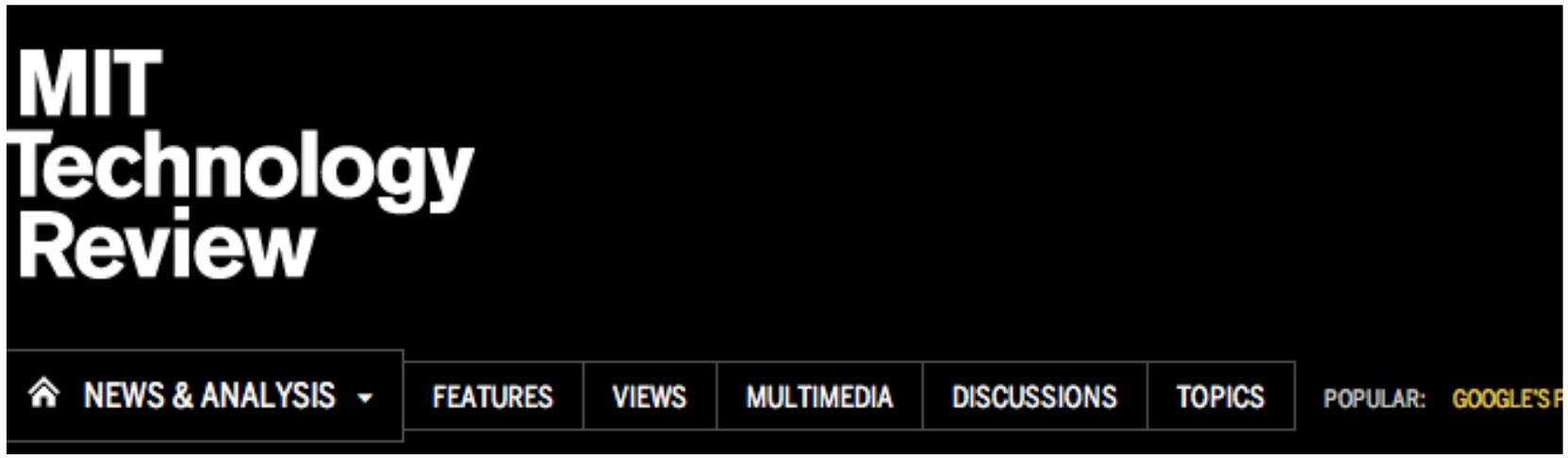
From SMT Solvers to Verifiers

Aws Albarghouthi
University of Toronto

Joint work with Arie Gurfinkel (CMU) & Marsha Chechik (UToronto)

May 7th 2013
@ HCSS

Software Failure...

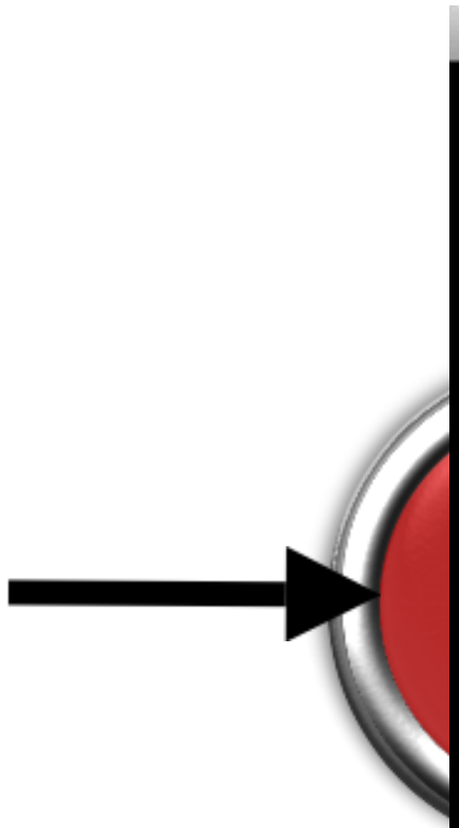


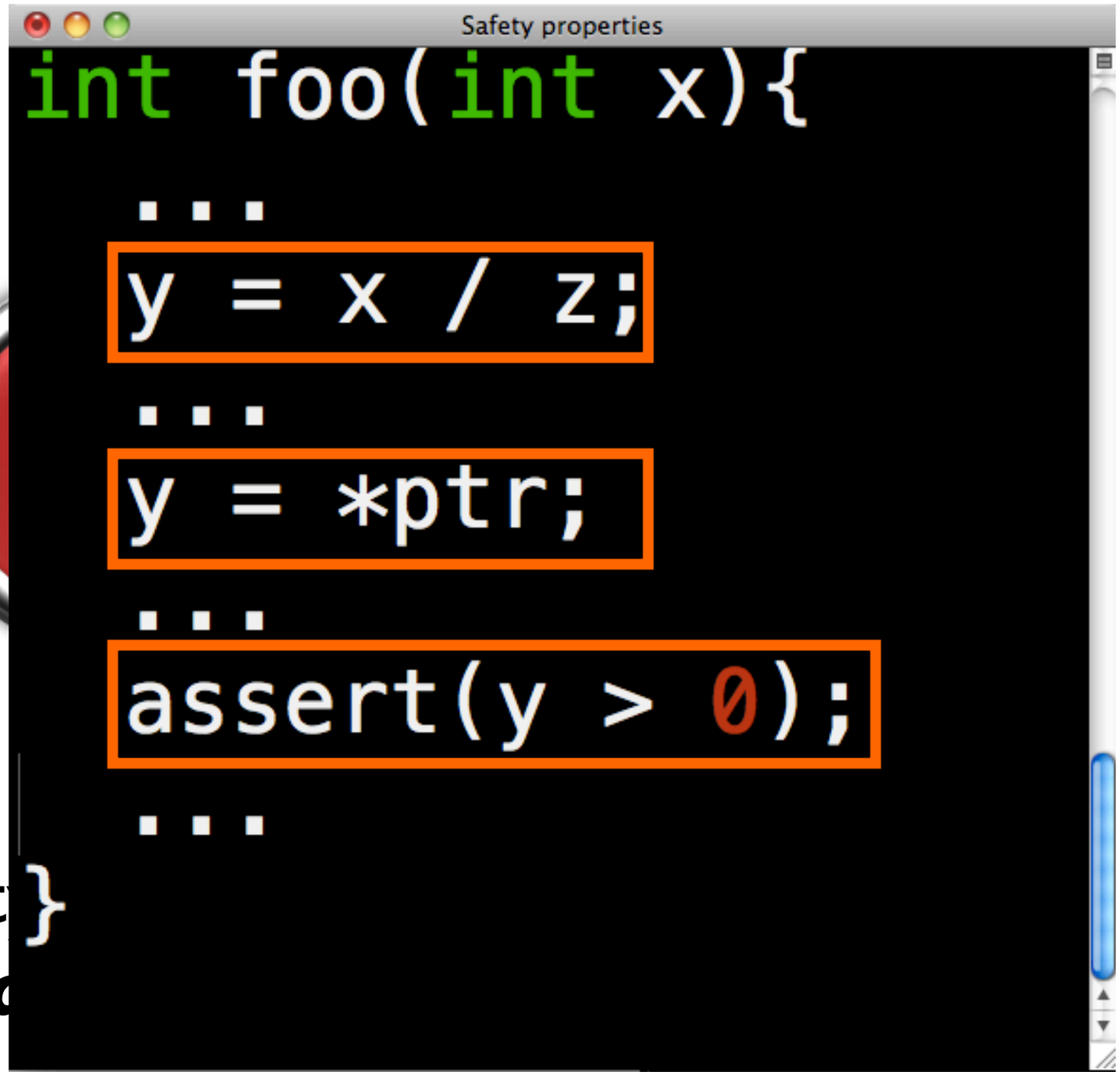
Will Knight
March 11, 2013

Buckle up for the Vehicular Zombie Apocalypse

Automated Verification

Source Program code

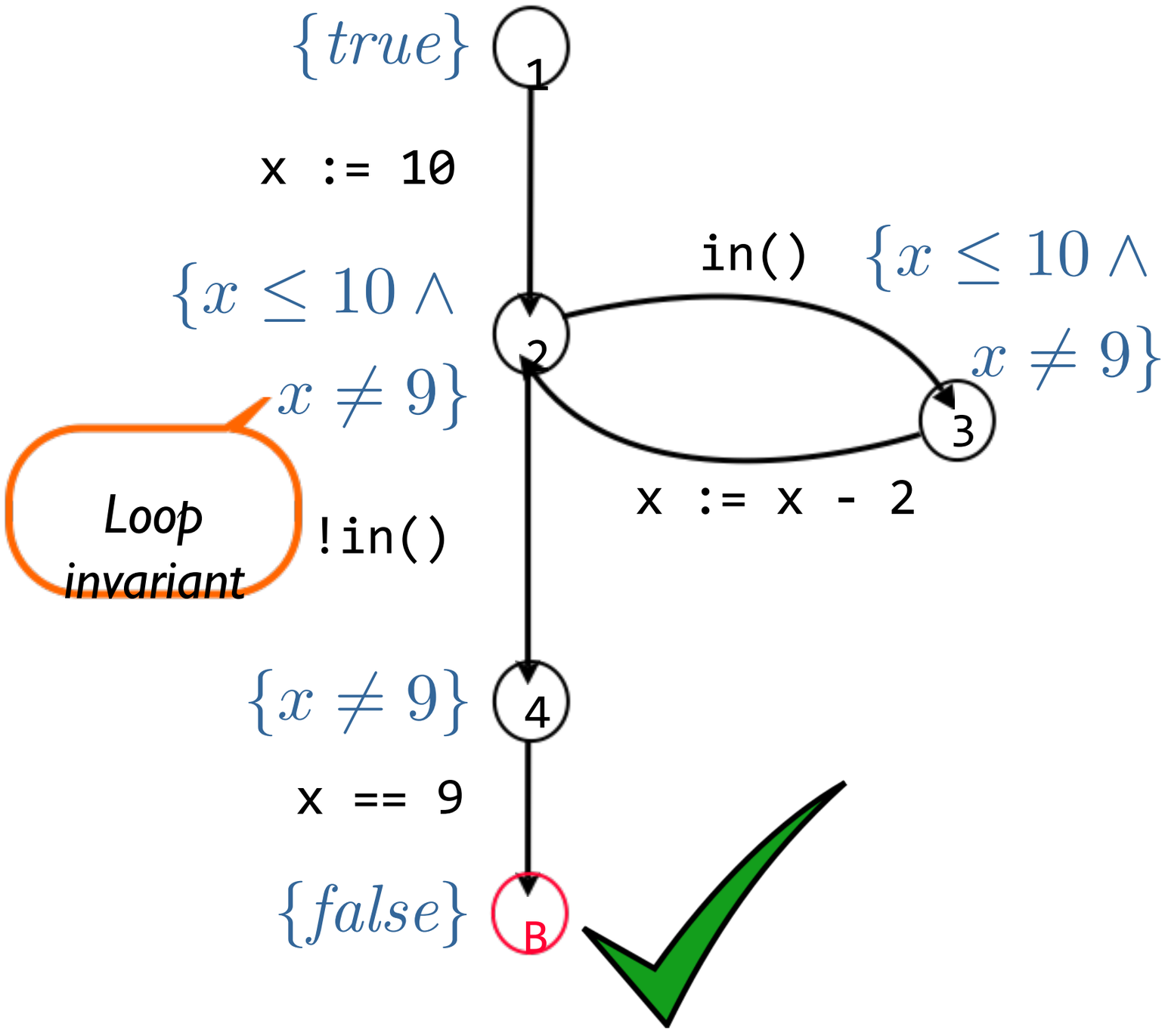




Safet

"Something ***bo***

Verification Example



$$true \Rightarrow \mathbf{I_1}$$

$$\mathbf{I_1} \wedge x' = 10 \Rightarrow \mathbf{I_2}$$

$$\mathbf{I_2} \wedge in \Rightarrow \mathbf{I_3}$$

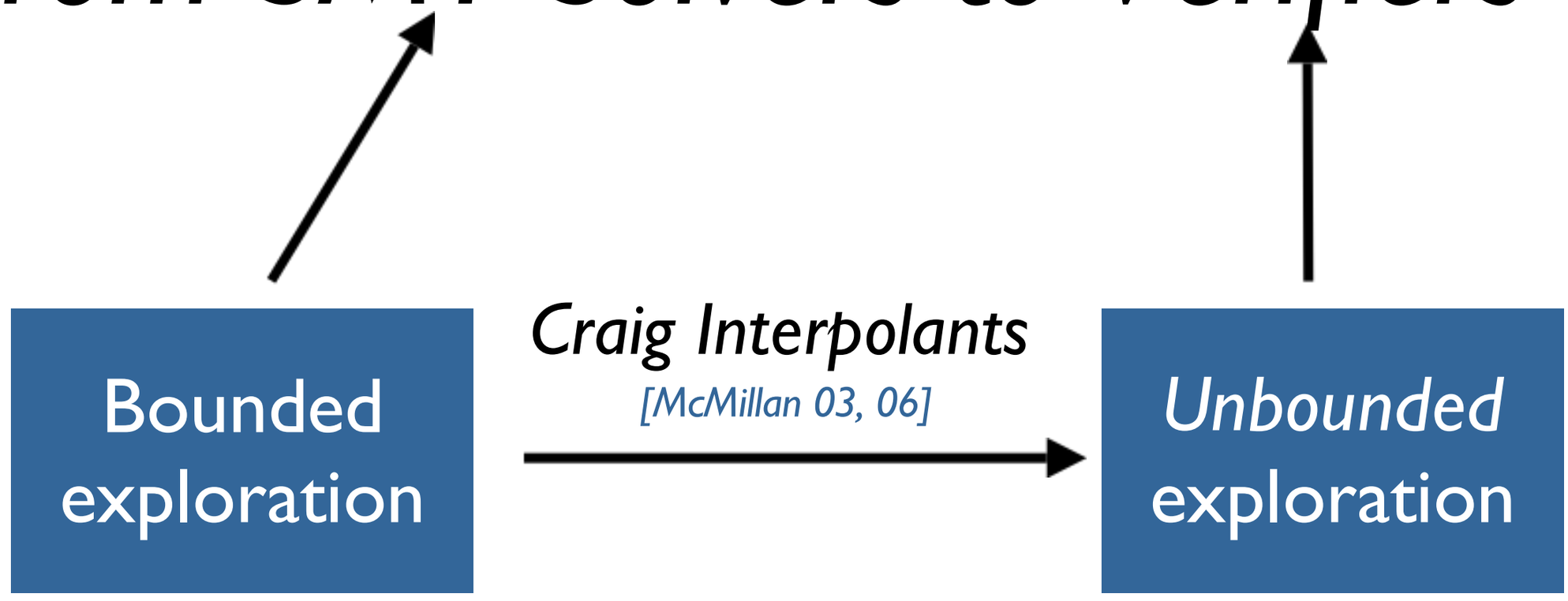
$$\mathbf{I_2} \wedge \neg in \Rightarrow \mathbf{I_4}$$

$$\mathbf{I_3} \wedge x' = x - 2 \Rightarrow \mathbf{I_2}$$

$$\mathbf{I_4} \wedge x = 9 \Rightarrow \mathbf{I_B}$$

$$\mathbf{I_B} \Rightarrow false$$

From SMT Solvers to Verifiers



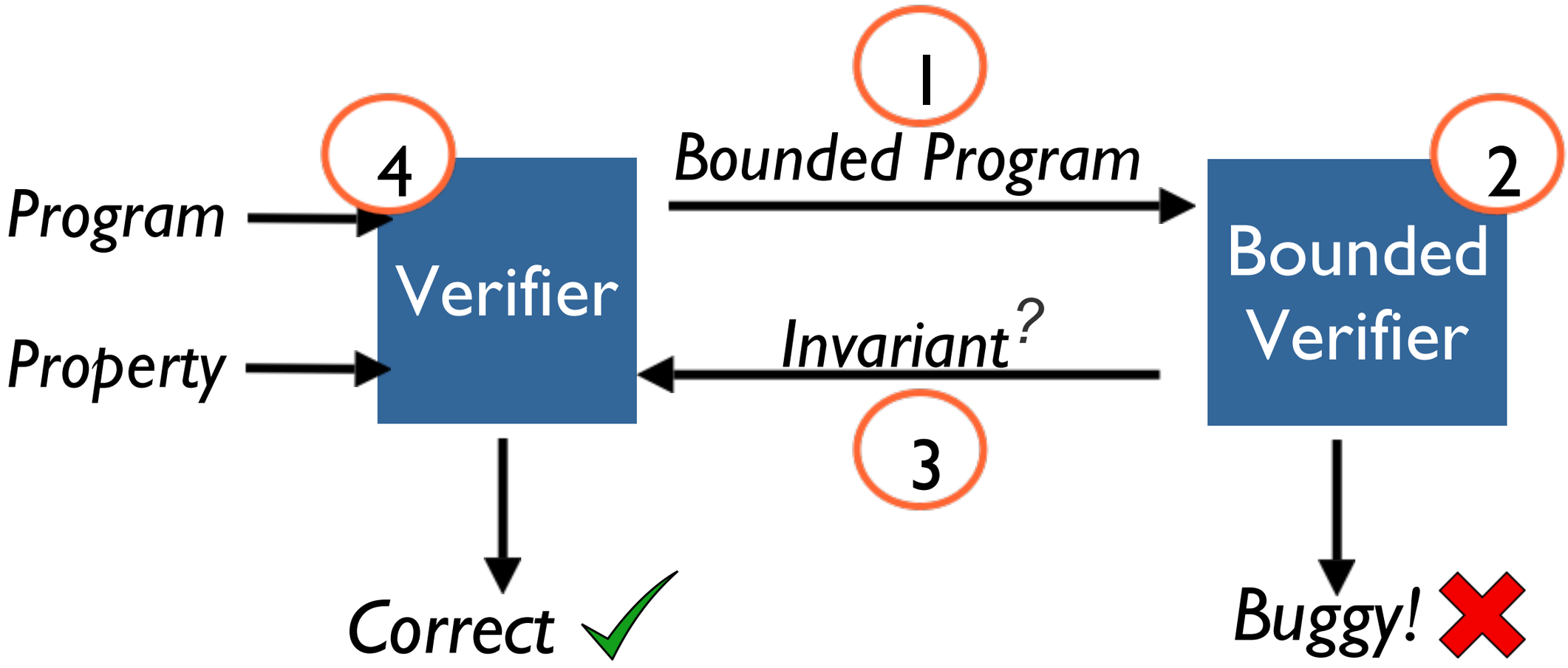
Today's Menu

The UFO approach

UFO  practice

Future goals

The UFO Approach



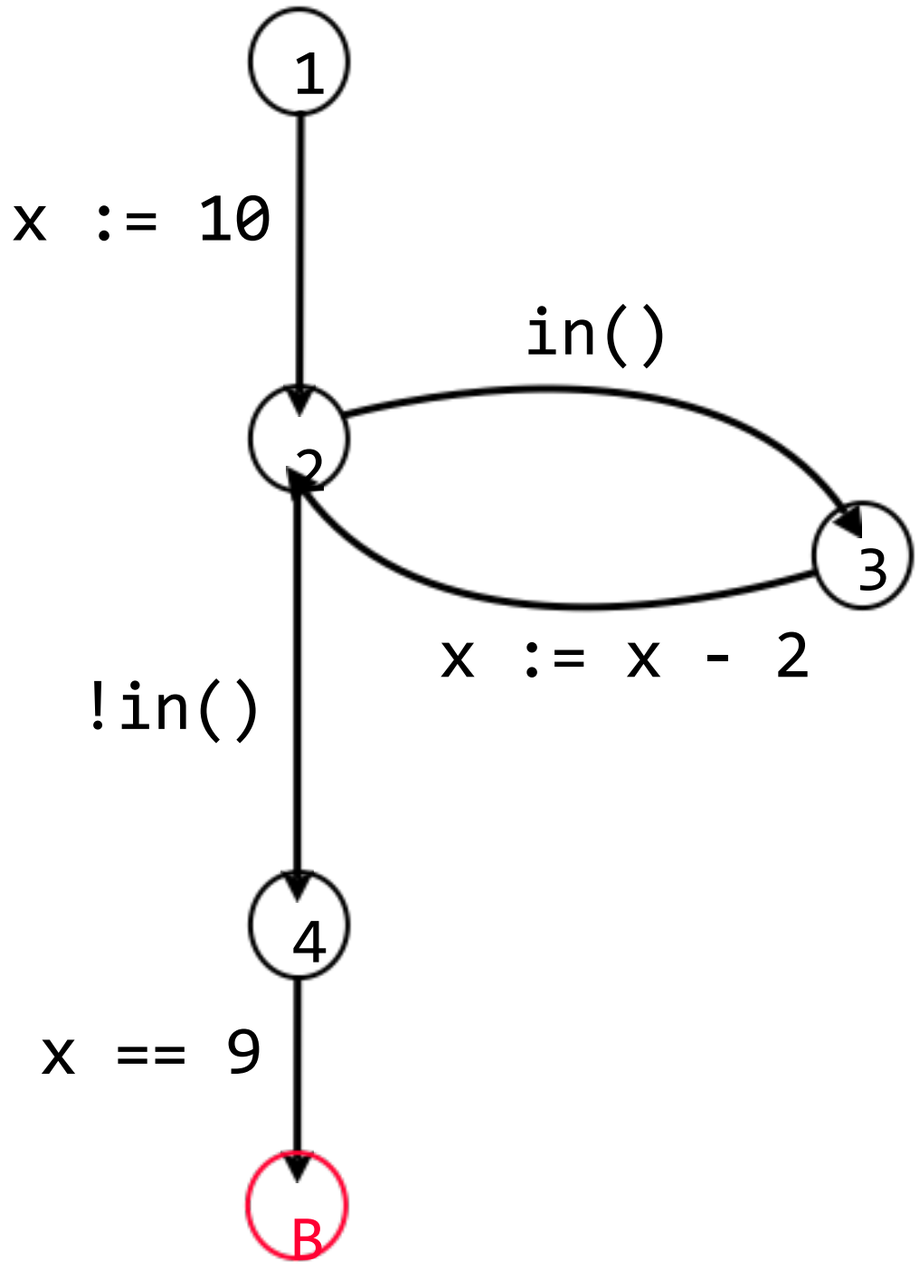
1: Unroll program loops

2: Verify bounded program

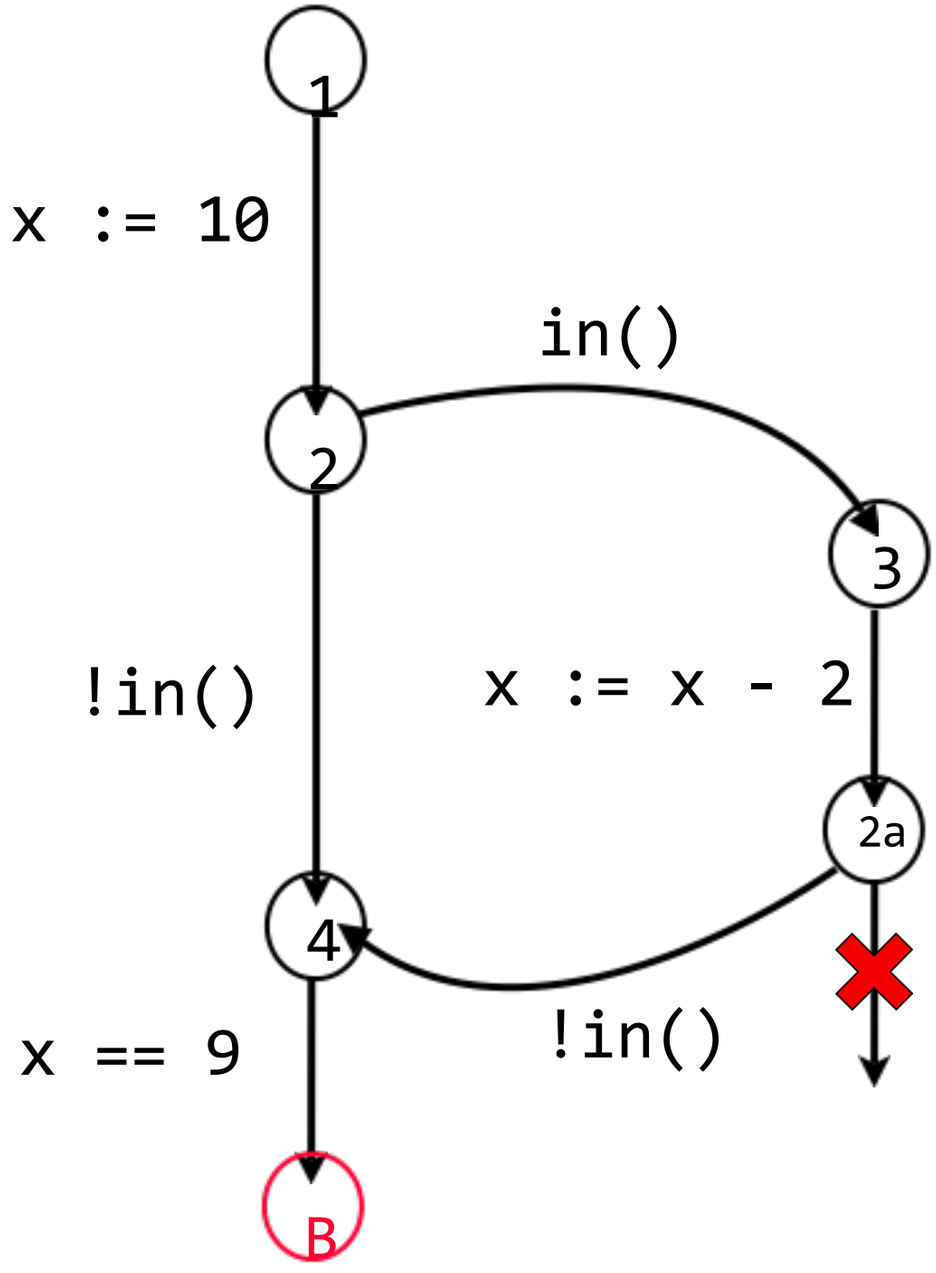
3: Compute *invariant*? using *interpolants*

4: Check *invariant*?

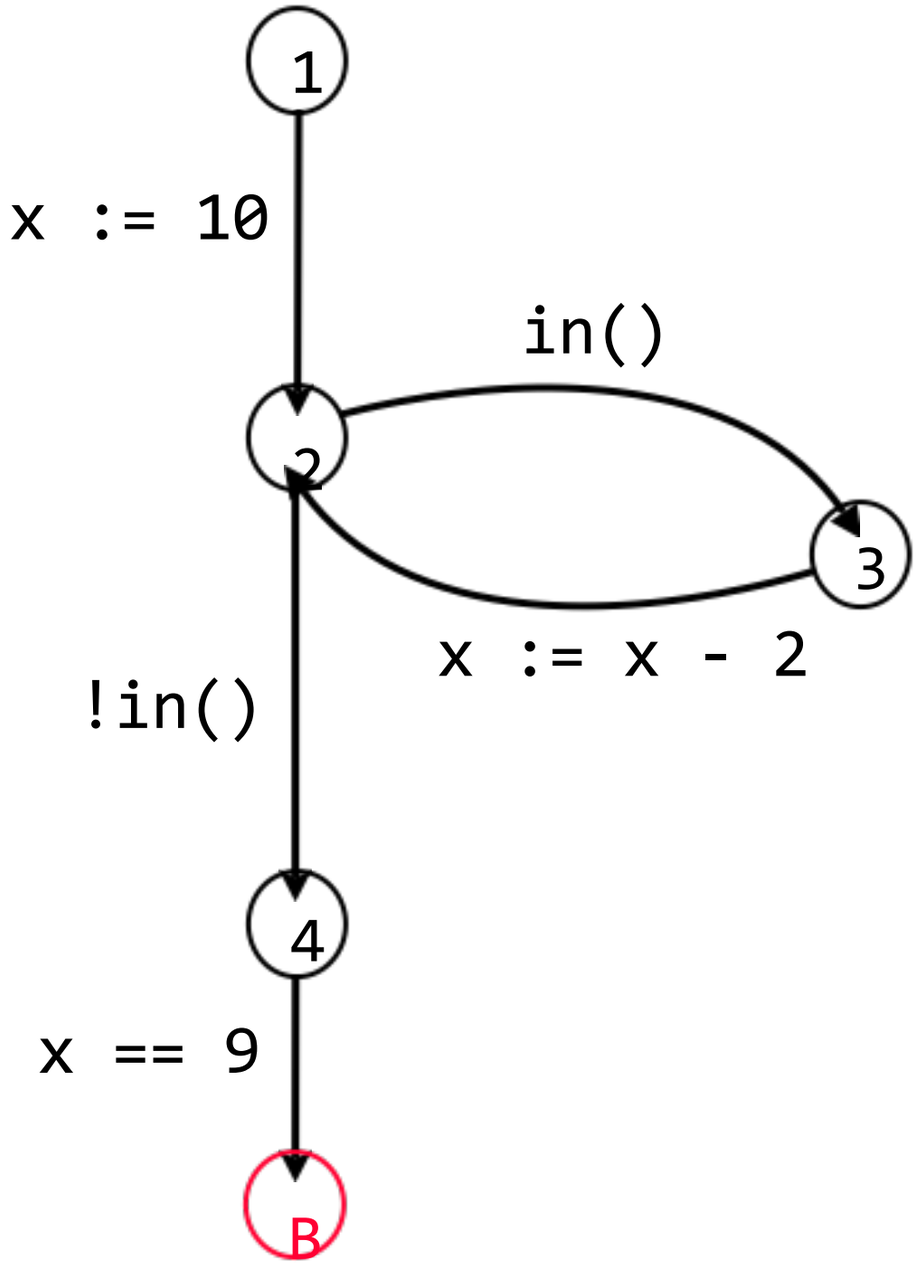
1: Bounded Program



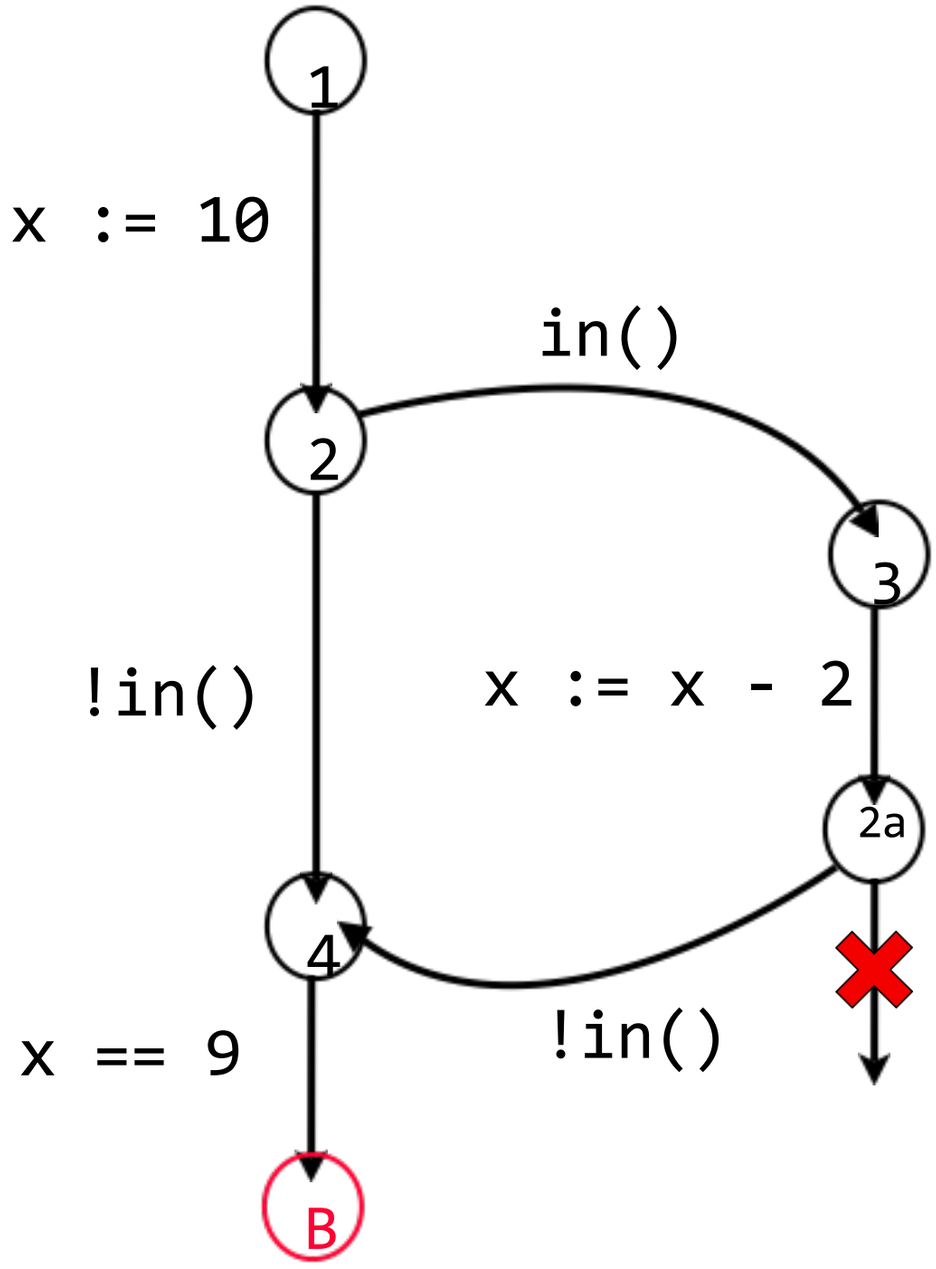
One loop unrolling



1: Bounded Program



One loop unrolling



2: Bounded Verification

Program Encoding

$$L_1$$

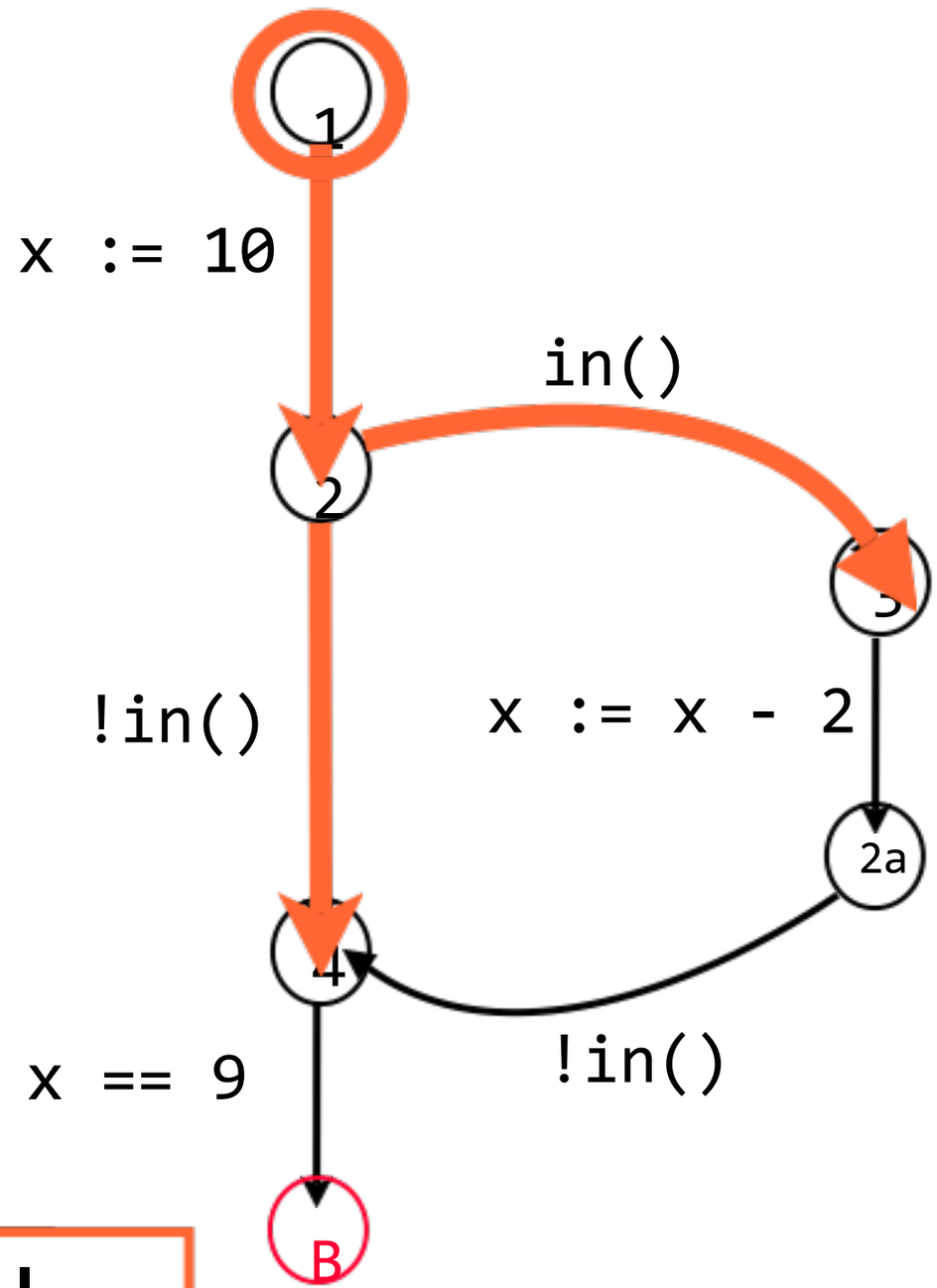
$$L_1 \Rightarrow x_1 = 10 \wedge L_2$$

$$L_2 \Rightarrow (in_1 \wedge L_3)$$
$$\vee\, (\neg in_1 \wedge L_4)$$

$$L_3 \Rightarrow \cdots$$

$$L_{2a} \Rightarrow \cdots$$

$$L_4 \Rightarrow \cdots$$



Use SMT solver

3: Compute *Invariant*?

Program Encoding

$$L_1$$

$$L_1 \Rightarrow x_1 = 10 \wedge L_2$$

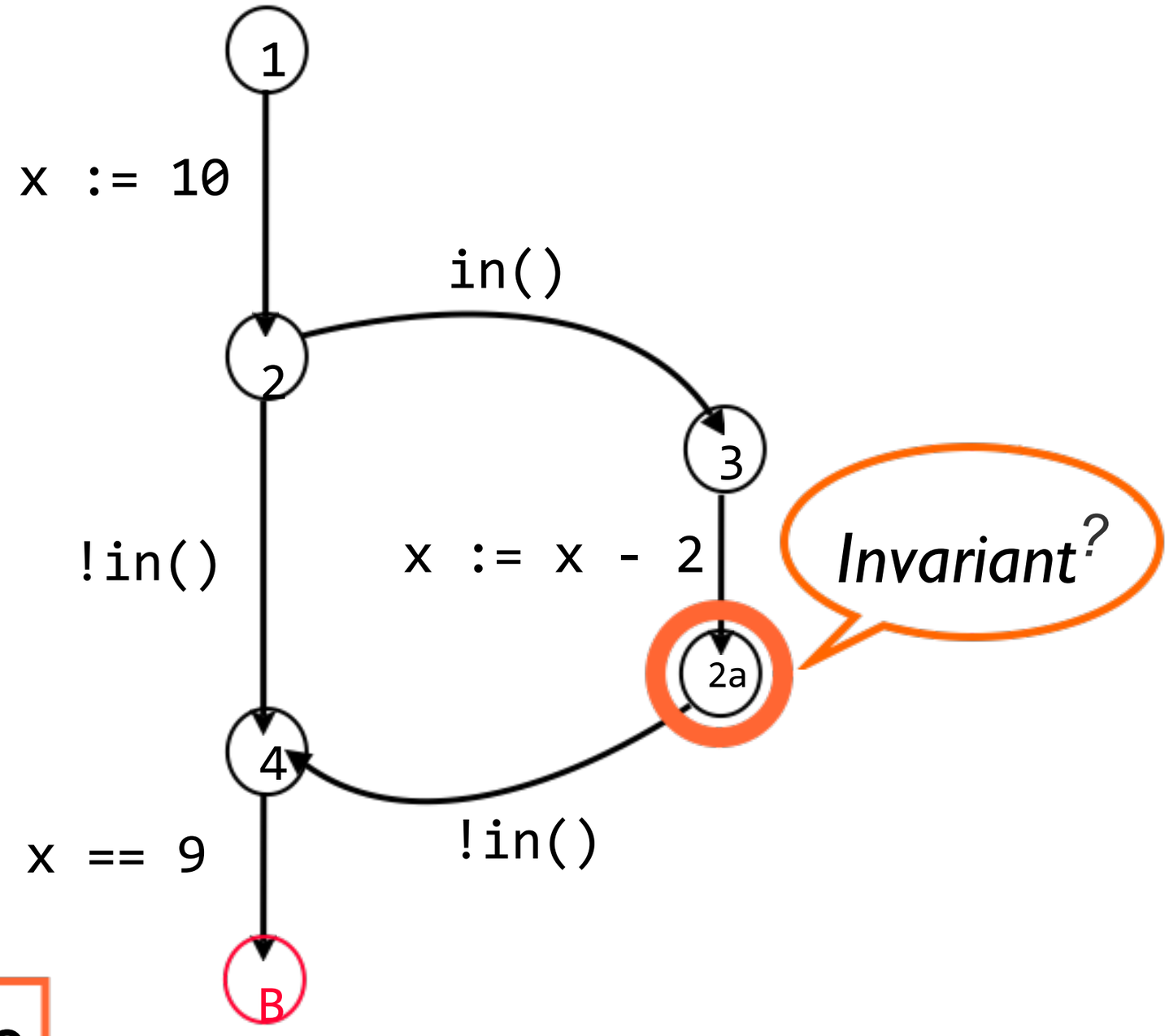
$$L_2 \Rightarrow (in_1 \wedge L_3)$$
$$\vee (\neg in_1 \wedge L_4)$$

$$L_3 \Rightarrow \cdots$$

$$L_{2a} \Rightarrow \cdots$$

$$L_4 \Rightarrow \cdots$$

Unsatisfiable



3: Compute *Invariant*?

Program Encoding

L_1

$L_1 \Rightarrow x_1 = 10 \wedge L_2$

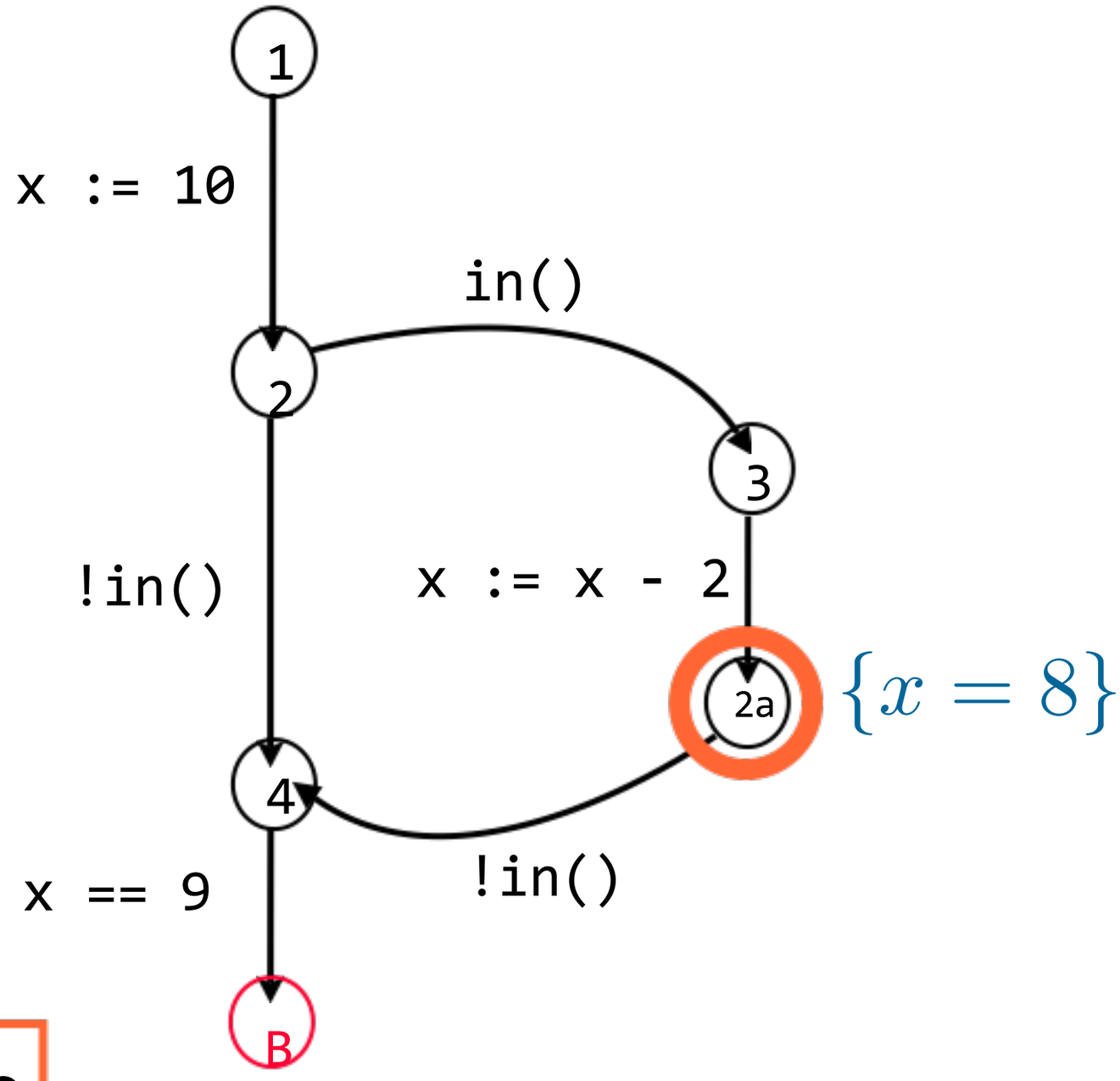
$L_2 \Rightarrow (in_1 \wedge L_3)$
$\quad \vee (\neg in_1 \wedge L_4)$

$L_3 \Rightarrow \cdots$

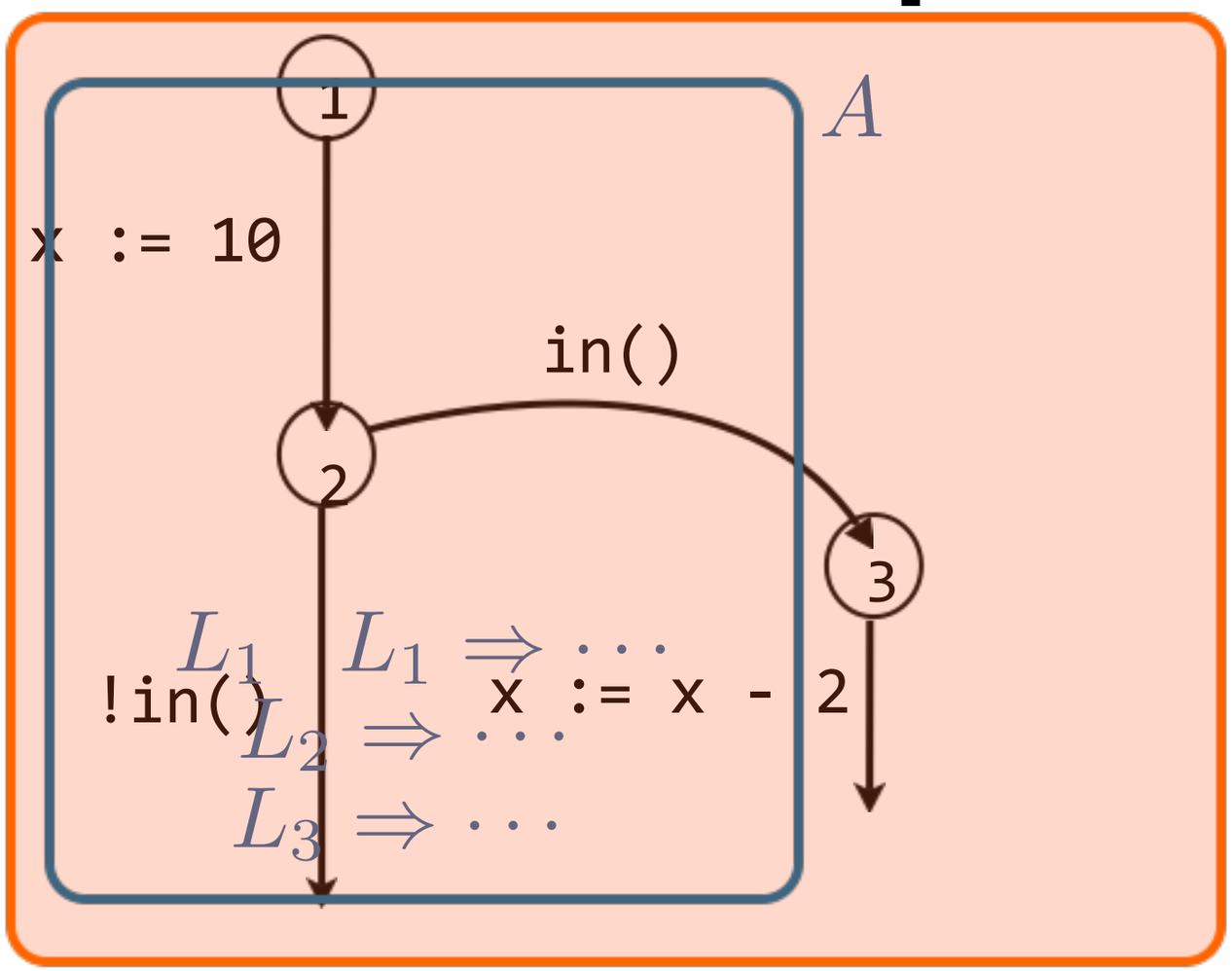
$L_{2a} \Rightarrow \cdots$

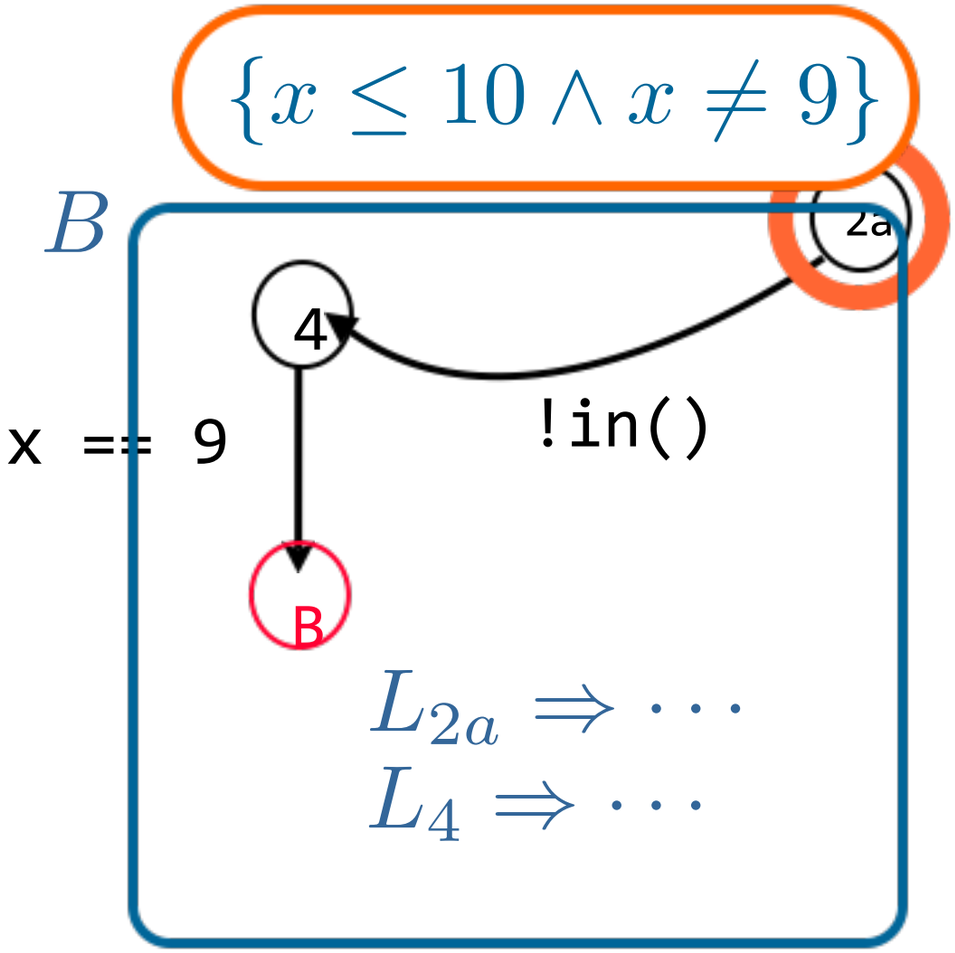
$L_4 \Rightarrow \cdots$

Unsatisfiable



3: Compute *Invariant*?

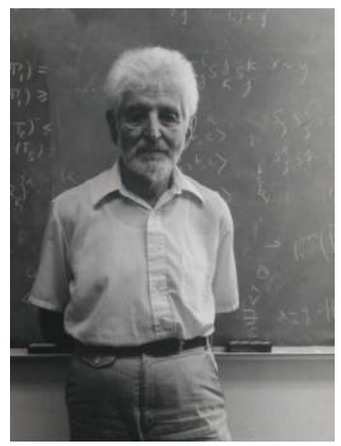




$$A \wedge B \equiv false$$

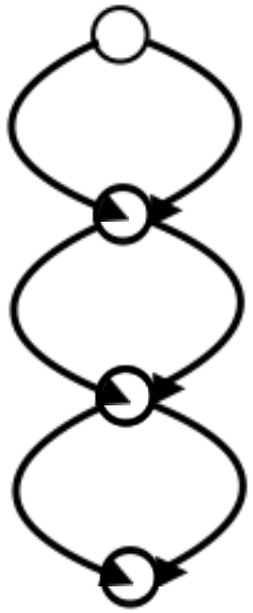
$$A \Rightarrow I \Rightarrow \neg B$$

Craig's interpolation theorem



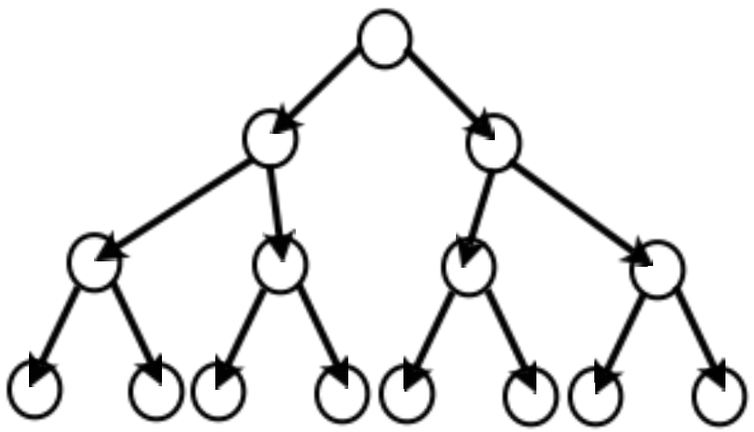
[Craig 57]

3: Compute *Invariant*?

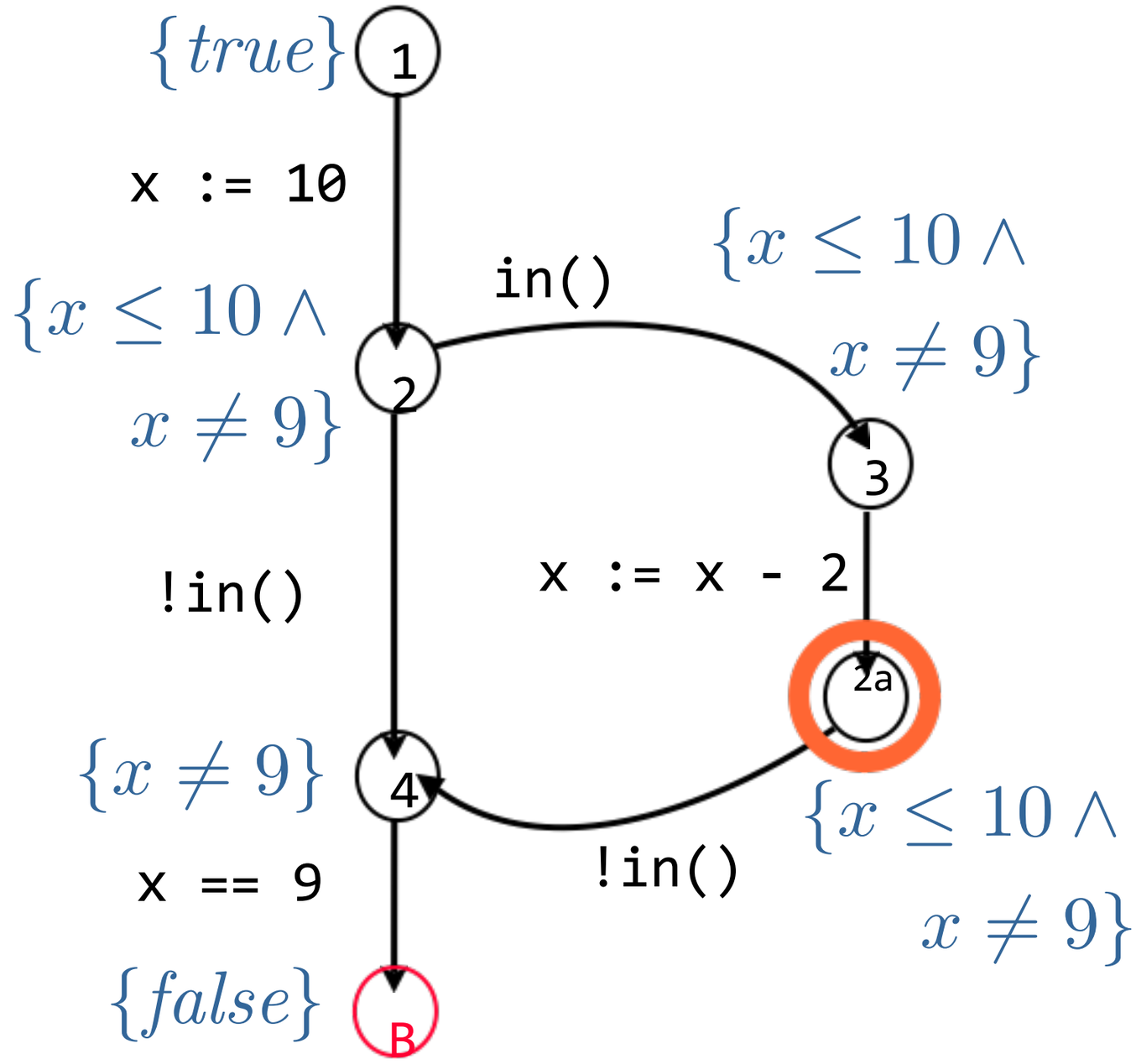


e.g., *[McMillan 06, Henzinger et al. 02]*

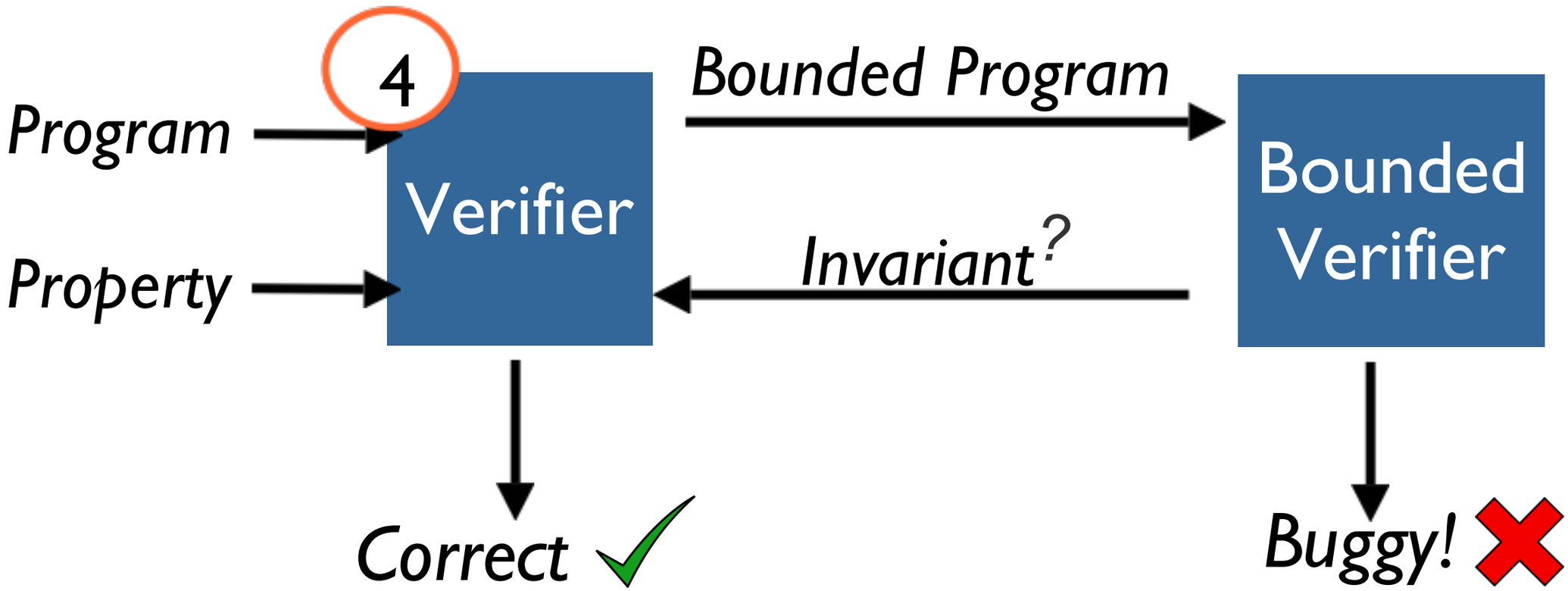
2^n *paths!*



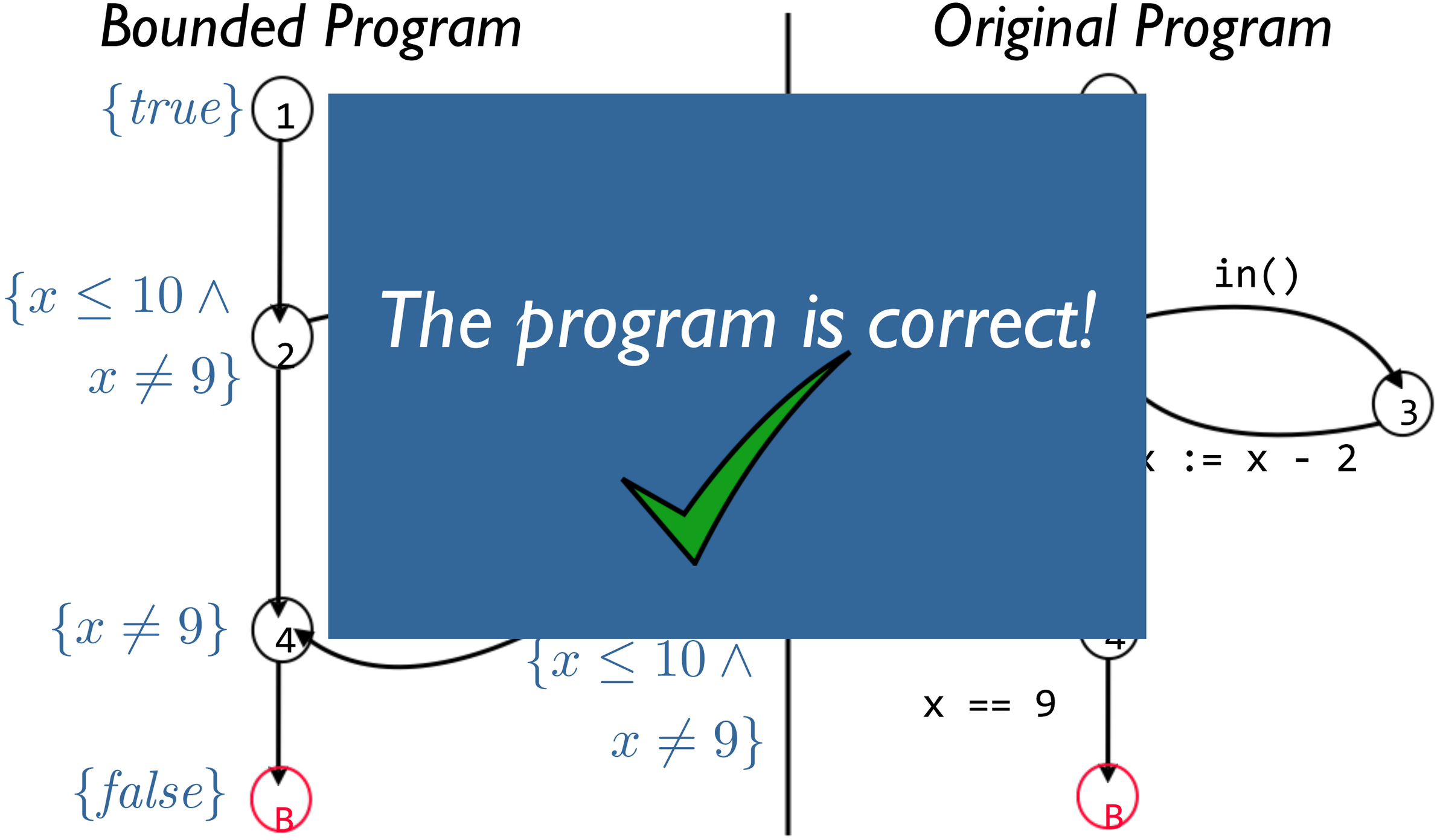
DAG Interpolants



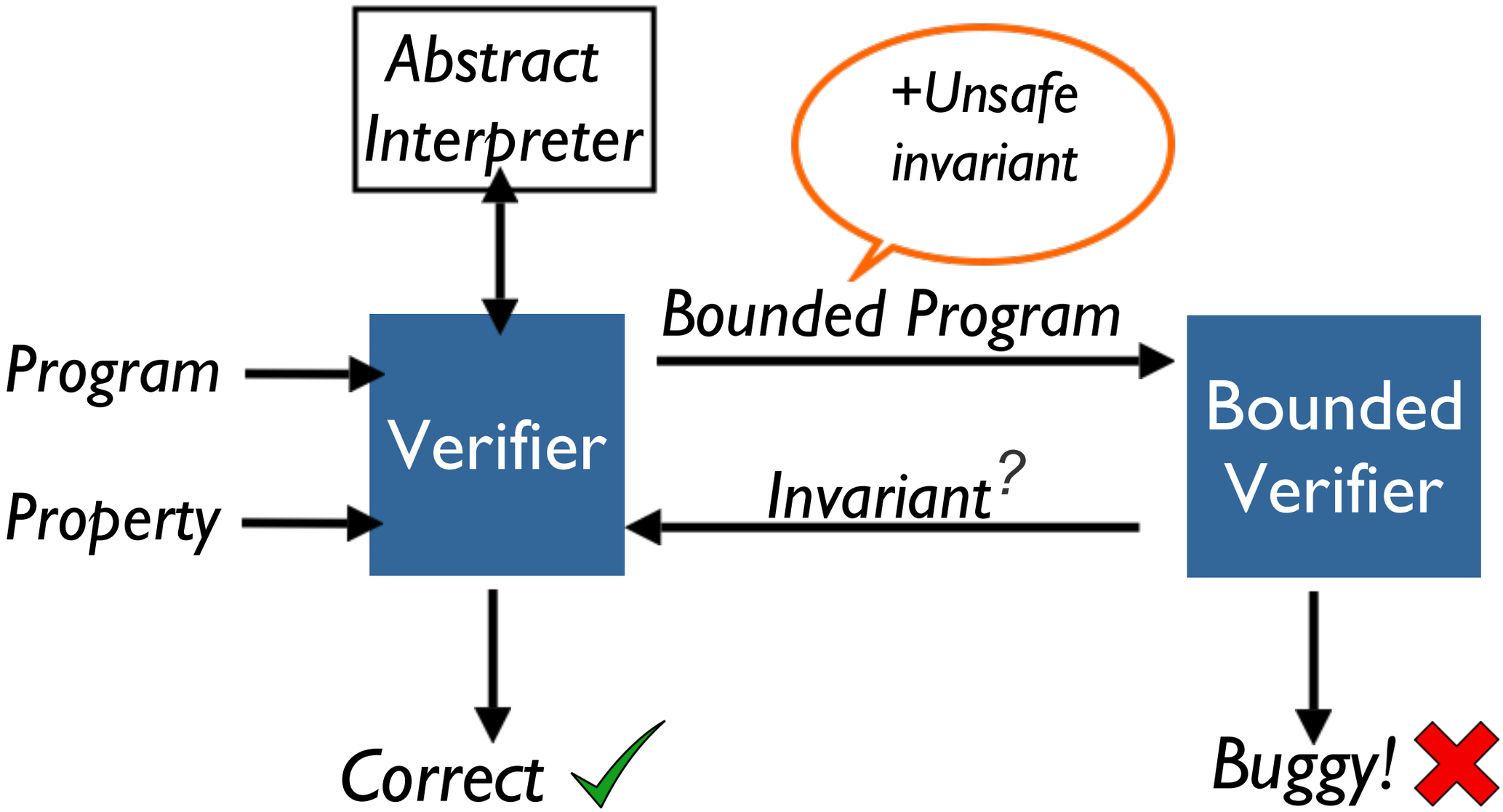
4: Check *Invariant*?



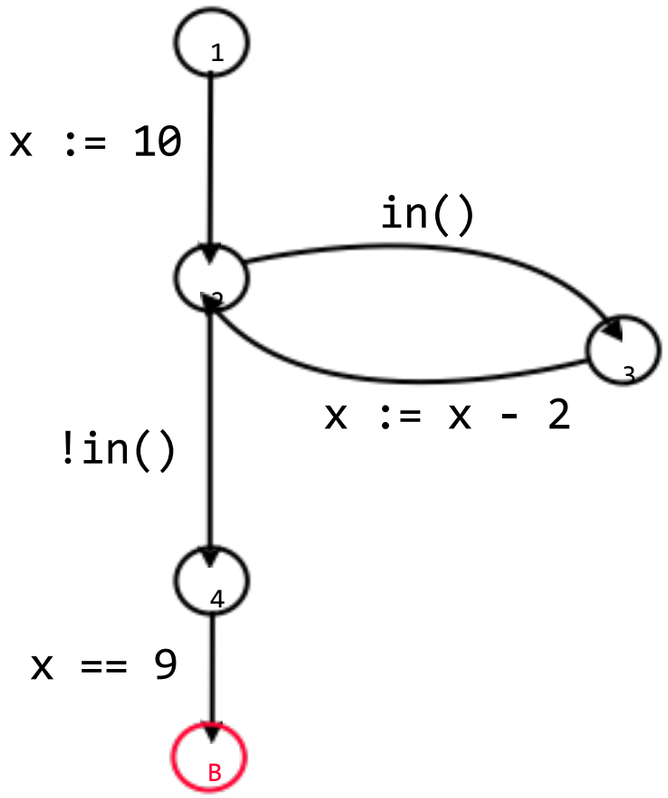
4: Check *Invariant*?



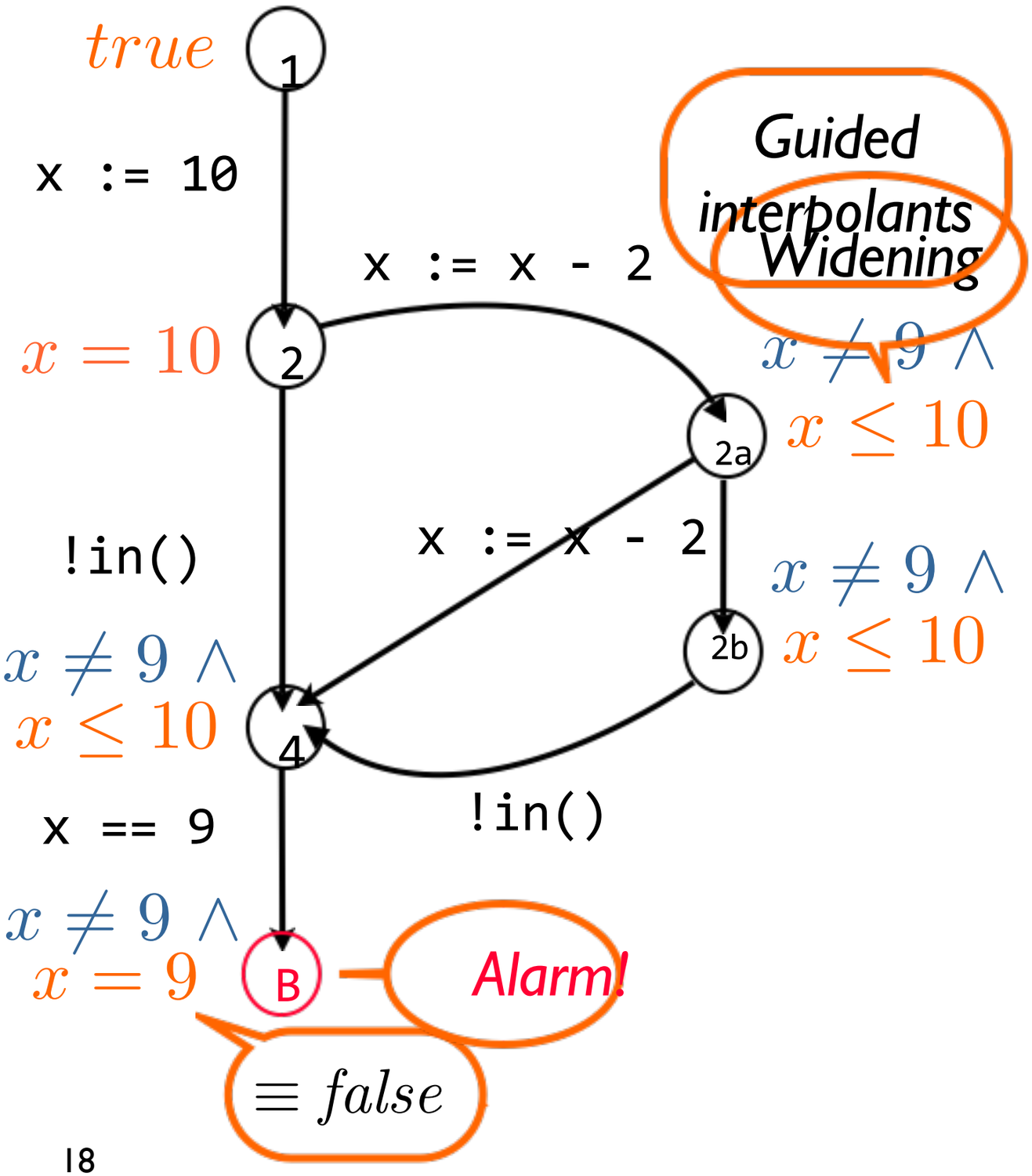
The UFO Approach



Guided Interpolants

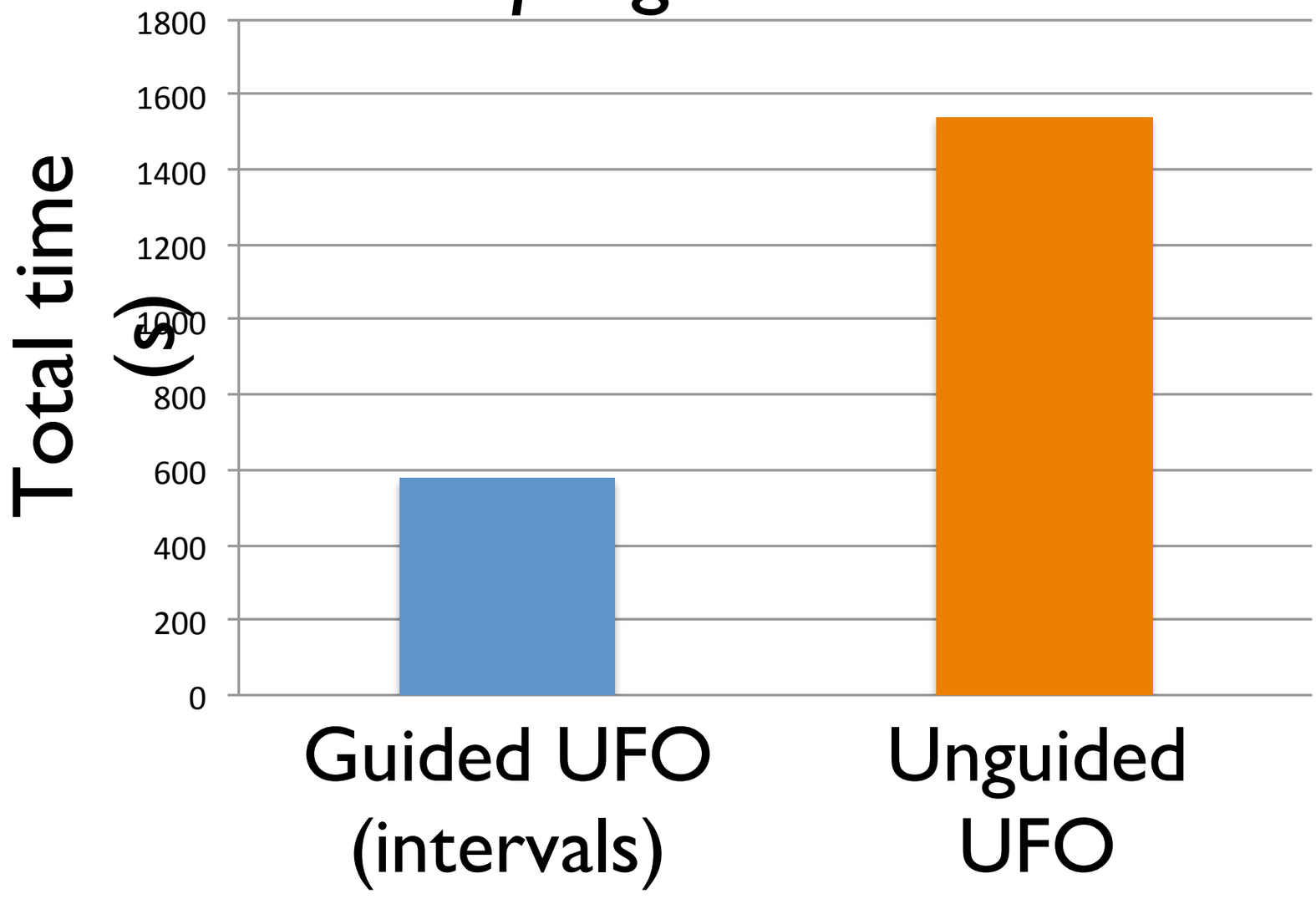


- Intervals domain
- Weak topological ordering
- Powerset widening
- AI dictates unrolling



Guided Interpolants

Guided vs. unguided UFO on 93 C programs



Contributions of UFO

Utilizes SMT solvers for path enumeration (*and DAG interpolation*)

Guided interpolation with weak abstractions

General techniques: independent of logical theory

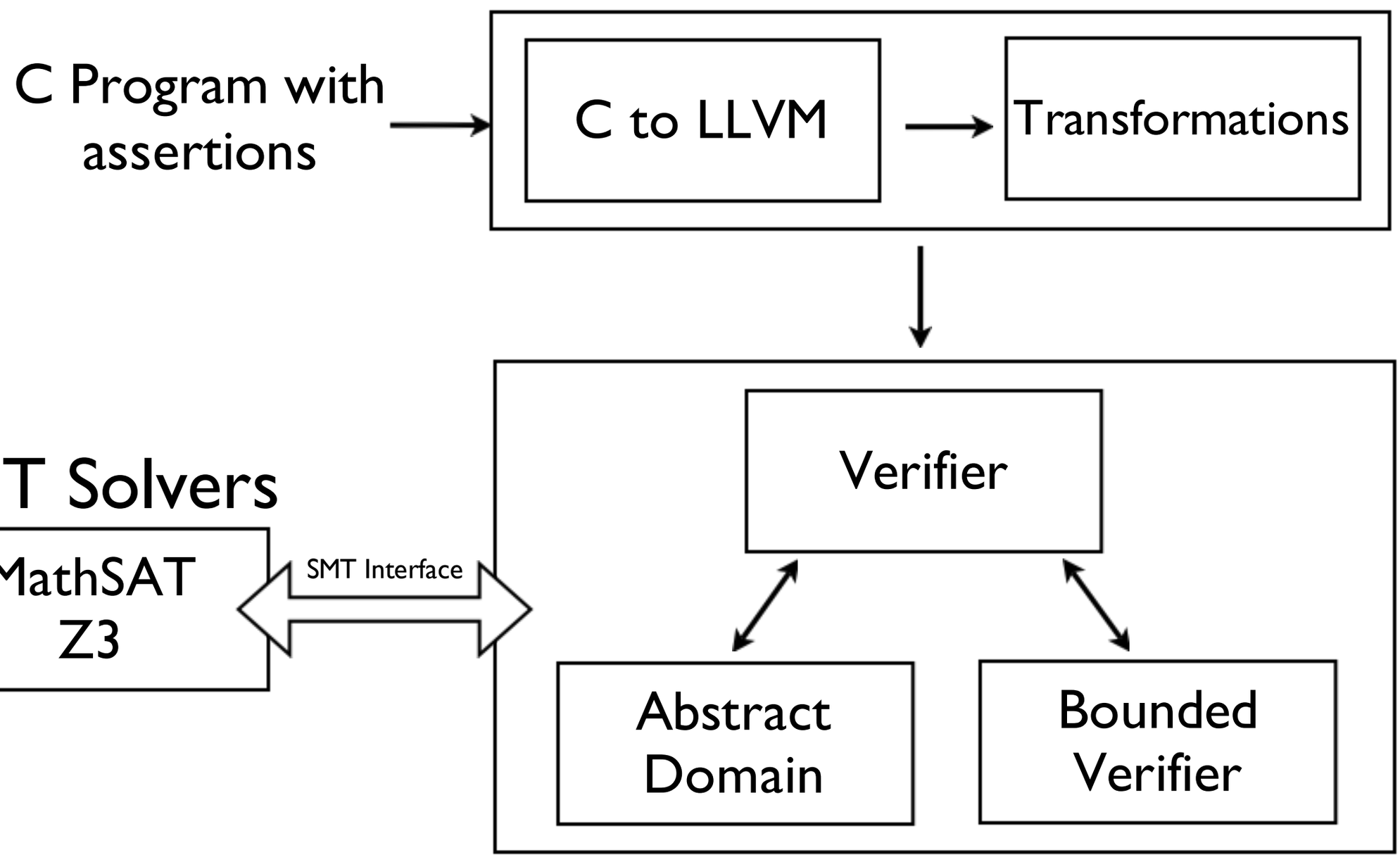
Today's Menu

The UFO approach

UFO  in practice

Future goals

UFO Framework



UFO in SVCOMP

2nd International Software Verification Competition, TACAS 2013

Lazy abstraction
[Henzinger et al. 02]

Impact
[McMillan 06]

Benchmarks

2000+ C programs
2 to 180 KLoC

Linux/Windows drivers, SystemC, SSH, Product Lines, etc.

CPAchecker, *BLAST, CSeq, ESBMC, LLBMC, Predator, Symbiotic, Threader,* **Ultimate**

Nested interpolants
[Heizmann et al. 10]

UFO in SVCOMP

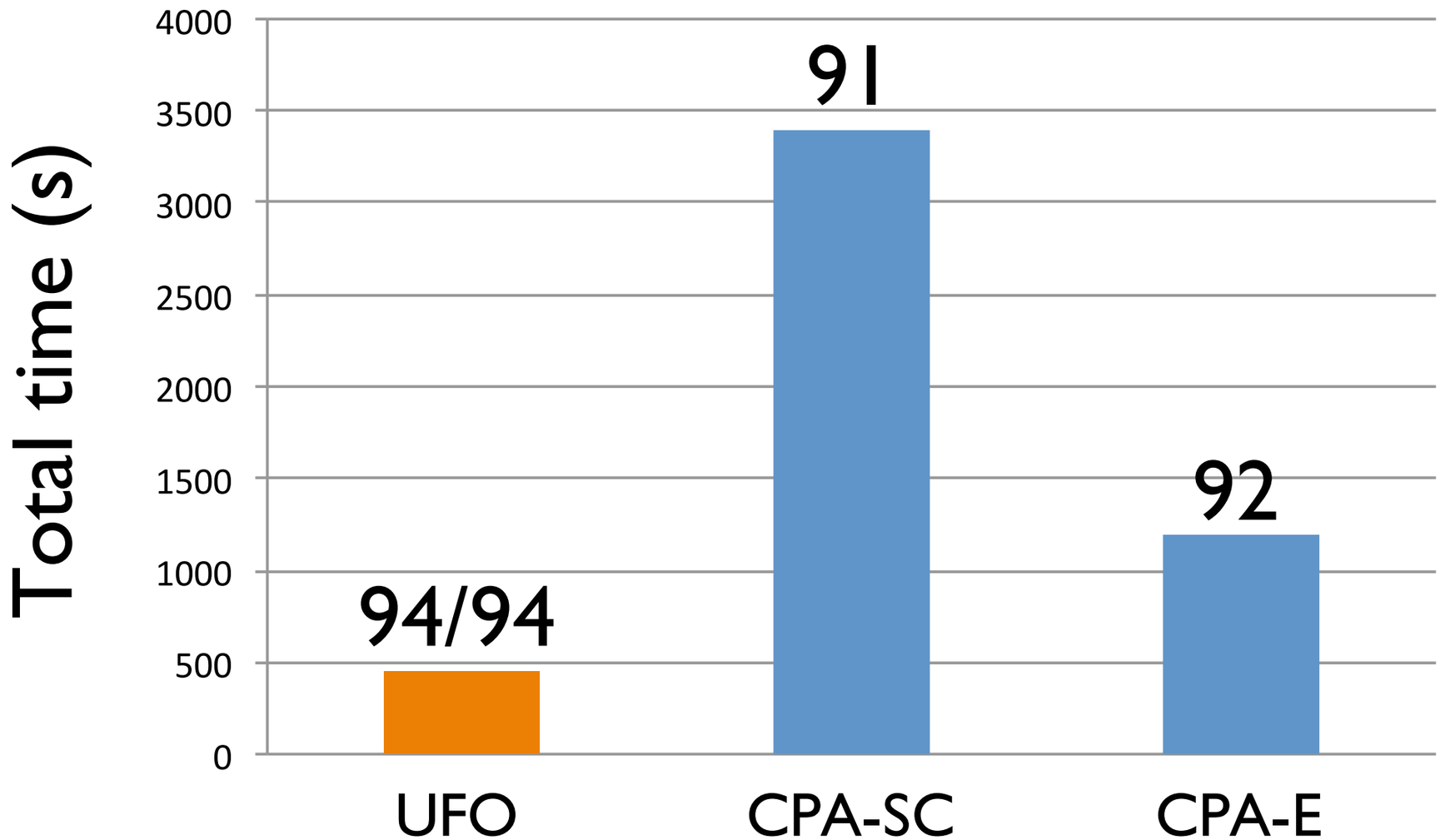
UFO won 4 out of 10 categories:

- *Control-flow integer*
- *Linux device drivers*
- *Software product lines*
- *SystemC*

UFO in SVCOMP

Control-flow Integer category results

(94 programs)



Future Goals

Concurrency

What are we bounding here?

Heap manipulation

Separation logic...

Summary

From SMT solvers to verifiers

The UFO approach

UFO in practice

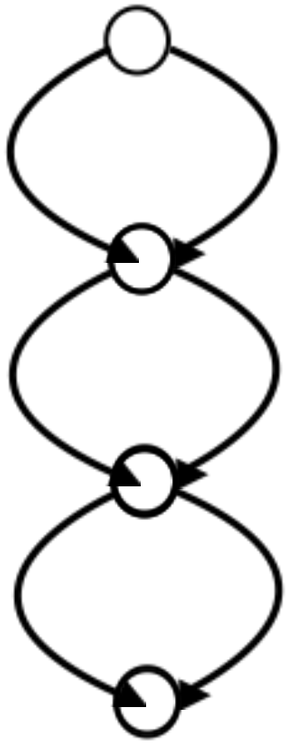
Big questions ahead

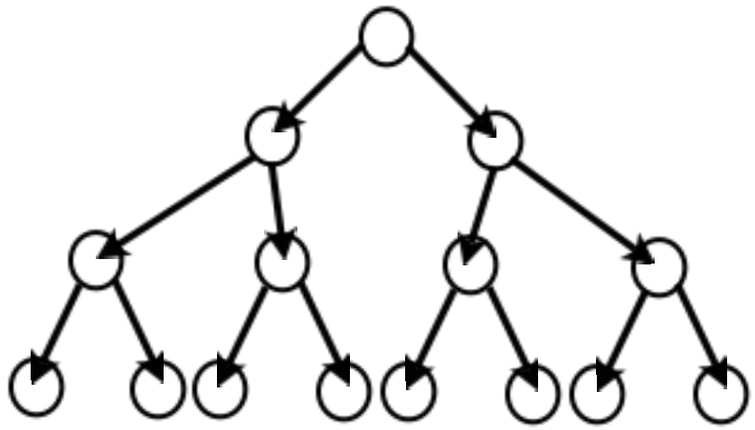
Thank You!

Email:
aws@cs.toronto.edu

UFO source code:
bitbucket.org/arieg/ufo

UFO's Advantages

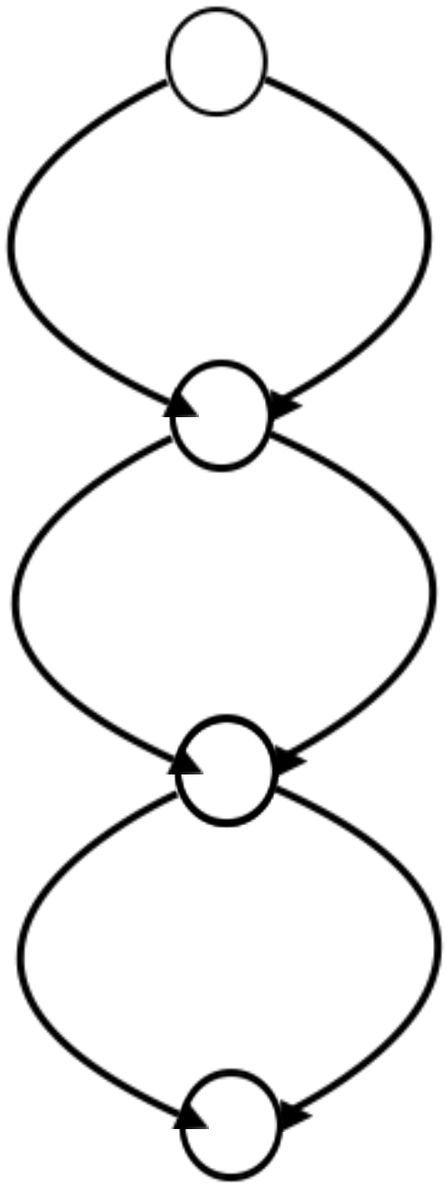




2^n *paths!*

e.g., [McMillan '03, Henzinger et al. '02]

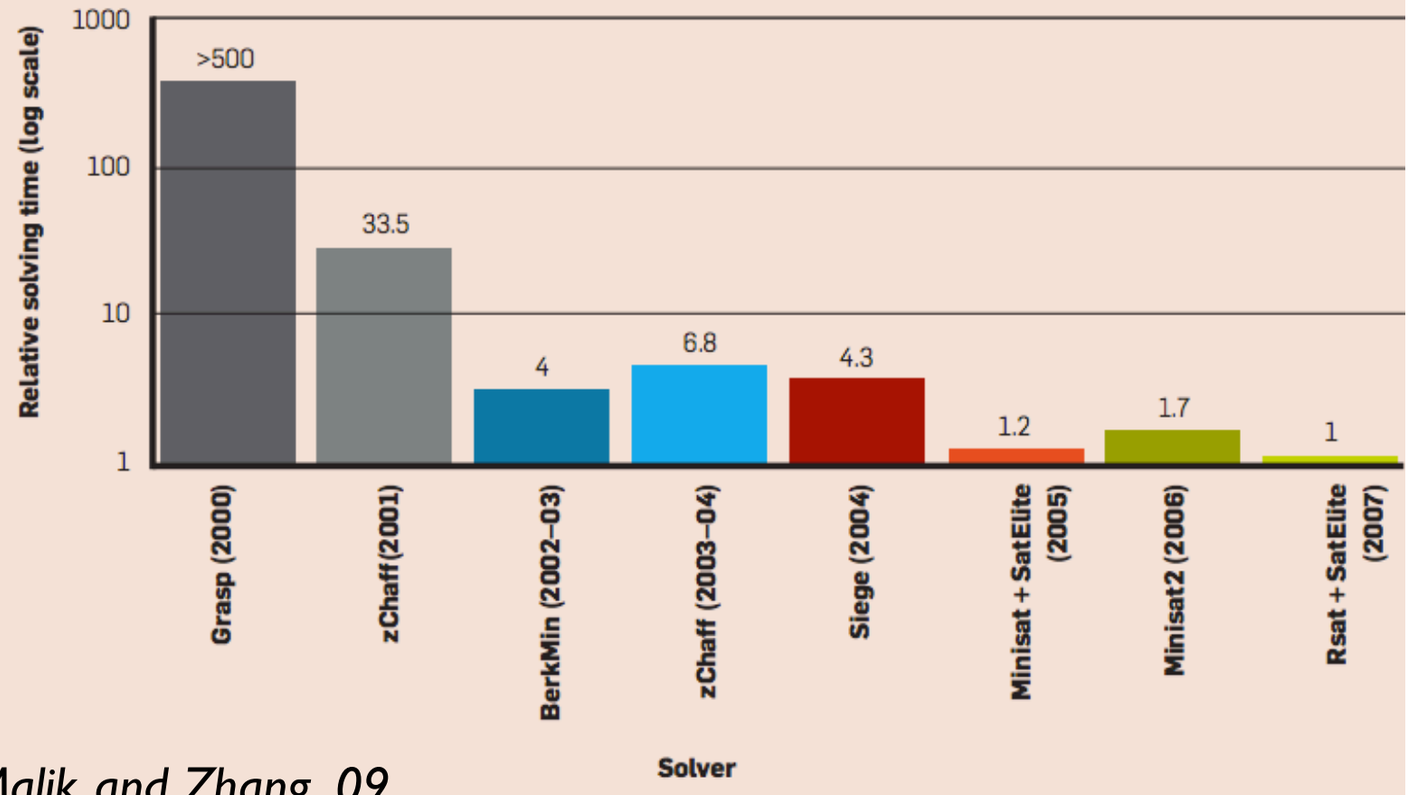
UFO's Advantages



Encode DAG as formula

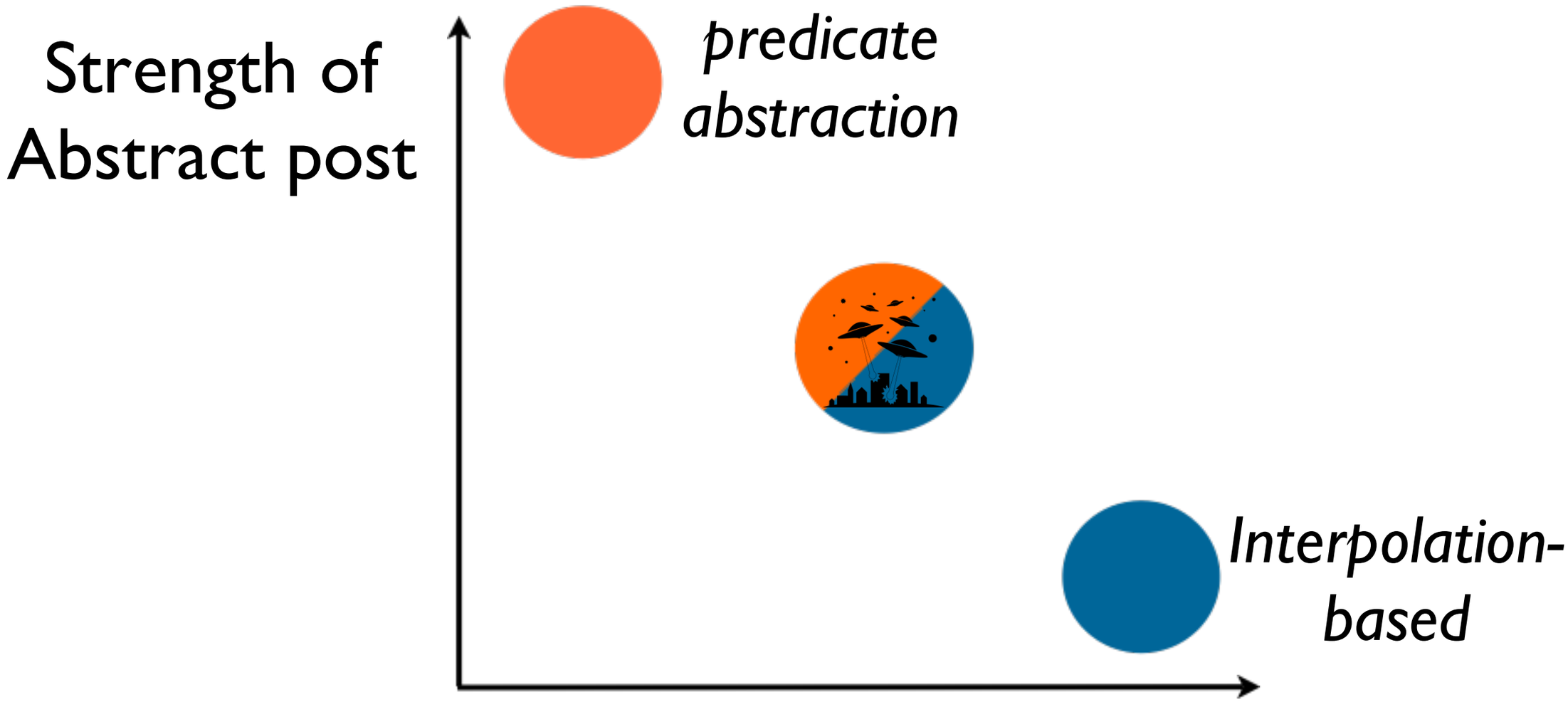
Utilize symbolic reasoning

Figure 4. Speedup of SAT solvers in recent years.

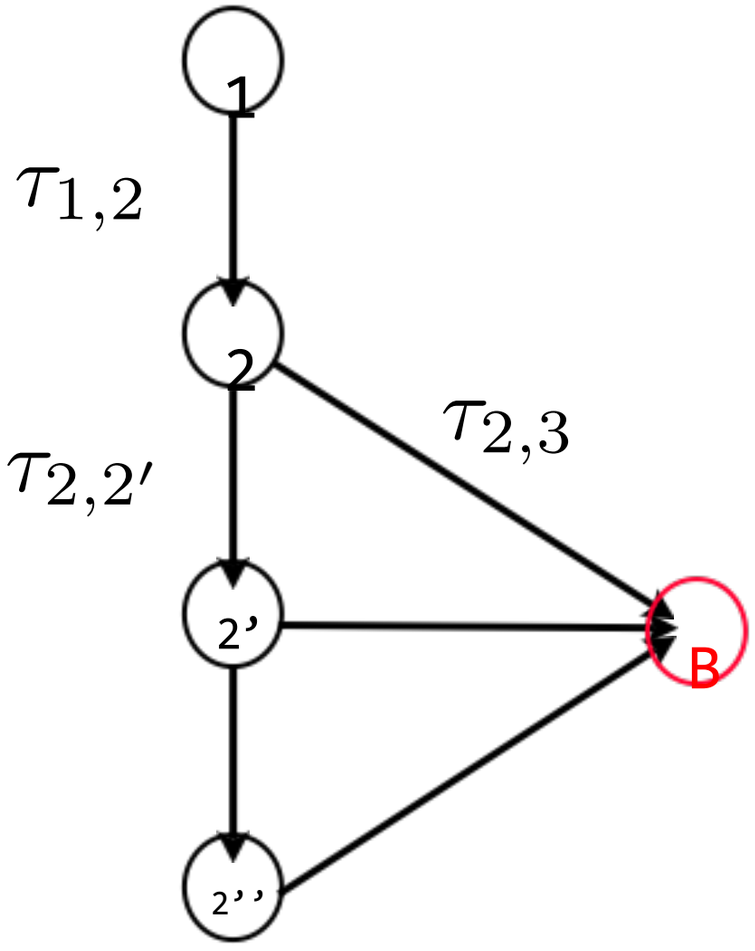


Malik and Zhang, 09

UFO's Advantages



DAG Interpolants



~~Guided DAG Itp~~

$$true \Rightarrow I_1$$
$$I_B \wedge A_B \Rightarrow false$$

For any edge (i, j)

$$I_i \wedge A(\tau_{i,j}) \Rightarrow I_j$$

Results of Abstract Interpretation

$$AI : V \rightarrow Expr$$

```
    if (! ((unsigned long )addr != (unsigned long )((void *)0))) {
      {
      ldv_assume_stop();
      }
    } else {

    }
    if (addr) {
      if (ldv_coherent_state >= 1) {

      } else {
        {
        ldv_blast_assert();
        }
      }
      ldv_coherent_state = ldv_coherent_state - 1;
    } else {

    }
    goto while_break;
  }
  while_break___0: ;
  }
```