QF_BV Property Directed Reachability with Mixed Type Atomic Reasoning Units

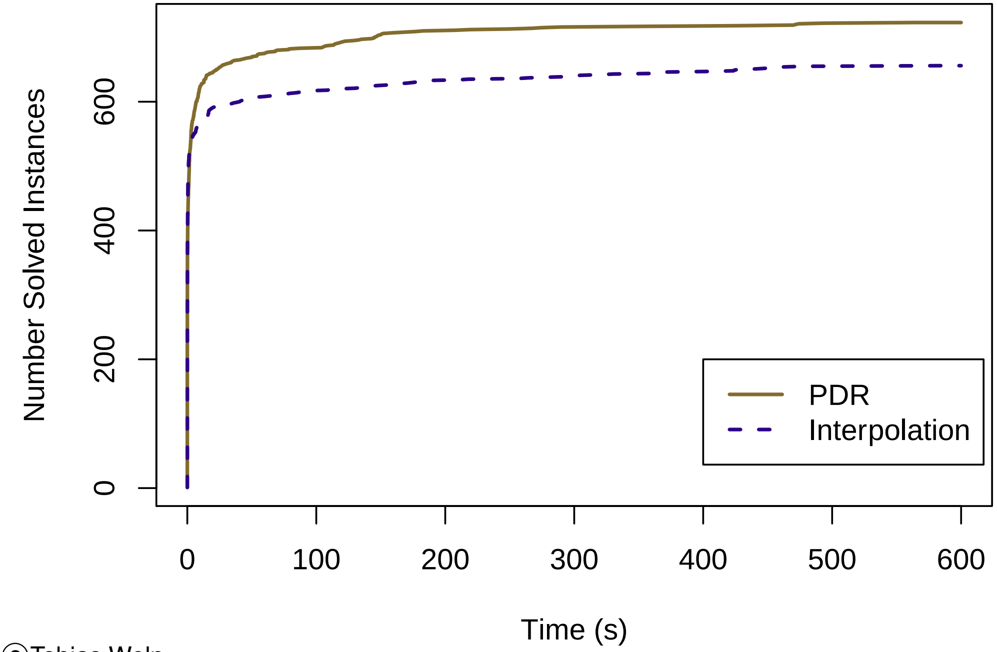
Tobias Welp and Andreas Kuehlmann

Outline

1. **Introduction**
2. QF_BV Property Directed Reachability
3. Mixed Type Atomic Reasoning Units
4. Experimental Results
5. Summary

Motivation for Property Directed Reachability

- In 2011, Bradley proposed *Property Directed Reachability* (a.k.a IC^3) for model checking [Brad11].
- Experiments indicate that PDR outperforms model checking based on *Interpolation* [McMi03] on representative benchmark sets.

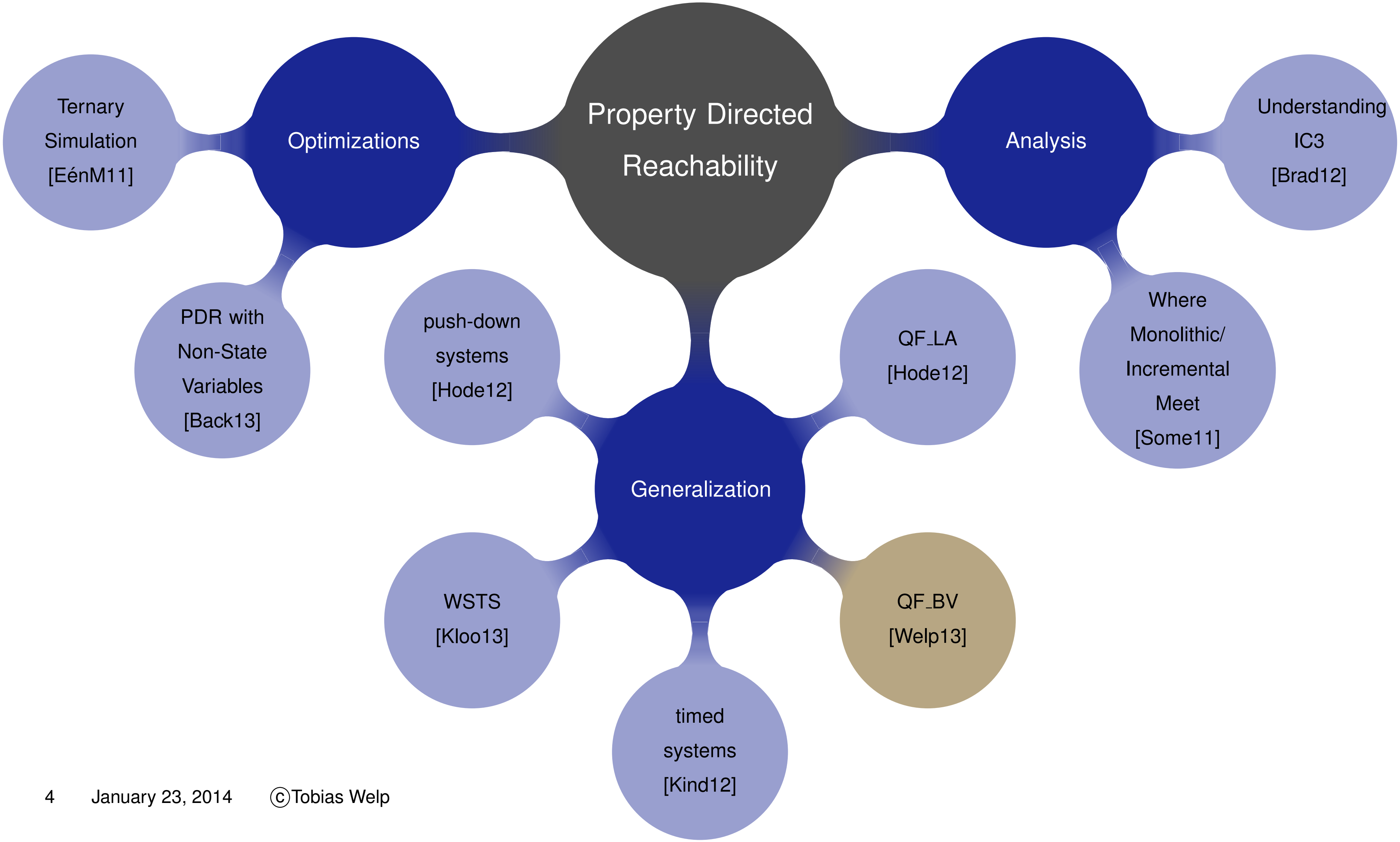


[EénM11]

Other Favorable Properties of PDR

- ☺ No unrolling of transition relation.
- ☺ Parallizable.
- ☺ Allows for initialization with known invariants.
- ☺ Good for finding counterexamples and proving that none exists.

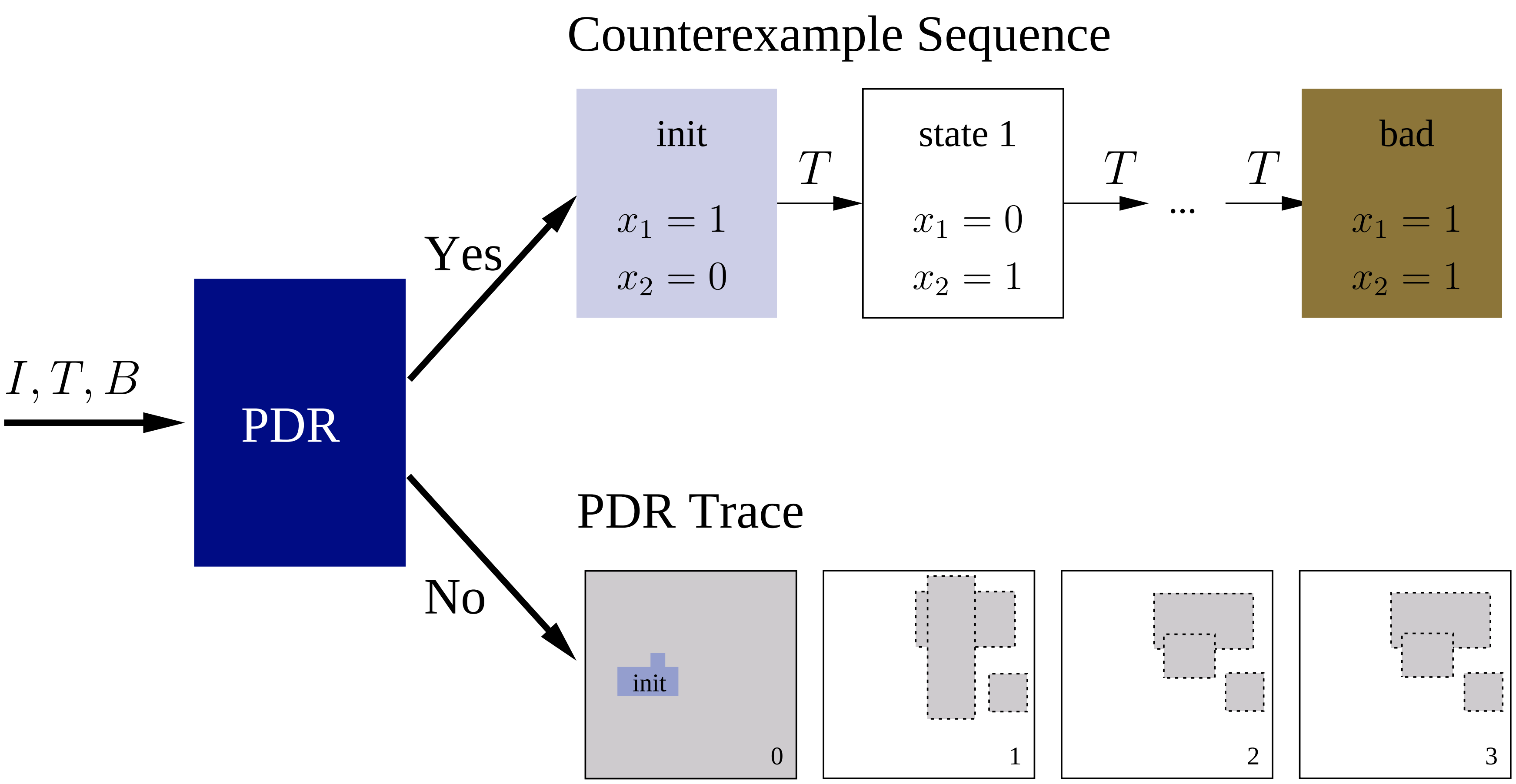
Research Pertaining PDR



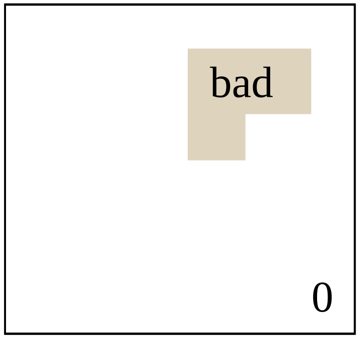
Model Checking

- Given are
  - A set of initial states: $I(\mathbf{x})$
  - A set of bad states: $B(\mathbf{x})$
  - A transition relation: $T(\mathbf{x}, \mathbf{x}')$
- Question: Is a bad state reachable from an initial state using valid transitions?

Model Checking with PDR



Proving a Safety Property with PDR

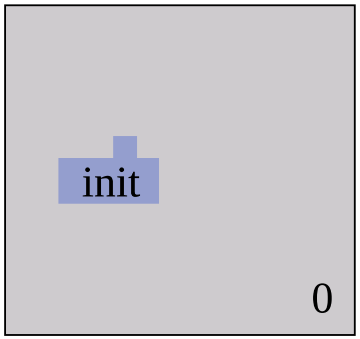


Legend:

- Initial set I
- Bad set B
- Proof oblig.
- Cover

- Can bad be reached within zero steps?

Proving a Safety Property with PDR



Legend:

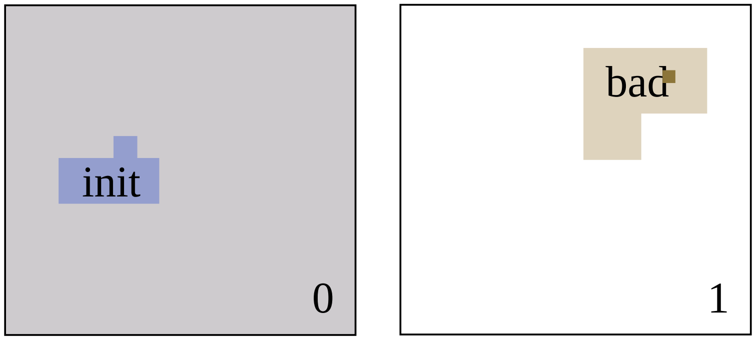
- Initial set I
- Bad set B
- Proof oblig.
- Cover

- No, only the initial set is reachable within zero steps.
- Everything else is *covered*, i.e. not reachable.

Proving a Safety Property with PDR

- Can bad be reached within one step?
- Conservatively, we initially assume that everything is *reachable*.

Proving a Safety Property with PDR

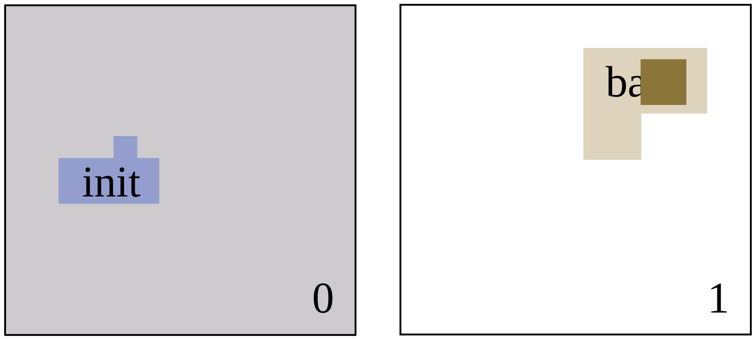


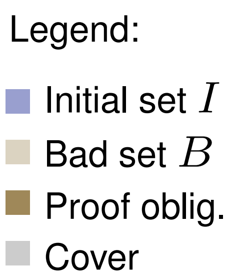
Legend:

- Initial set I
- Bad set B
- Proof oblig.
- Cover

- Find a point in bad that is not yet covered.

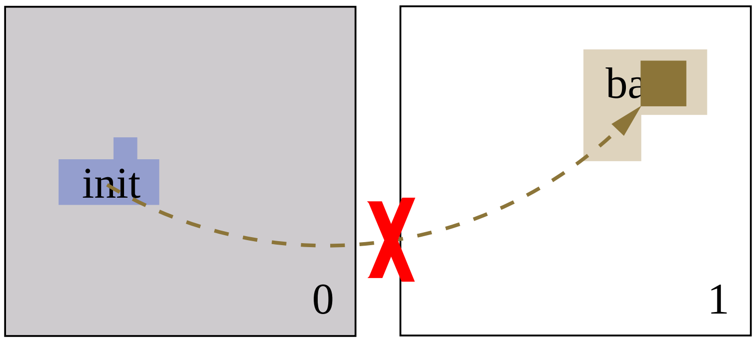
Proving a Safety Property with PDR





- Expand proof obligation using simulation.

Proving a Safety Property with PDR

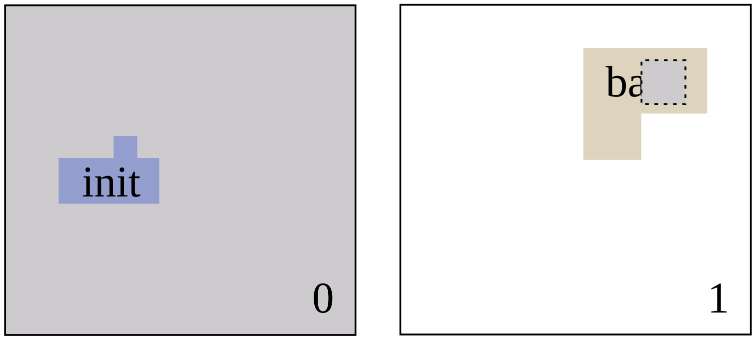


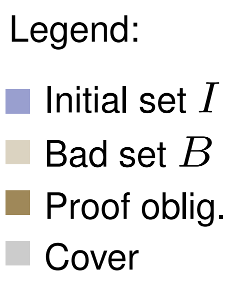
Legend:

- Initial set I
- Bad set B
- Proof oblig.
- Cover

- The cube cannot be reached from the reachable area in frame 0.

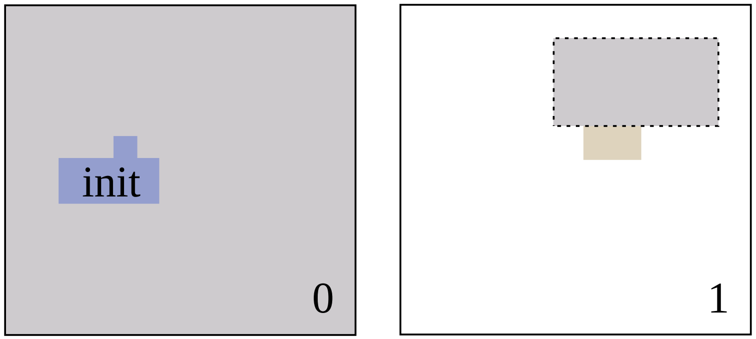
Proving a Safety Property with PDR





- Hence, we can consider the proof obligation covered.

Proving a Safety Property with PDR

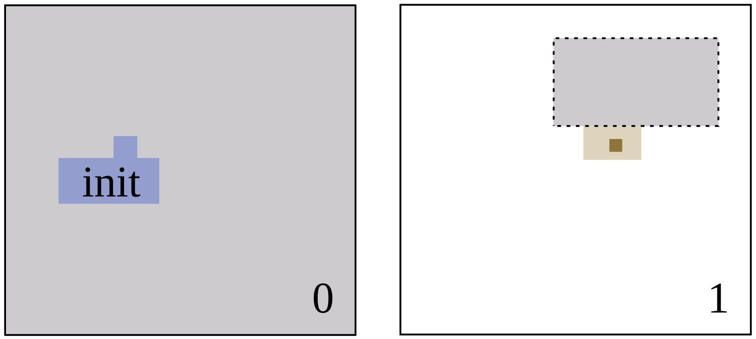


Legend:

- Initial set I
- Bad set B
- Proof oblig.
- Cover

- Expand the covered cube as much as possible.

Proving a Safety Property with PDR

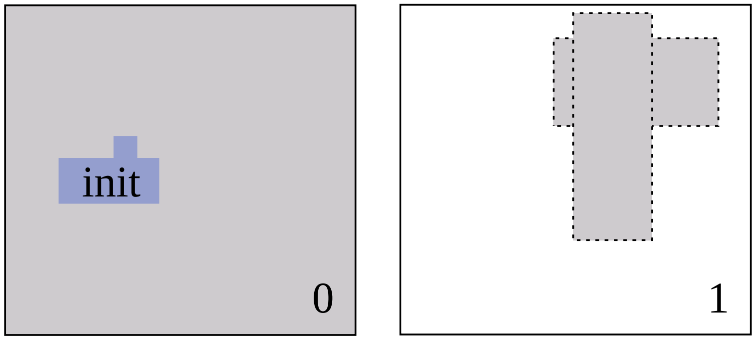


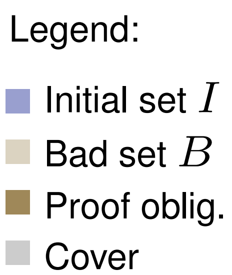
Legend:

- Initial set I
- Bad set B
- Proof oblig.
- Cover

- Repeat with finding a new point in bad that is not covered.

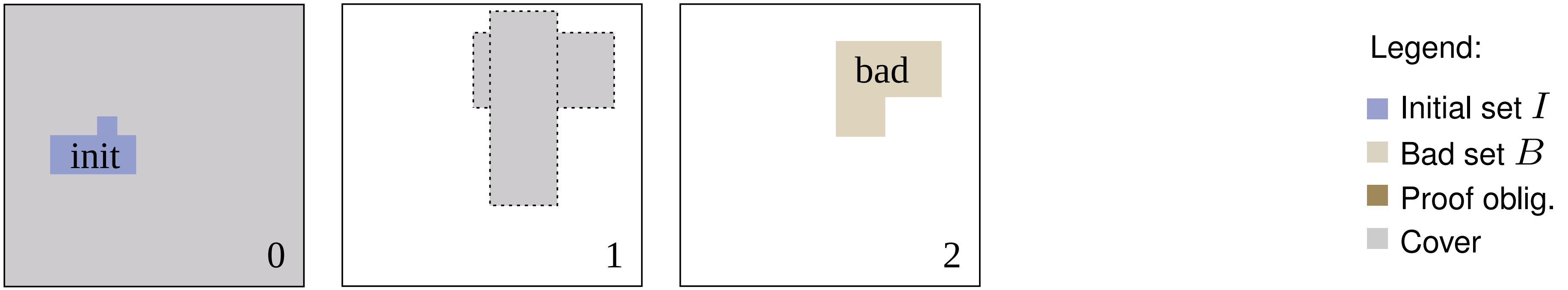
Proving a Safety Property with PDR





- Again, the point cannot be reached from the reachable area in the previous frame.
- Expand the covered cube.
- Now, bad is completely covered.

Proving a Safety Property with PDR

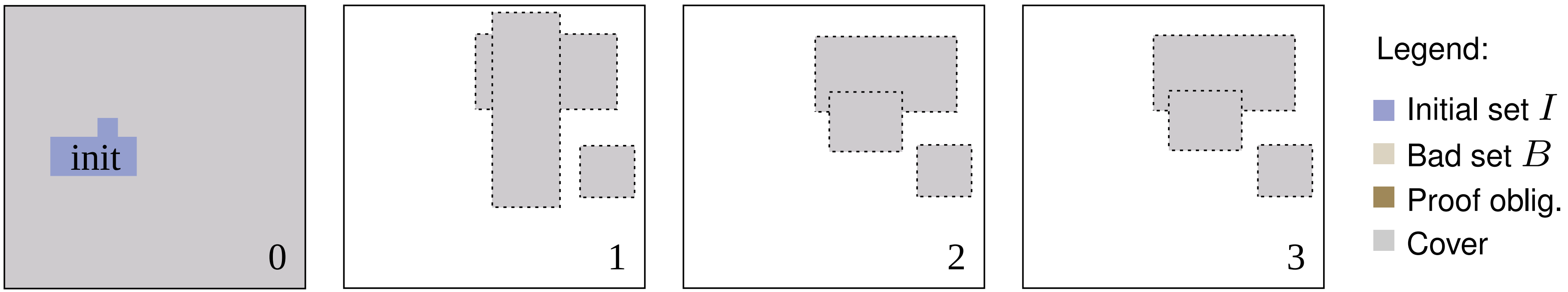


- Can bad be reached within two steps?

Proving a Safety Property with PDR

. . .

Proving a Safety Property with PDR

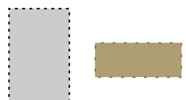
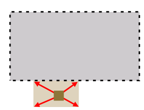


- Identified an inductive invariant disjoint from bad.
- This proves the property.

Outline

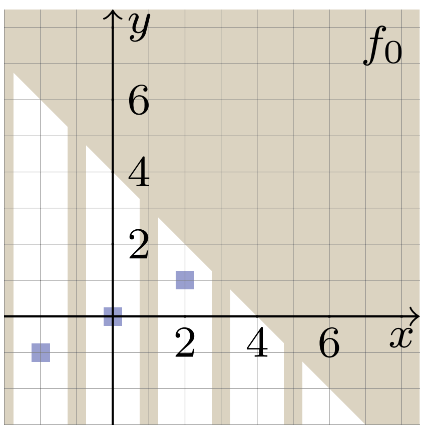
1. Introduction
2. **QF_BV Property Directed Reachability**
3. Mixed Type Atomic Reasoning Units
4. Experimental Results
5. Summary

Property Directed Reachability for QF_BV

	Original Formulation		
Atomic Reasoning Unit 	Boolean Cubes		
Expansion of Proof Obligations 	Ternary Simulation		
Strengths			
Weaknesses			

Example Hybrid Invariant

$$
\begin{aligned}
I &:= (2 \times y \equiv x) \wedge (x + y \leq 3) \\
T &:= (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x) \\
B &:= (x + y \geq 4) \vee (x \bmod 2 \equiv 1)
\end{aligned}
$$



Legend:

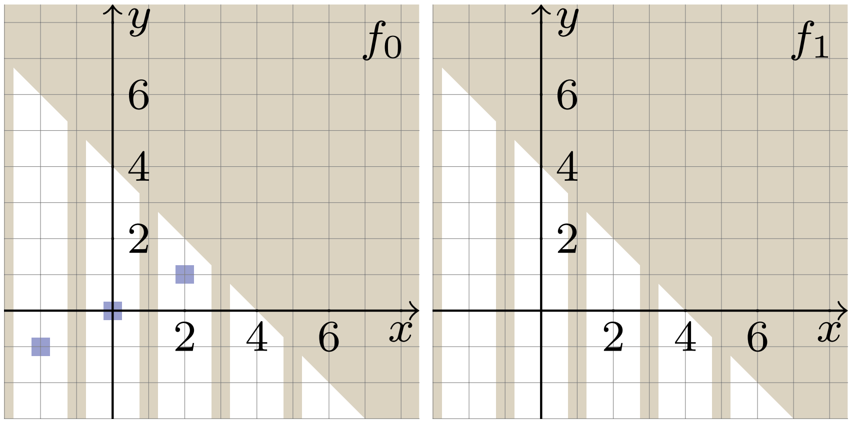
- Initial set I
- Bad set B
- Proof oblig.
- Cover

Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$

$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$

$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$



Legend:

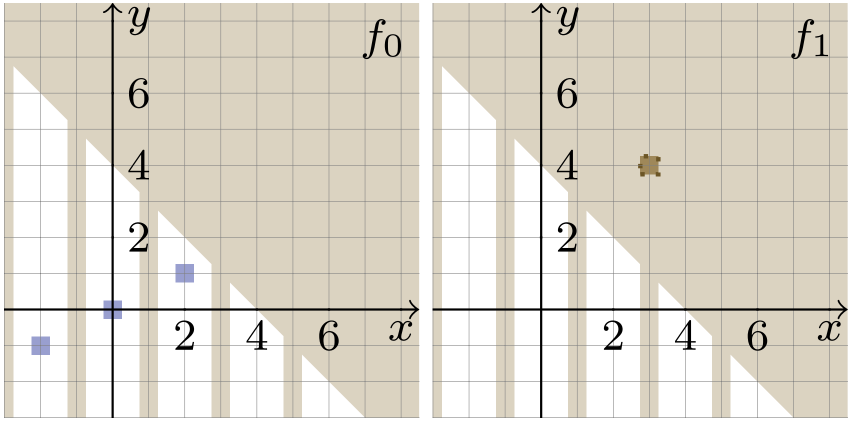
- Initial set I
- Bad set B
- Proof oblig.
- Cover

Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$

$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$

$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$



Legend:

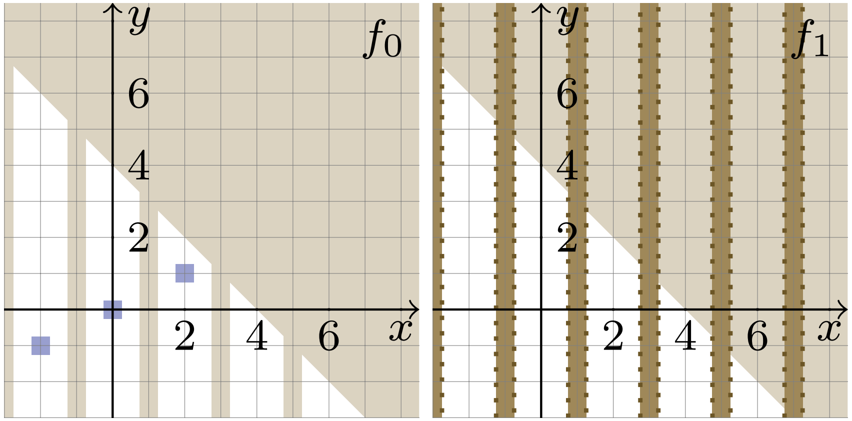
- Initial set I
- Bad set B
- Proof oblig.
- Cover

Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$

$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$

$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$

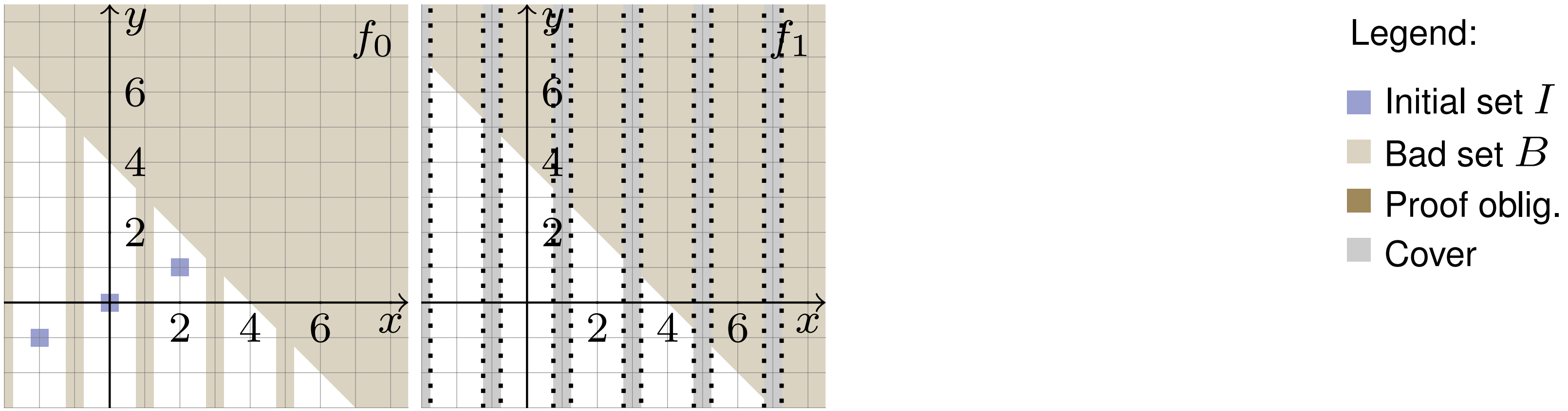


Legend:

- Initial set I
- Bad set B
- Proof oblig.
- Cover

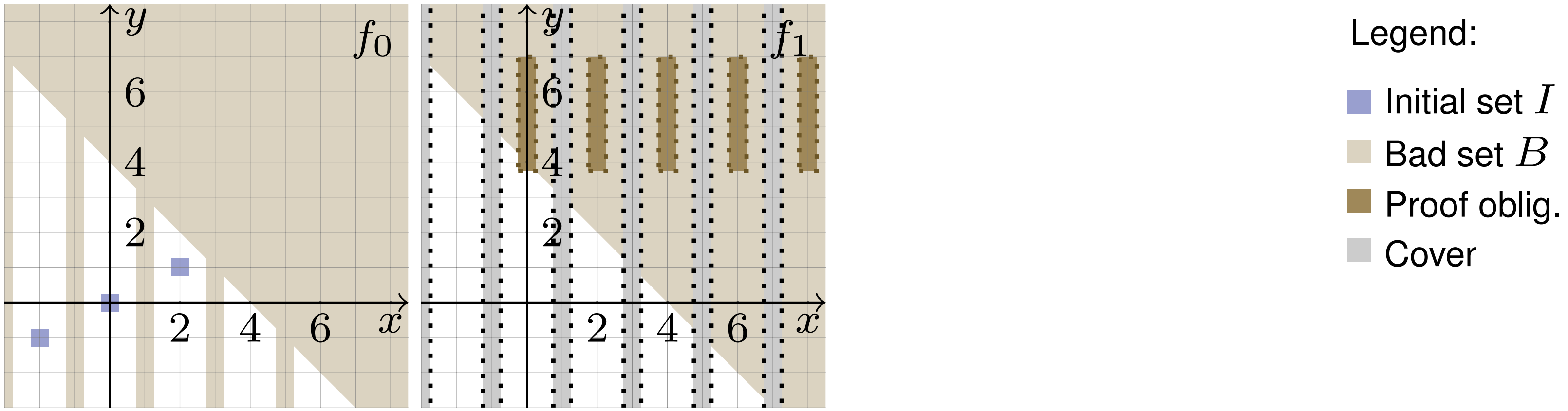
Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$
$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$
$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$



Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$
$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$
$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$

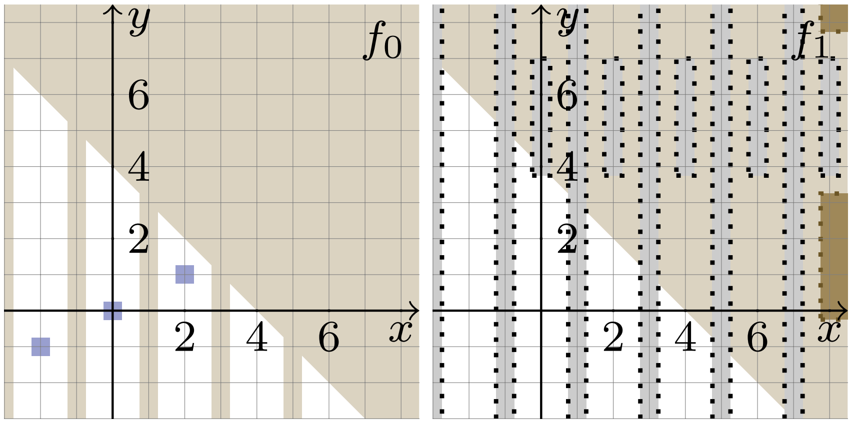


Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$

$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$

$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$



Legend:

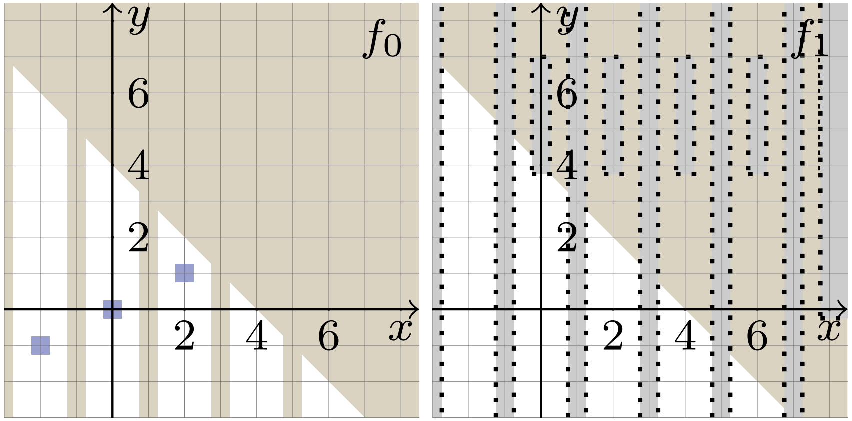
- Initial set I
- Bad set B
- Proof oblig.
- Cover

Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$

$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$

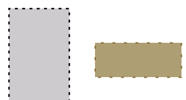
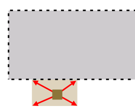
$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$



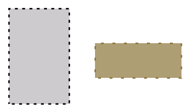
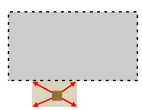
Legend:

- Initial set I
- Bad set B
- Proof oblig.
- Cover

Property Directed Reachability for QF_BV

	Original Formulation		
Atomic Reasoning Unit 	Boolean Cubes		
Expansion of Proof Obligations 	Ternary Simulation		
Strengths	logic		
Weaknesses	arithmetic		

Property Directed Reachability for QF_BV

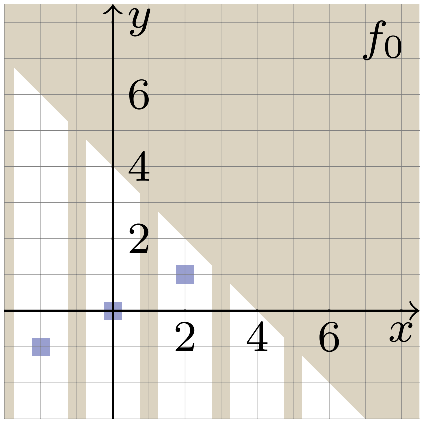
	Original Formulation	Polytopes [Welp13]	
Atomic Reasoning Unit 	Boolean Cubes	Polytopes	
Expansion of Proof Obligations 	Ternary Simulation	Interval Simulation	
Strengths	logic		
Weaknesses	arithmetic		

Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$

$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$

$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$



Legend:

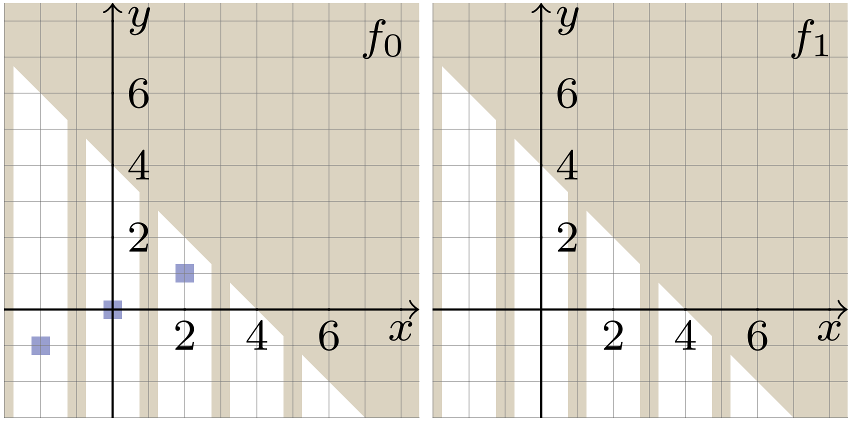
- Initial set I
- Bad set B
- Proof oblig.
- Cover

Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$

$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$

$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$

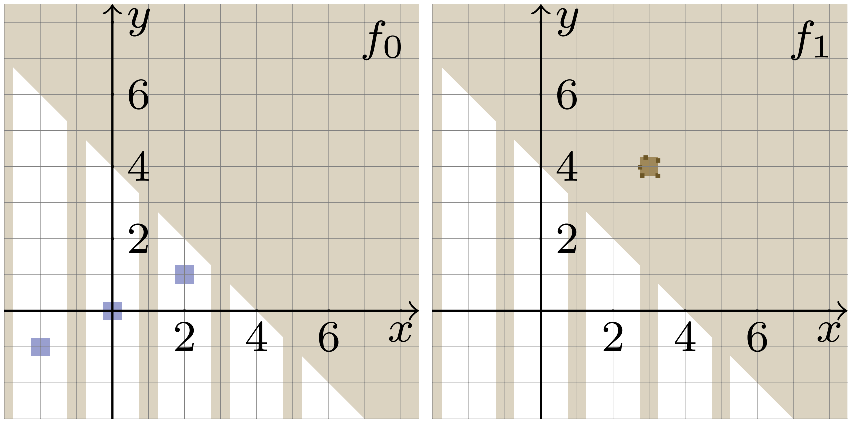


Legend:

- Initial set I
- Bad set B
- Proof oblig.
- Cover

Example Hybrid Invariant

$$
\begin{aligned}
I &:= (2 \times y \equiv x) \wedge (x + y \leq 3) \\
T &:= (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x) \\
B &:= (x + y \geq 4) \vee (x \bmod 2 \equiv 1)
\end{aligned}
$$

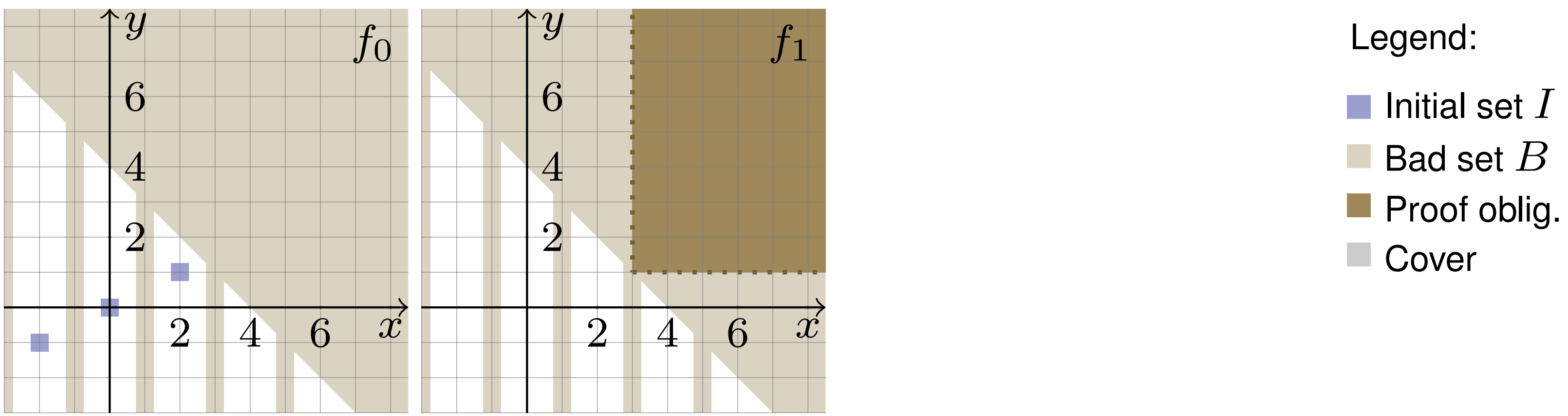


Legend:

- Initial set I
- Bad set B
- Proof oblig.
- Cover

Example Hybrid Invariant

$$
\begin{aligned}
I &:= (2 \times y \equiv x) \wedge (x + y \leq 3) \\
T &:= (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x) \\
B &:= (x + y \geq 4) \vee (x \bmod 2 \equiv 1)
\end{aligned}
$$

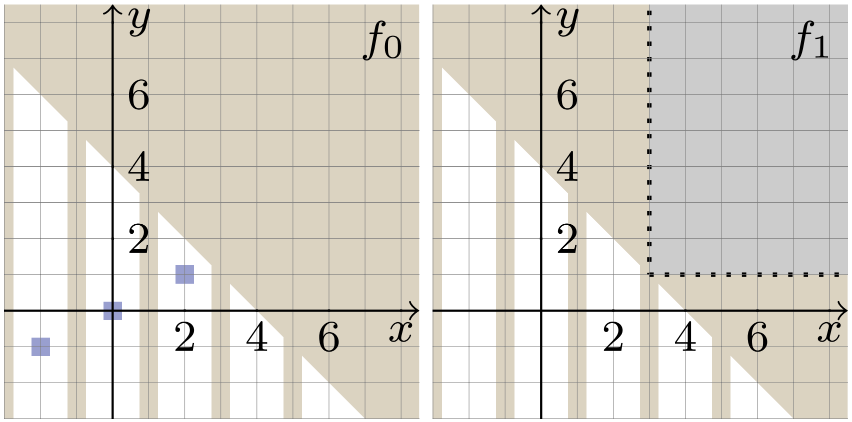


Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$

$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$

$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$

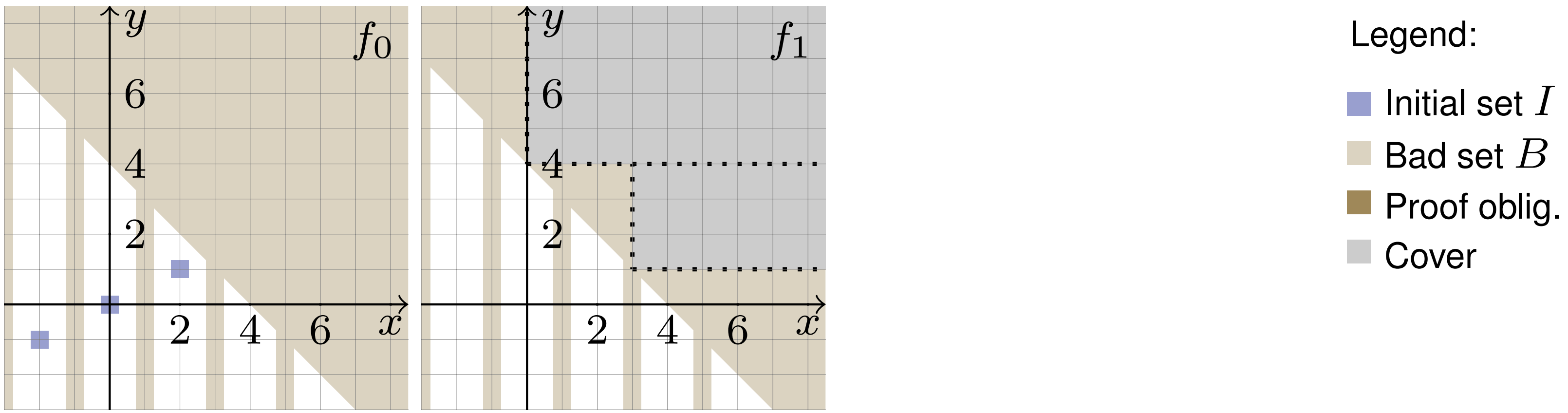


Legend:

- Initial set I
- Bad set B
- Proof oblig.
- Cover

Example Hybrid Invariant

$$
\begin{aligned}
I &:= (2 \times y \equiv x) \wedge (x + y \leq 3) \\
T &:= (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x) \\
B &:= (x + y \geq 4) \vee (x \bmod 2 \equiv 1)
\end{aligned}
$$

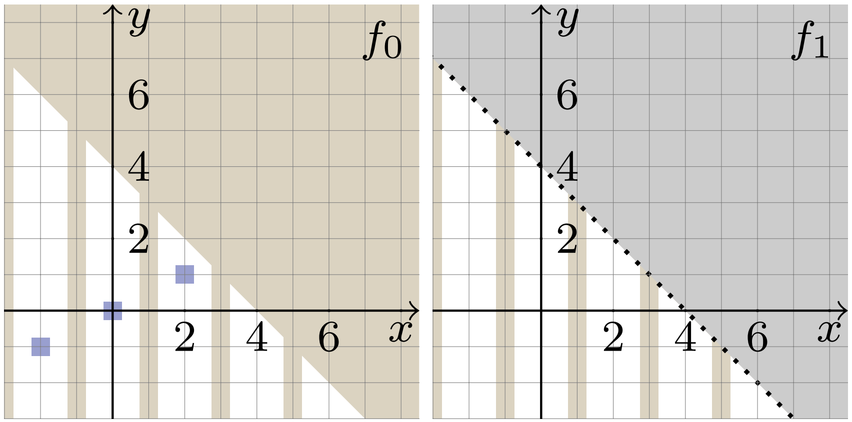


Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$

$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$

$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$

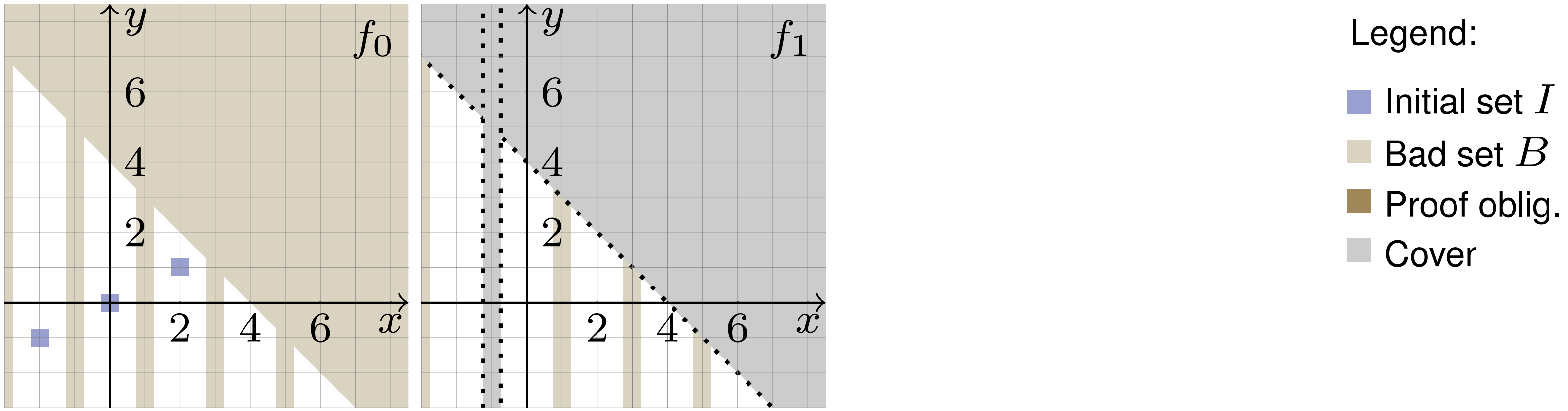


Legend:

- Initial set I
- Bad set B
- Proof oblig.
- Cover

Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$
$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$
$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$

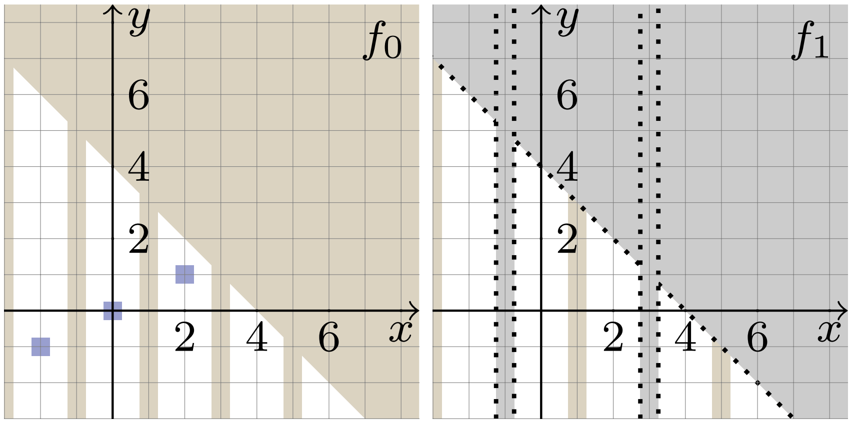


Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$

$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$

$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$

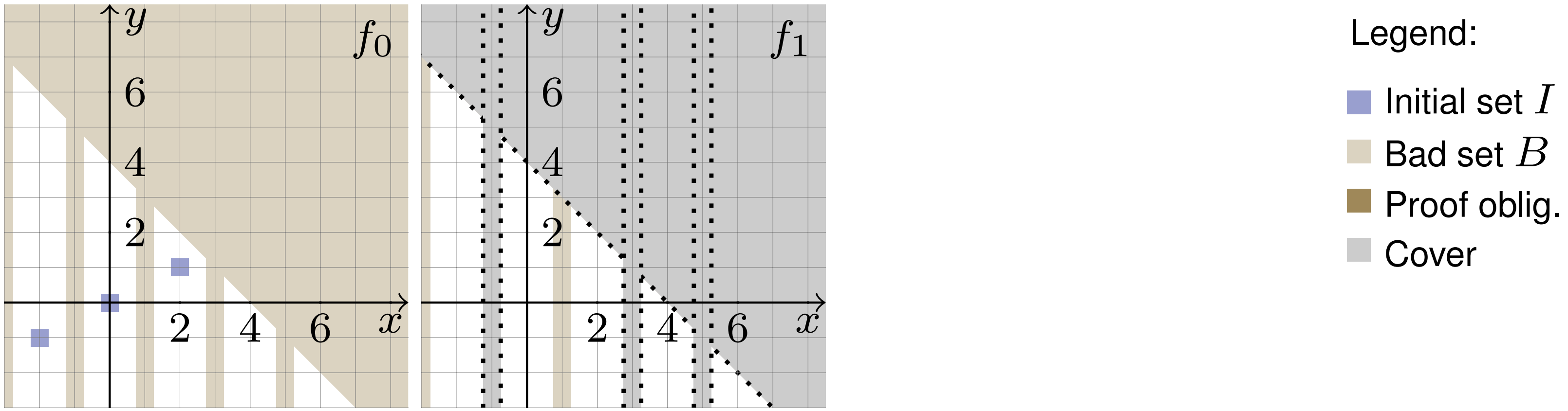


Legend:

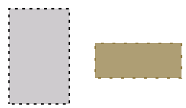
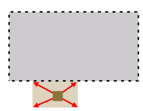
- Initial set I
- Bad set B
- Proof oblig.
- Cover

Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$
$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$
$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$



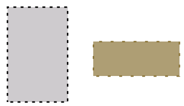
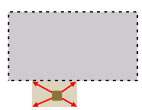
Property Directed Reachability for QF_BV

	Original Formulation	Polytopes [Welp13]	
Atomic Reasoning Unit 	Boolean Cubes	Polytopes	
Expansion of Proof Obligations 	Ternary Simulation	Interval Simulation	
Strengths	logic	arithmetic	
Weaknesses	arithmetic	logic	

Outline

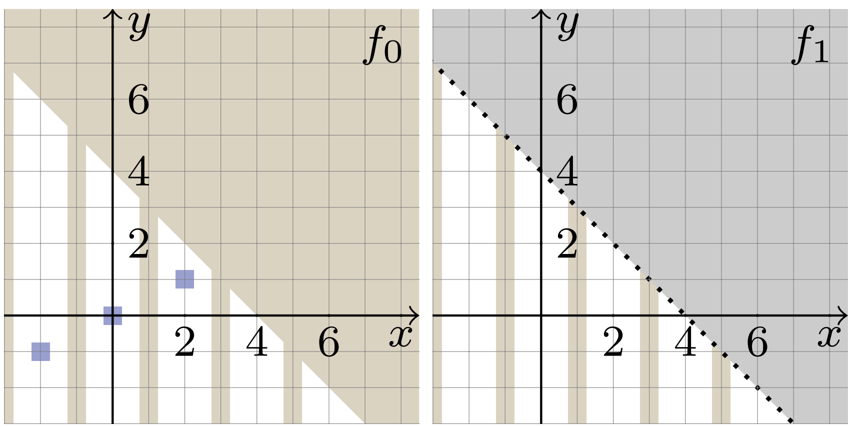
1. Introduction
2. QF_BV Property Directed Reachability
3. **Mixed Type Atomic Reasoning Units**
4. Experimental Results
5. Summary

Property Directed Reachability for QF_BV

	Original Formulation	Polytopes [Welp13]	Hybrid
Atomic Reasoning Unit 	Boolean Cubes	Polytopes	Boolean Cubes and Polytopes
Expansion of Proof Obligations 	Ternary Simulation	Interval Simulation	Hybrid Simulation
Strengths	logic	arithmetic	
Weaknesses	arithmetic	logic	

Example Hybrid Invariant

$$
\begin{aligned}
I &:= (2 \times y \equiv x) \wedge (x + y \leq 3) \\
T &:= (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x) \\
B &:= (x + y \geq 4) \vee (x \bmod 2 \equiv 1)
\end{aligned}
$$

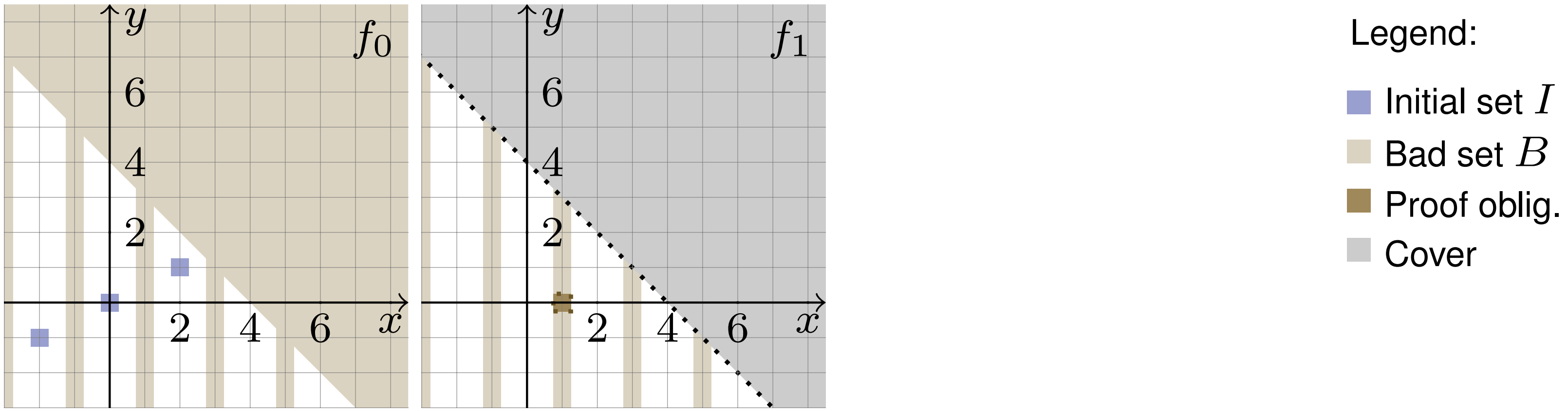


Legend:

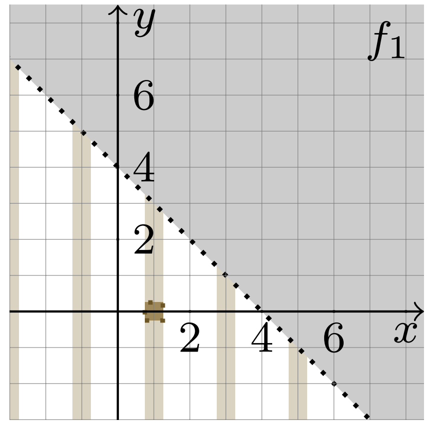
- Initial set I
- Bad set B
- Proof oblig.
- Cover

Example Hybrid Invariant

$$I := (2 \times y \equiv x) \wedge (x + y \leq 3)$$
$$T := (y' \equiv y + 1) \wedge (x' \equiv x - 2) \wedge (y' > y) \wedge (x' < x)$$
$$B := (x + y \geq 4) \vee (x \bmod 2 \equiv 1)$$

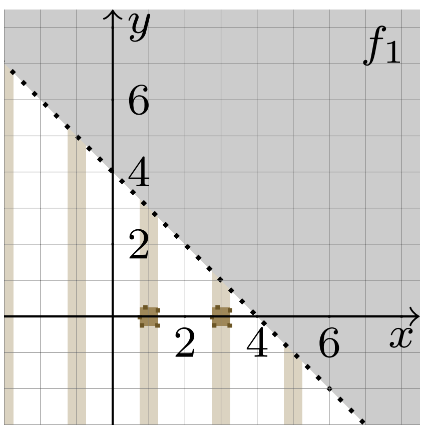


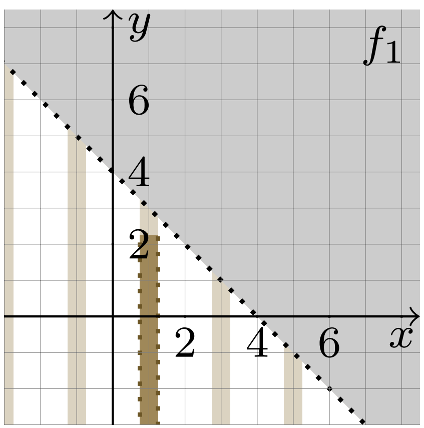
Probabilistic Specialization



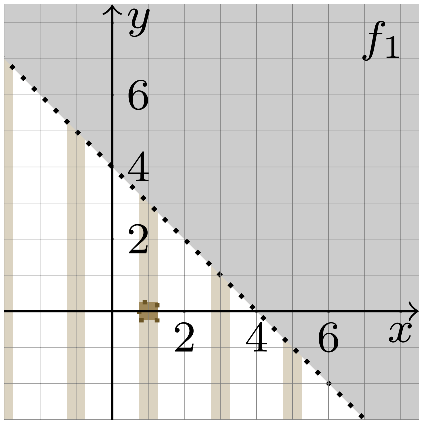
specialize to Boolean cube with probability c

specialize to polytope with probability $1 - c$



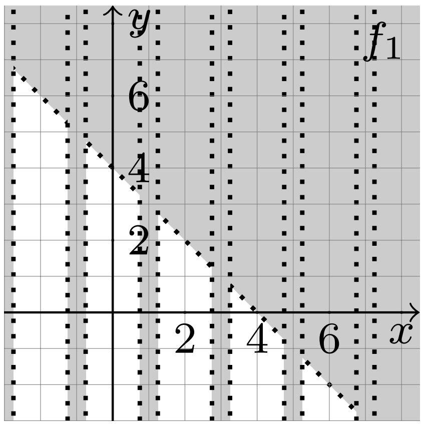


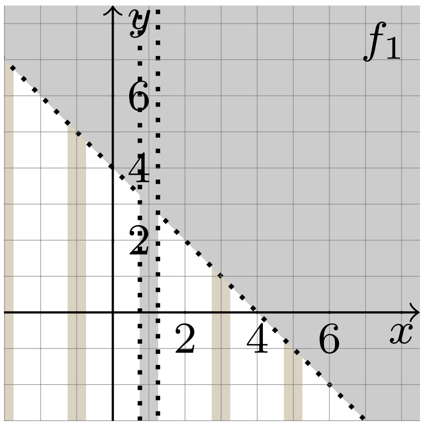
Probabilistic Specialization

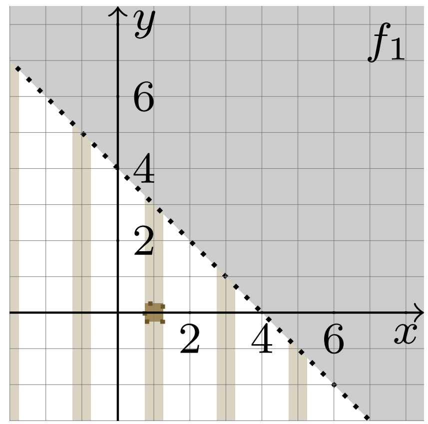


specialize to Boolean cube with probability c

specialize to polytope with probability $1 - c$

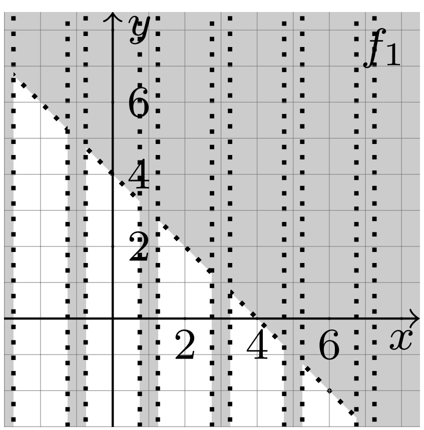


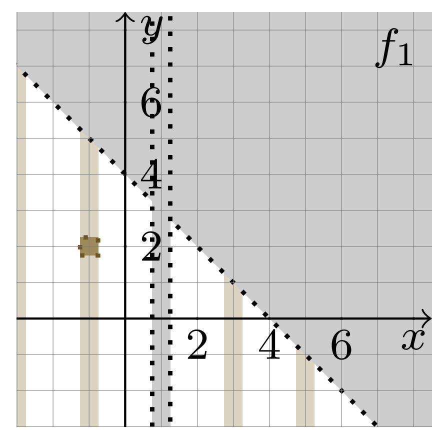




specialize to Boolean cube with probability c

specialize to polytope with probability $1 - c$





Probabilistic Specialization

- The favorable event can be expected to happen in a constant number of steps.

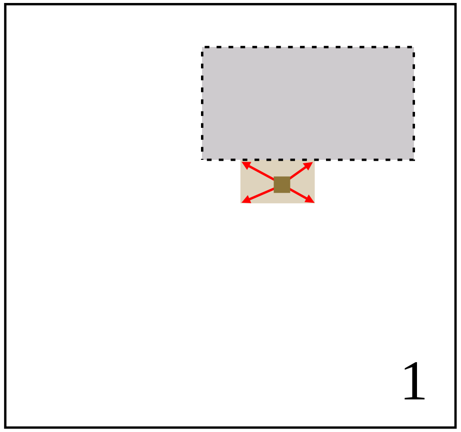
$$E\{\text{trials until Boolean cube specialization}\} = c\sum_{i=1}^{\infty} i(1-c)^{i-1} = \frac{1}{c}$$

- Analogously, one calculates

$$E\{\text{trials until polytope specialization}\} = (1-c)\sum_{i=1}^{\infty} ic^{i-1} = \frac{1}{1-c}$$

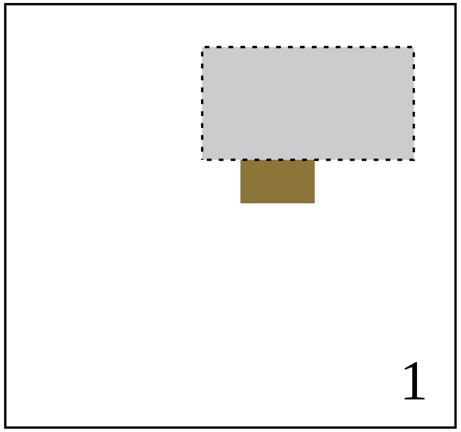
Simulation-based Expansion

Simulation-based expansion of proof obligations is e.g. used to expand a bad ARU that is not yet covered:



Simulation-based Expansion

Simulation-based expansion of proof obligations is e.g. used to expand a bad ARU that is not yet covered:

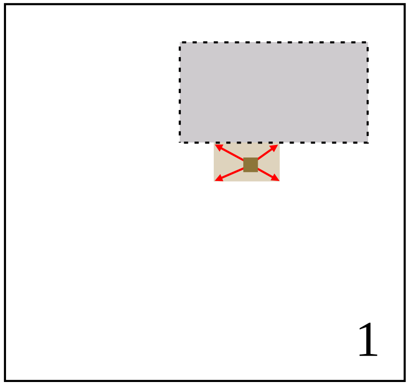


Simulation-based Expansion

The check whether or not an expansion is valid can be reduced to simulation on sets of points [EénM11].

Assume bad is defined as $e_1 < 2$ and we already covered $e_1 < -1$ with

$$e_1 := (x_1 - x_2 + 2) \wedge (y_1 \vee y_2)$$

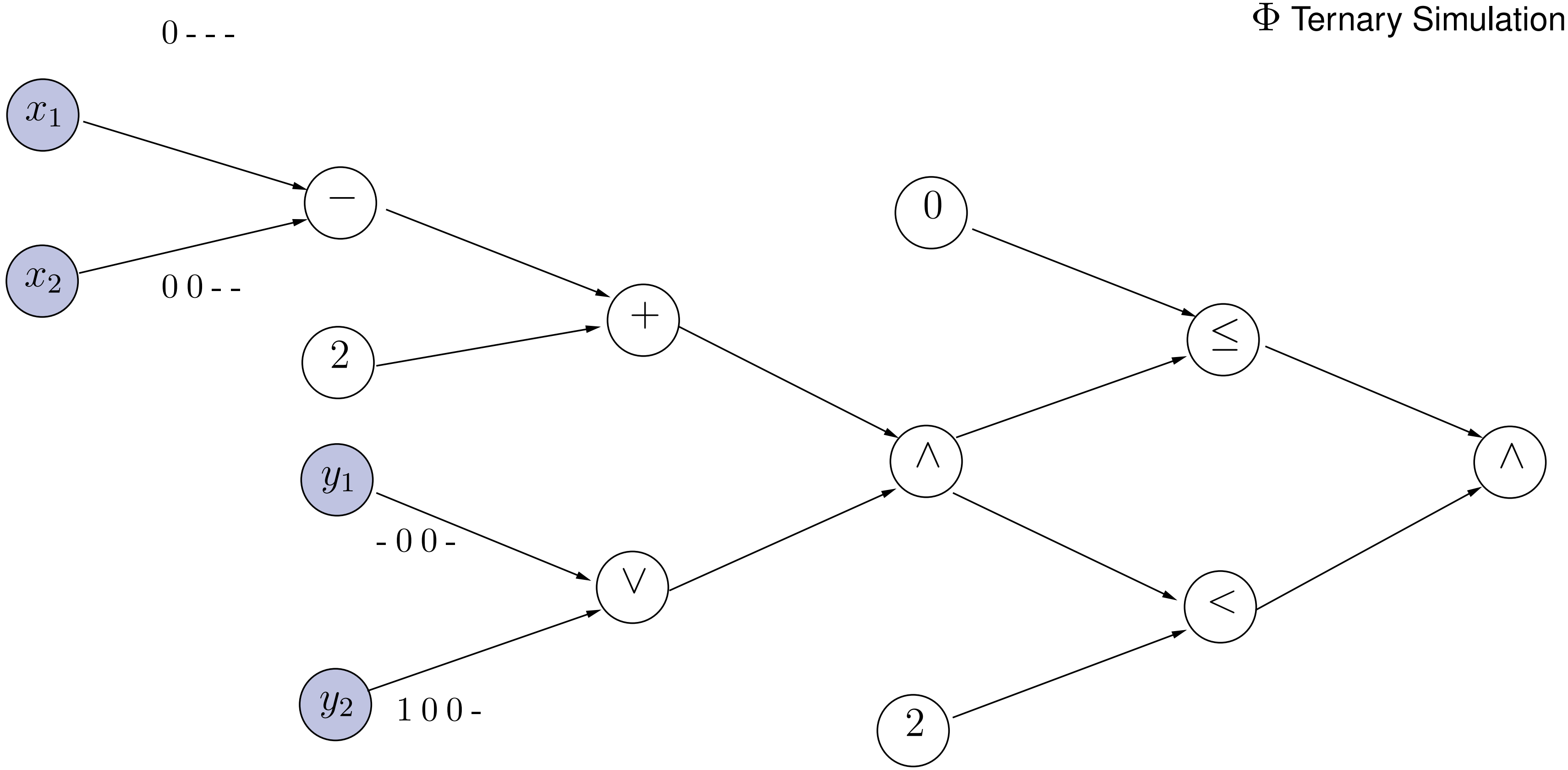


Then we may expand an ARU to a larger ARU $\hat{a}$ if $e := \underbrace{(e_1 < 2)}_{\text{bad}} \wedge \neg \underbrace{(e_1 < -1)}_{\text{covered}}$ evaluates to **true** for all values in $\hat{a}$.

Ternary Simulation

Let $\hat{a} := (1 \leq x_1 \leq 5) \wedge (0 \leq x_2 \leq 3) \wedge (y_1 \in \text{-00-}) \wedge (y_2 \in 100\text{-})$

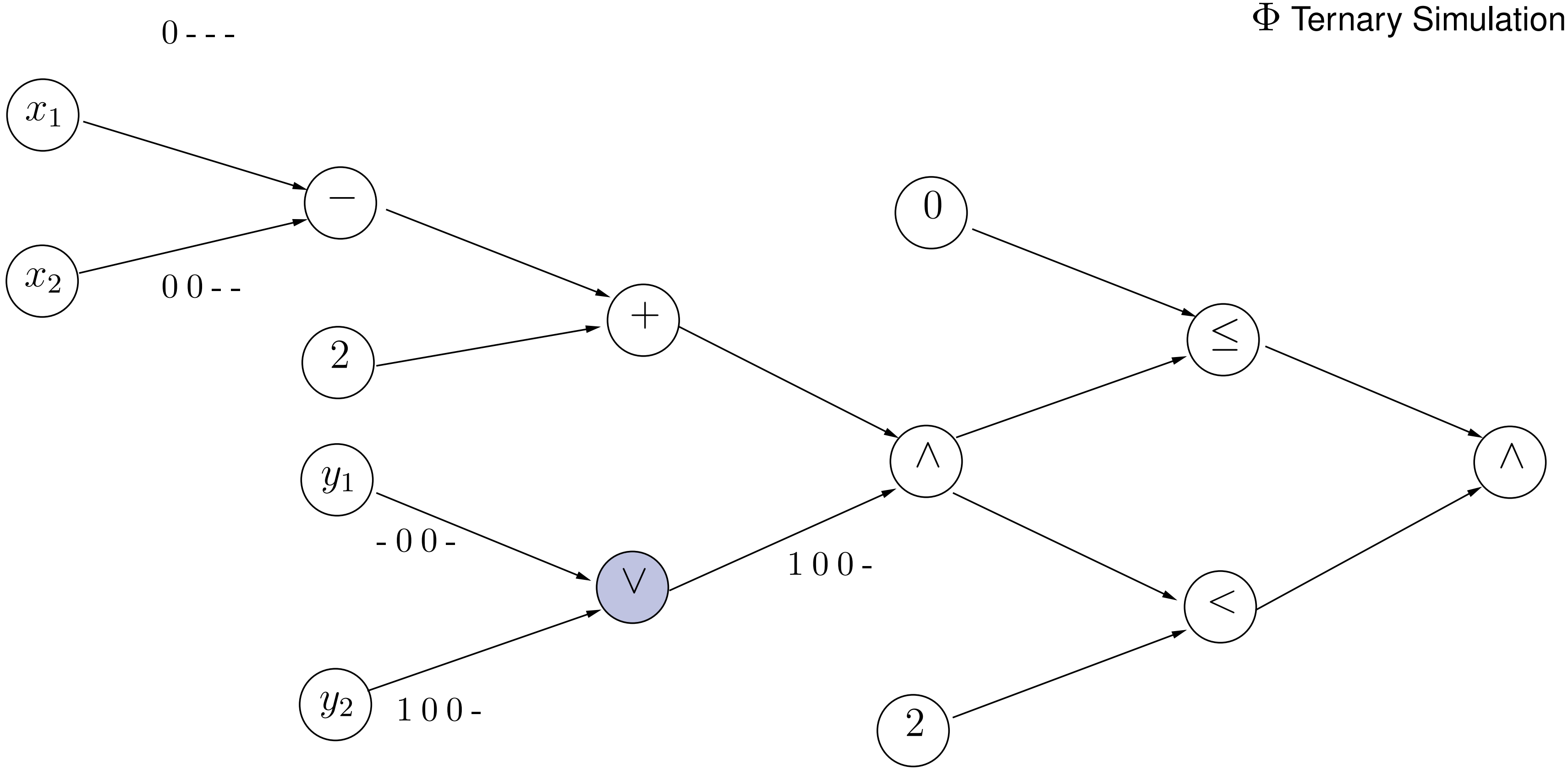
Φ Ternary Simulation



Ternary Simulation

Let $\hat{a} := (1 \leq x_1 \leq 5) \wedge (0 \leq x_2 \leq 3) \wedge (y_1 \in \text{-00-}) \wedge (y_2 \in 100\text{-})$

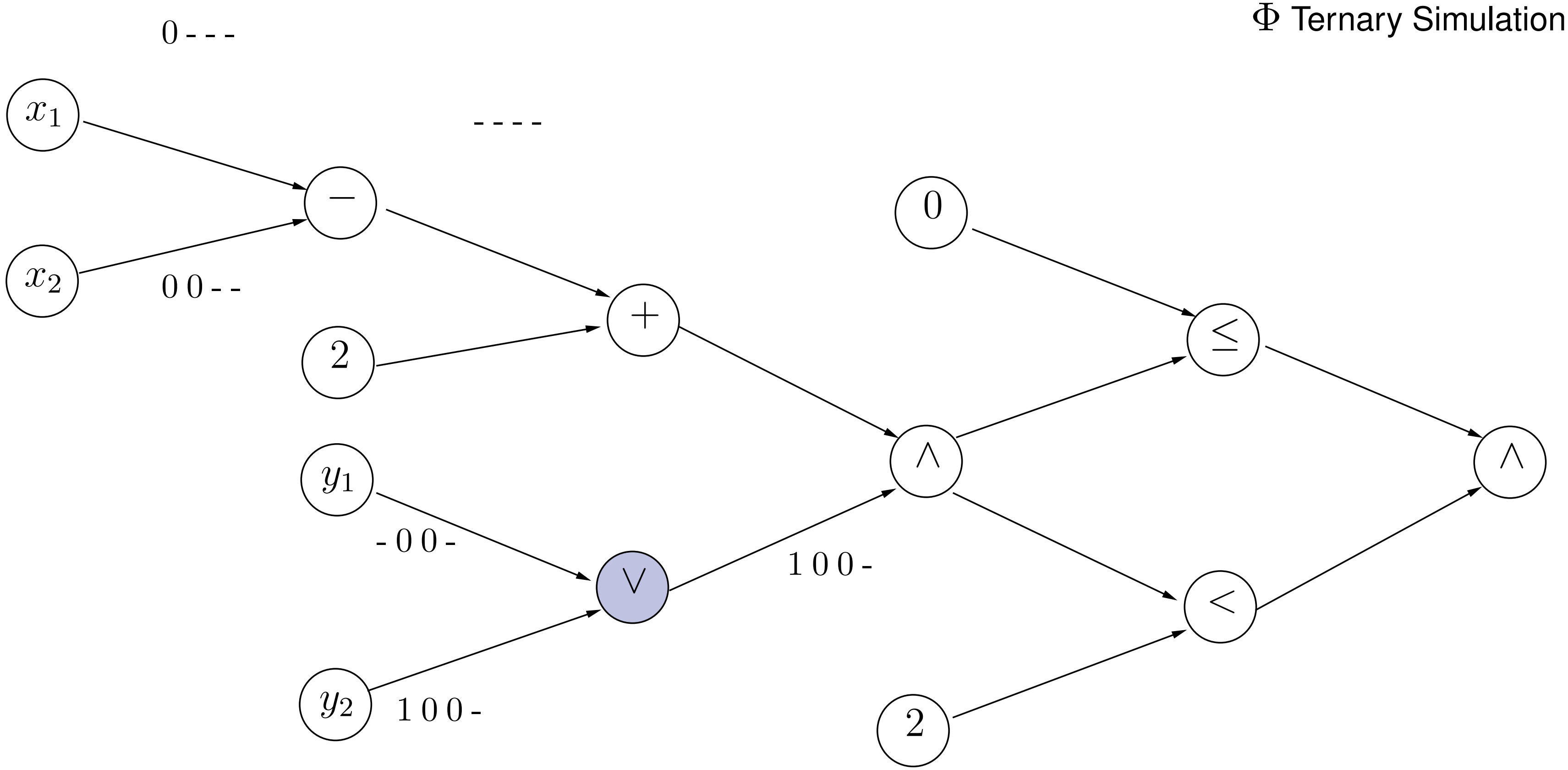
Φ Ternary Simulation



Ternary Simulation

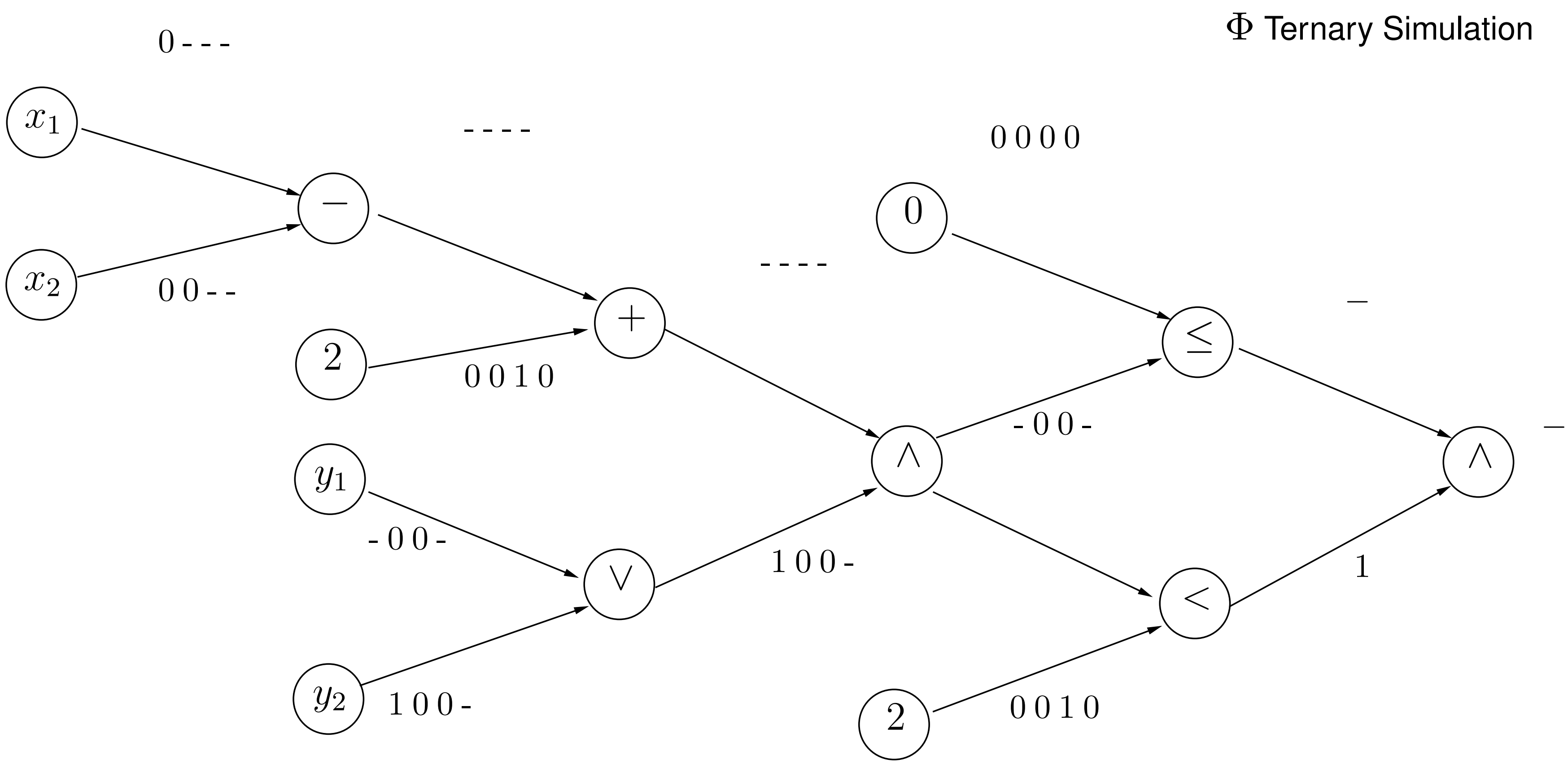
Let $\hat{a} := (1 \leq x_1 \leq 5) \wedge (0 \leq x_2 \leq 3) \wedge (y_1 \in \text{-00-}) \wedge (y_2 \in 100\text{-})$

Φ Ternary Simulation



Ternary Simulation

Let $\hat{a} := (1 \leq x_1 \leq 5) \wedge (0 \leq x_2 \leq 3) \wedge (y_1 \in \text{-00-}) \wedge (y_2 \in 100\text{-})$

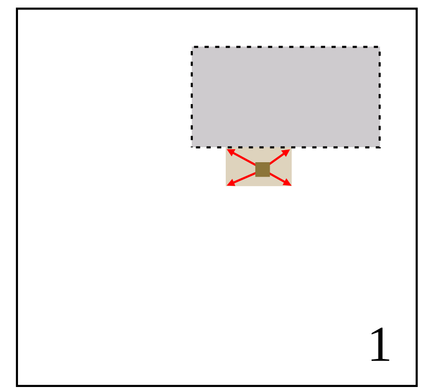


Simulation-based Expansion

The check whether or not an expansion is valid can be reduced to simulation on sets of points [EénM11].

Assume bad is defined as $e_1 < 2$ and we already covered $e_1 < -1$ with

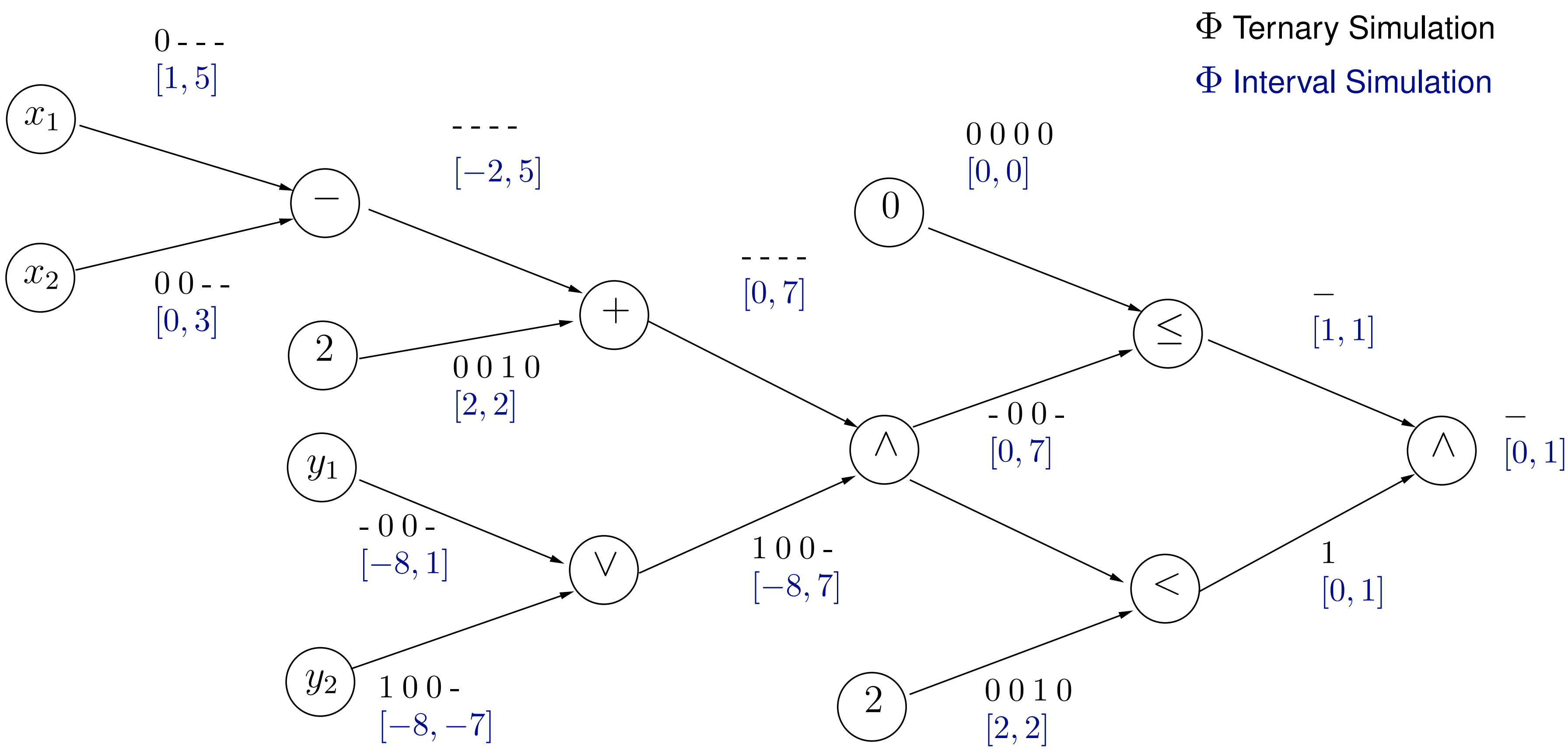
$$e_1 := (x_1 - x_2 + 2) \wedge (y_1 \vee y_2)$$



Then we may expand an ARU  to a larger ARU $\hat{a}$  if $e := \underbrace{(e_1 < 2)}_{\text{bad}} \wedge \neg \underbrace{(e_1 < -1)}_{\text{covered}}$ evaluates to **true** for all values in $\hat{a}$.

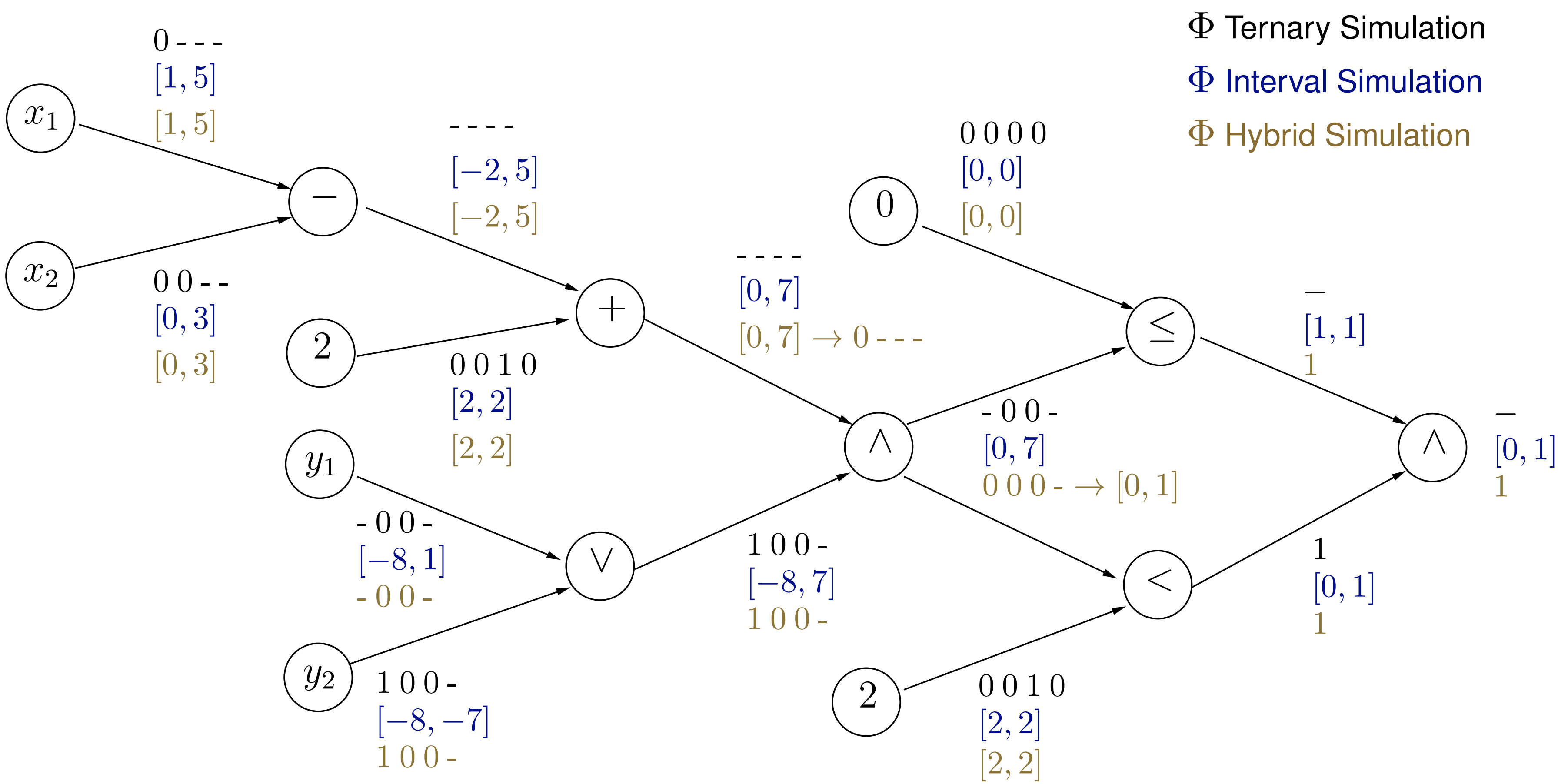
Interval Simulation

Let $\hat{a} := (1 \leq x_1 \leq 5) \wedge (0 \leq x_2 \leq 3) \wedge (y_1 \in \text{-00-}) \wedge (y_2 \in \text{100-})$

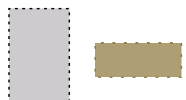
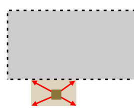


Hybrid Simulation

Let $\hat{a} := (1 \leq x_1 \leq 5) \wedge (0 \leq x_2 \leq 3) \wedge (y_1 \in \text{-00-}) \wedge (y_2 \in 100\text{-})$



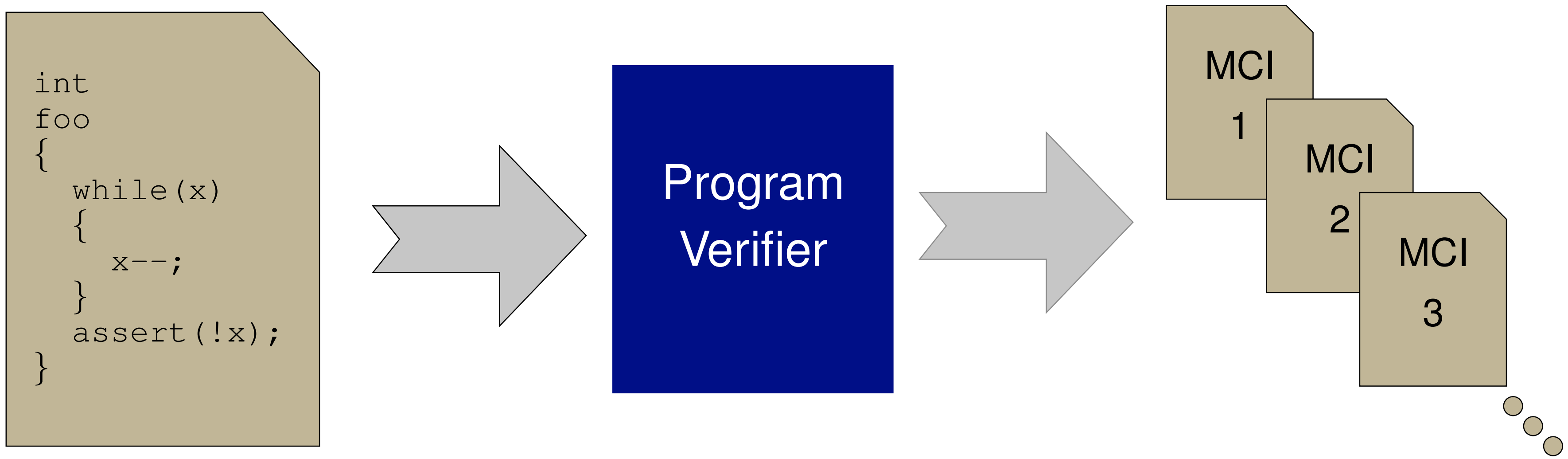
Property Directed Reachability for QF_BV

	Original Formulation	Polytopes [Welp13]	Hybrid
Atomic Reasoning Unit 	Boolean Cubes	Polytopes	Boolean Cubes and Polytopes
Expansion of Proof Obligations 	Ternary Simulation	Interval Simulation	Hybrid Simulation
Strengths	logic	arithmetic	arithmetic logic
Weaknesses	arithmetic	logic	-

Outline

1. Introduction
2. QF_BV Property Directed Reachability
3. Mixed Type Atomic Reasoning Units
4. **Experimental Results**
5. Summary

Experimental Setup



Benchmark Sets

Bitvector set of SV-Comp [Beye12]

```
int
foo(int n)
{
  int x = 1;
  while(1)
  {
    x += 2*n;
    assert(x);
  }
}
```

Mostly Logic Invariants

InvGen-Benchmarks [Gupt09]

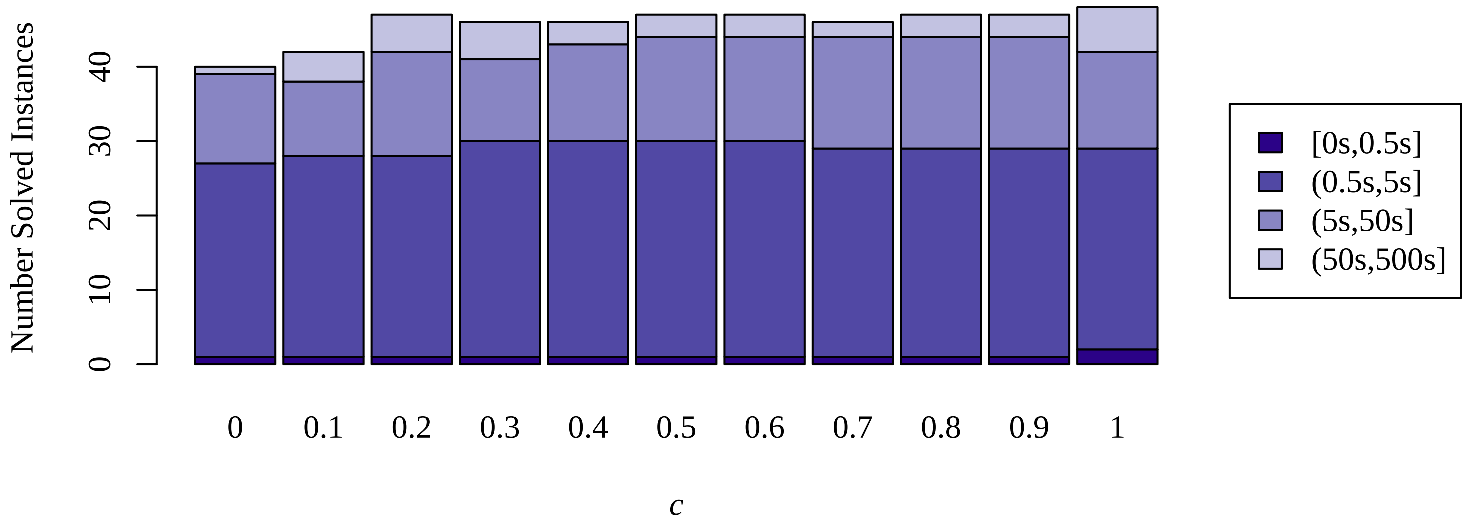
```
int
foo(int n)
{
  int x = 0;
  assume(n>=0);
  while(x < n)
  {
    x--;
  }
  assert(x <= n);
}
```

Mostly Arithmetic Invariants

Overall Performance

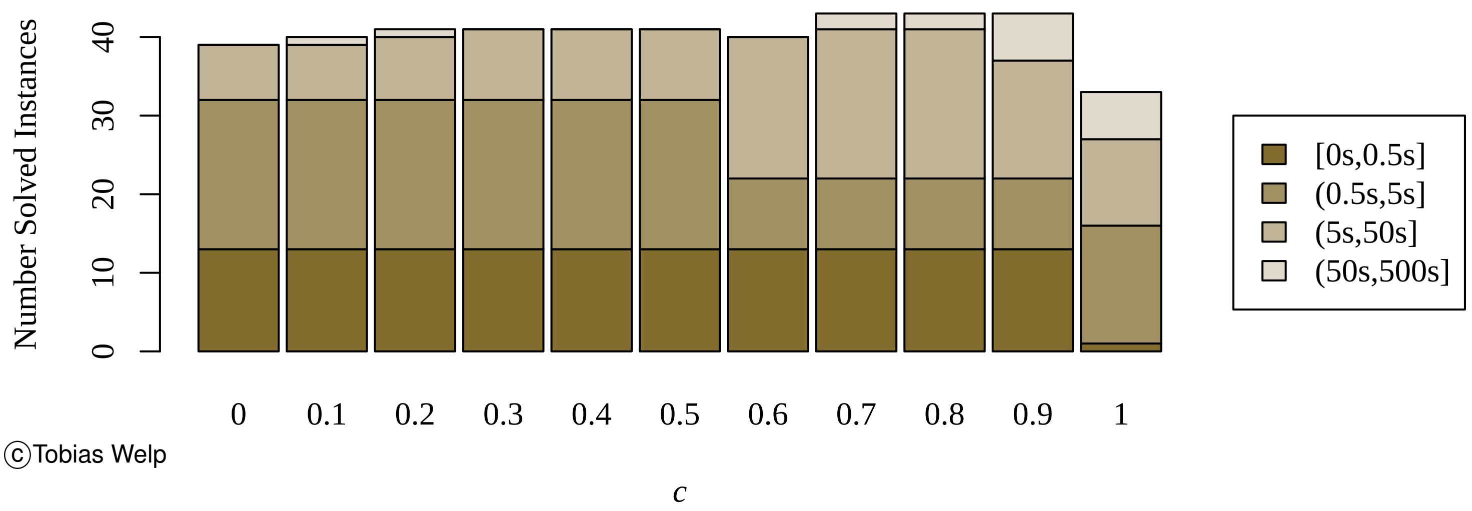
SV-COMP

[Beye12]



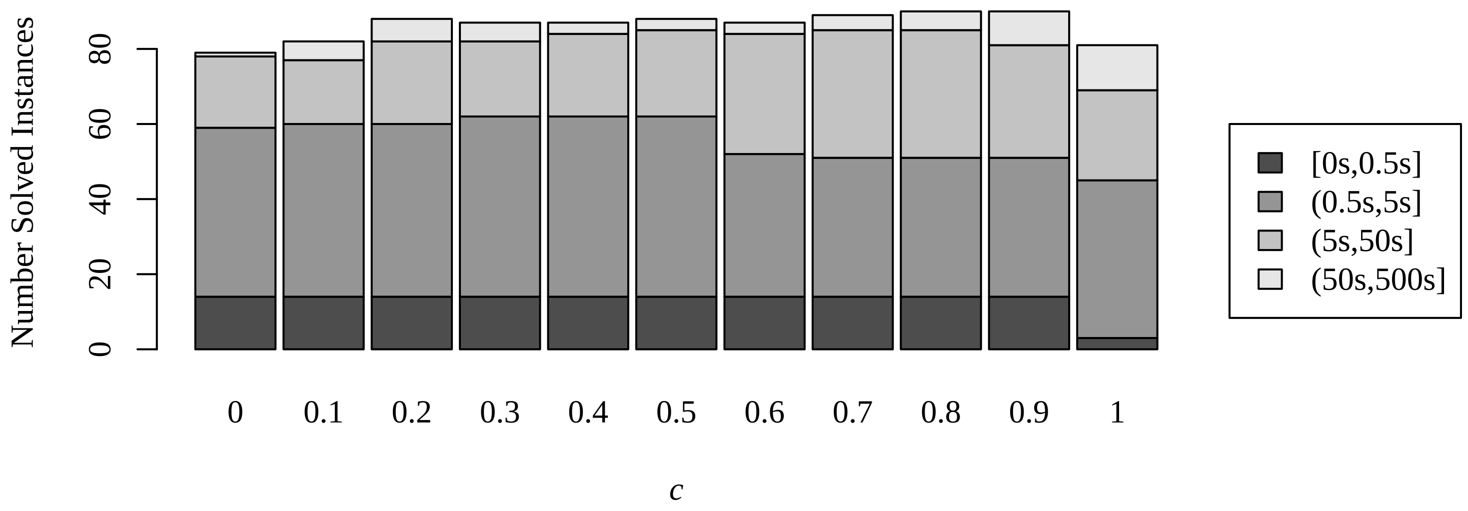
InvGen

[Gupt09]



Overall Performance

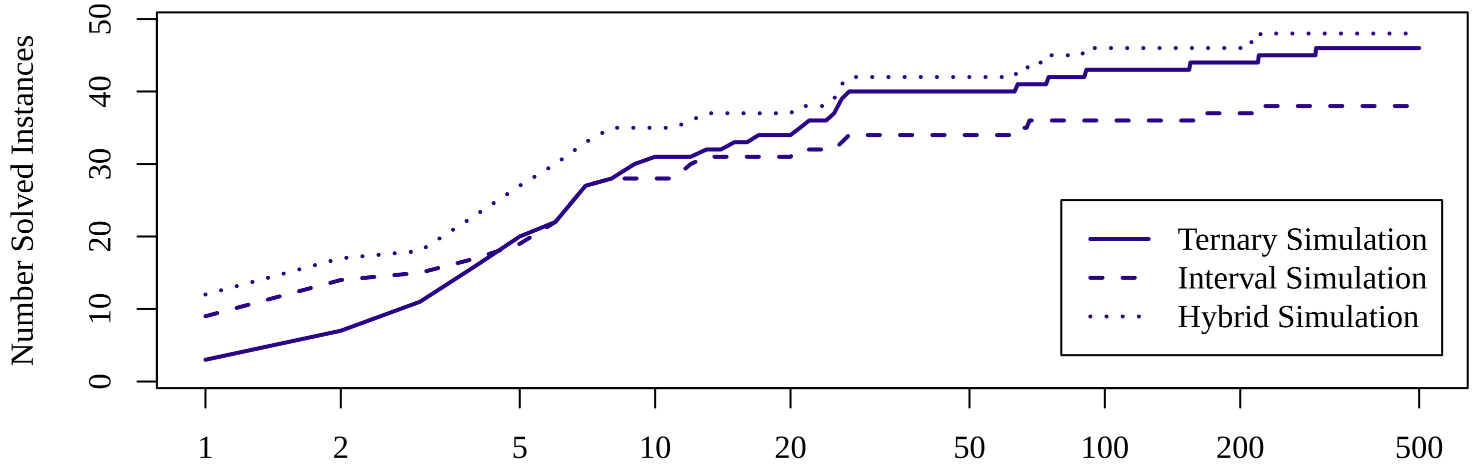
All



Impact of Simulation Type

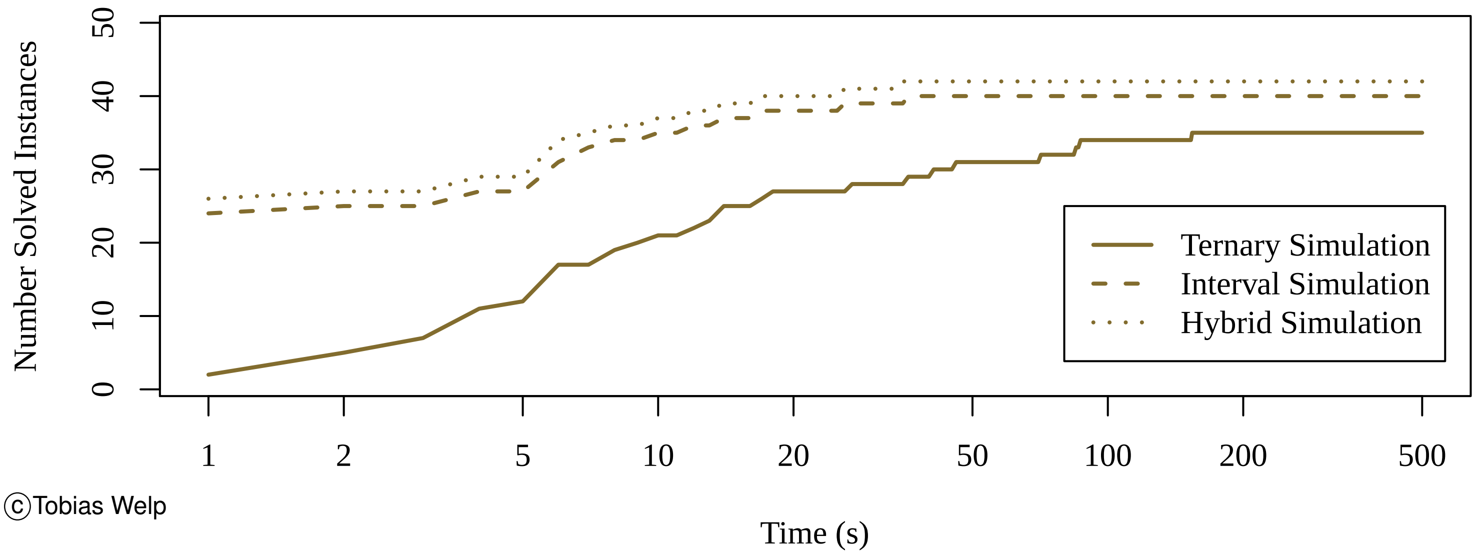
SV-COMP

[Beye12]



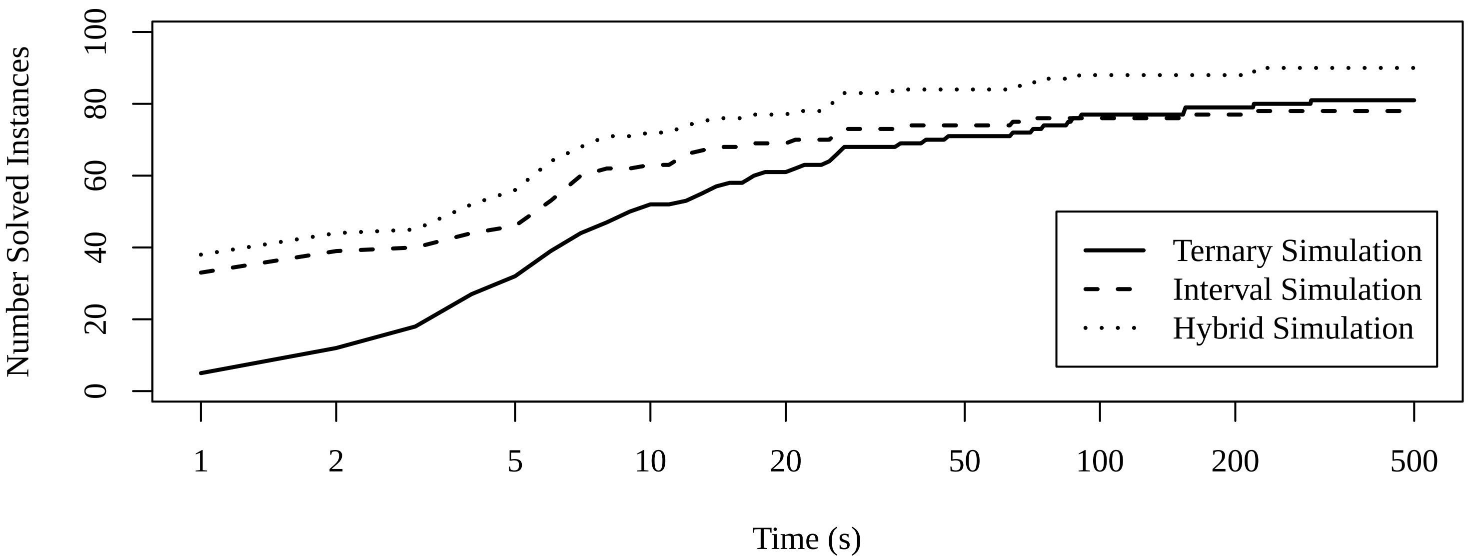
InvGen

[Gupt09]

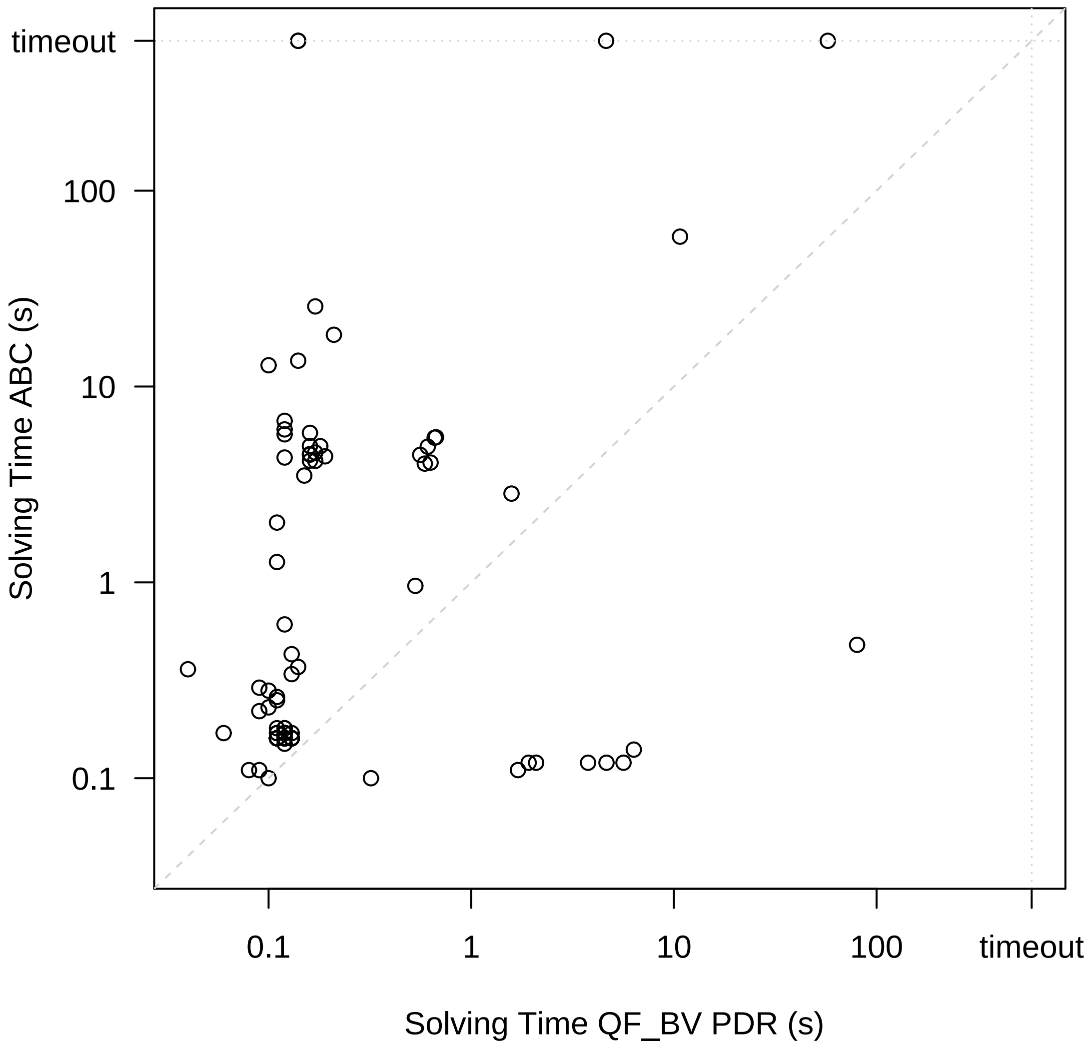


Impact of Simulation Type

All



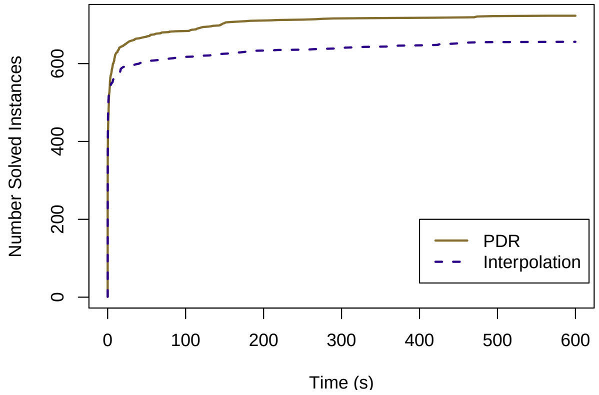
Comparison vs ABC PDR



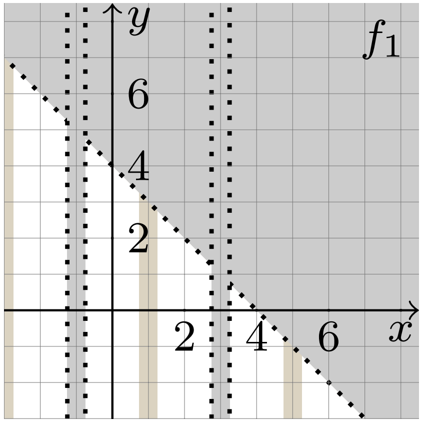
Outline

1. Introduction
2. Property Directed Reachability
3. Generalization of PDR to QF_BV
4. Experimental Results
5. **Summary**

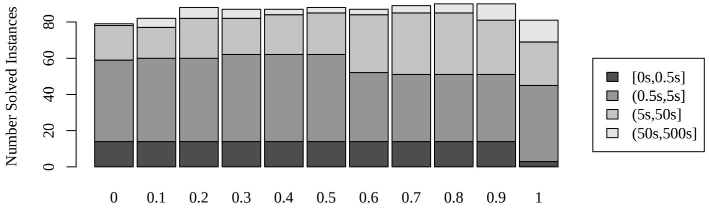
Summary



- PDR is an efficient algorithm for solving model checking problems.



- PDR with Boolean cubes performances poorly with arithmetic invariants.
- PDR with polytopes performances poorly with bit-level invariants.



- The hybrid formulation outperforms the pure versions.

Thank you!

for your

attention

twelp@berkeley.edu

References

[Some11] F. Somenzi, A. R. Bradley: *IC3: Where Monolithic and Incremental Meet.*, FMCAD 2011.

[Brad12] A. R. Bradley: *Understanding IC3.*, SAT 2012.

[Welp13] T. Welp: *QF_BV Model Checking with Property Directed Reachability.* DATE 2013.

[Brad11] A. R. Bradley, Z. Manna: *SAT-based model checking without unrolling.*, VMCAI 2011.

[EénM11] N. Eén, A. Mishchenko, R. Brayton: *Efficient Implementation of Property Directed Reachability*, FMCAD 2011.

References

[McMi03] K. L. McMillan: *Interpolation and SAT-based Model Checking*, CAV 2003.

[Kind12] R. Kindermann, T. Junttila, I. Niemelä: *SMT-based Induction Methods for Timed Systems*, FORMATS 2012.

[Hode12] K. Hoder, N. Bjørner: *Generalized Property Directed Reachability*, SAT 2012.

[Kloo13] J. Kloos, R. Majumdar, F. Niksic and R. Piskac: *Incremental, Inductive Coverability*, CAV 2013.

[Back13] J. Backes, M. Riedel: *Using Cubes of Non-state Variables With Property Directed Reachability*, DATE 2013.

References

[Beye12] D. Beyer: *Competition on Software Verification*, TACAS 2012.

[Gupt09] A. Gupta, A. Rybalchenko: *InvGen: An Efficient Invariant Generator*, CAV 2009.

