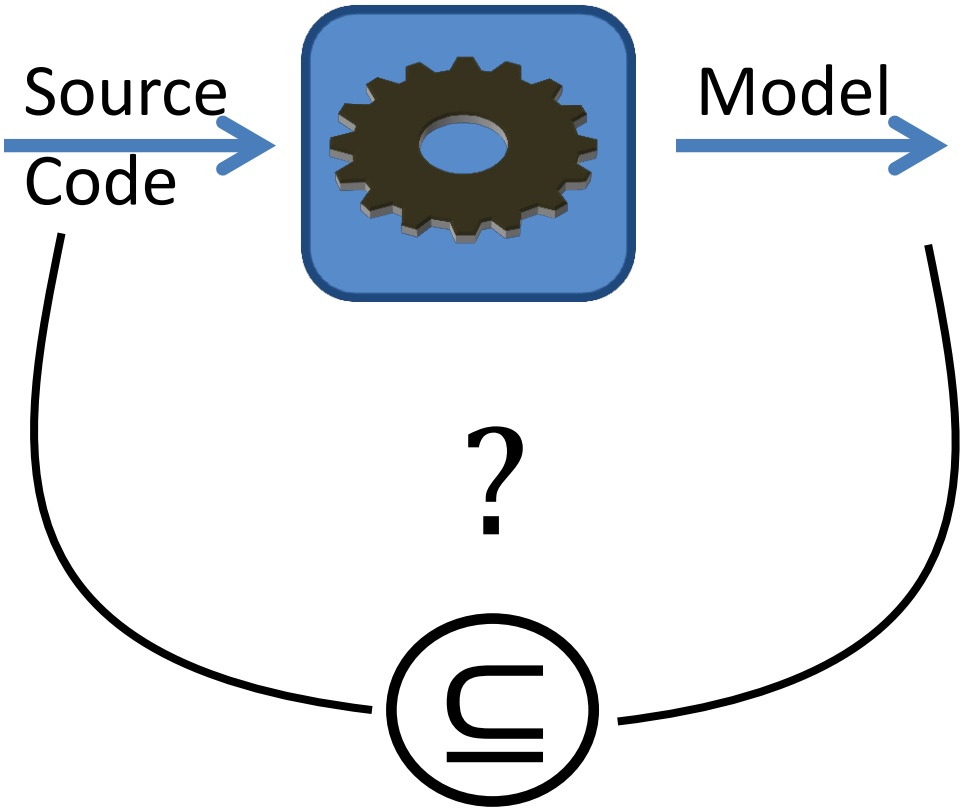
Case Study	Short World	S^2W	Model Checking
Bochs' TLB	8 steps	120 min	∞
CAM-based Cache	2 steps	2 sec	∞
Shadow Page Tables	4 steps	1 min	∞

Model Checking



Model Validation

Model Validation



Approach

1. Use symbolic execution to find paths through the code
2. For each path, note the path constraints and resulting output
3. Check validity of (path constraints, output) pair in the model

Code

```
if (curr_privilege_level == 3)
    page_fault = 1;
else
    page_fault = 0;
```

Model

```
page_fault :=
    (curr_privilege_level[0] &
     curr_privilege_level[1]);
```

Code

```
if (curr_privilege_level == 3)
    page_fault = 1;
else
    page_fault = 0;
```

(cpl = 3, pf = 1)

(cpl ≠ 3, pf = 0)

Model

```
page_fault :=
     (curr_privilege_level[0] &
      curr_privilege_level[1]);
```

valid? (curr_privilege_level == 3
→ page_fault = 1)

Model

```
page_fault :=
    (curr_privilege_level[0] &
     curr_privilege_level[1]);
```

valid? (curr_privilege_level ≠ 3 → page_fault = 0)

Counter-example: curr_privilege_level = 7, page_fault = 1

Evaluation

Is model validation effective?

Case Study: Bochs Address Translation Function

- Small function
  - 98 LOC
- But, interesting
  - 219 paths explored

  (KLEE symbolic execution engine)
- A well-studied model
  - Used in our S^2W work

➢ Model validation found 5 bugs

Case Study: Traffic Collision Avoidance System

- Safety-critical software
- Implementation from SW Engineering literature
  - 173 LOC
  - 9 procedures
- 23 versions, each with a different injected fault

➢ Model validation found all injected bugs

Conclusion

- Practical verification of security properties of virtualization software
- Model validation strengthens results of verification

➢ Substantial inroads toward verifying legacy virtualization software

Thank you