SAT-Based Model Checking with Interpolation

Orna Grumberg

Technion, Haifa, Israel

iPRA workshop
July 2014

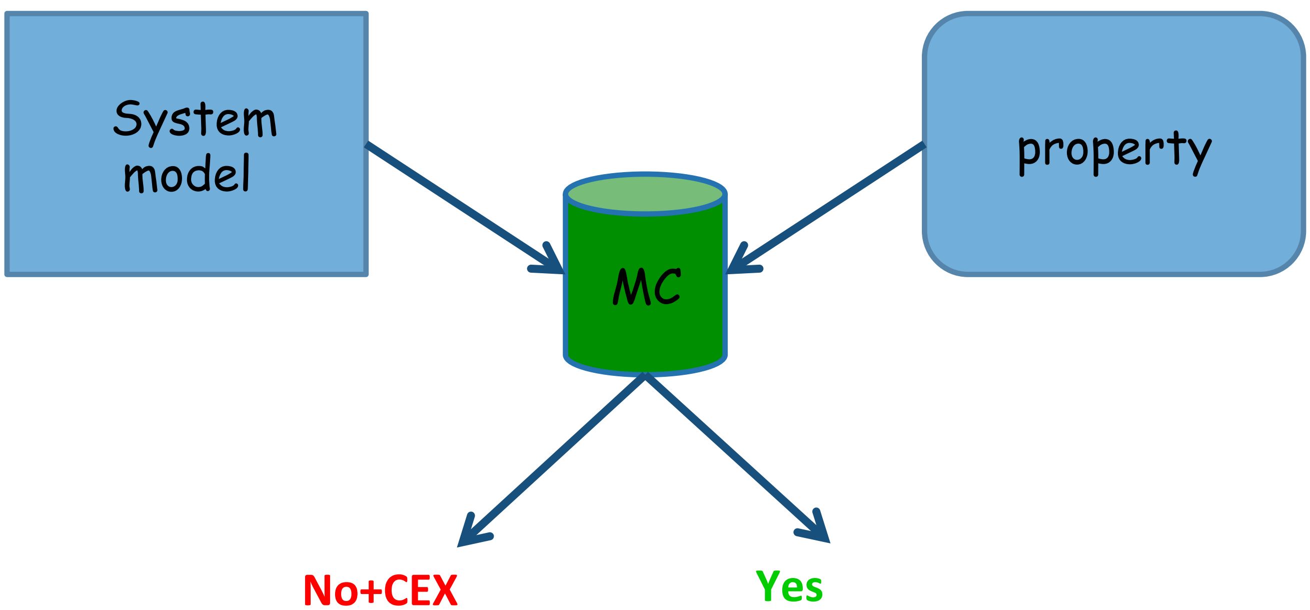
The beautiful slides are mostly borrowed from Yakir Vizel

Focus of the talk

- Interpolants in the propositional logic and their use in verification
- Interpolation for other logics is used, for instance, for software verification
  - Linear arithmetic, Reals, and others

Model Checking

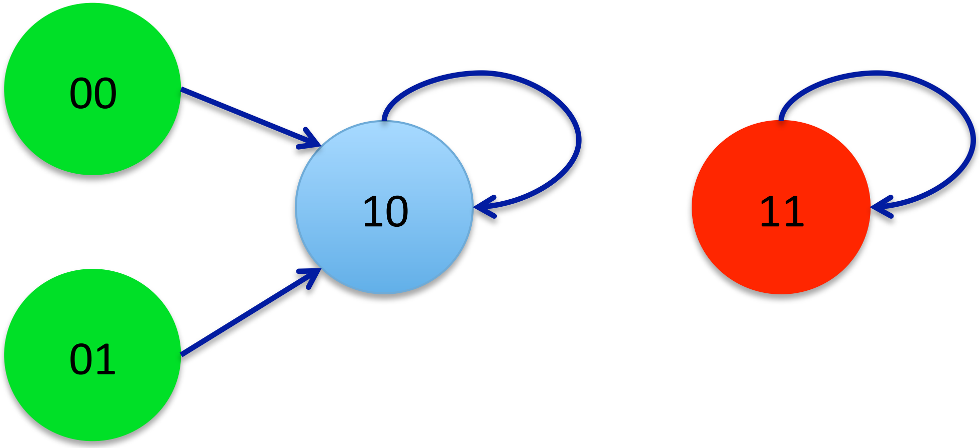
- Given a system and a specification, does the system satisfy the specification.



Outline

- Background on model checking
- SAT-based model checking with interpolation
- Model checking with interpolation sequence
- Model checking with backward and forward interpolations

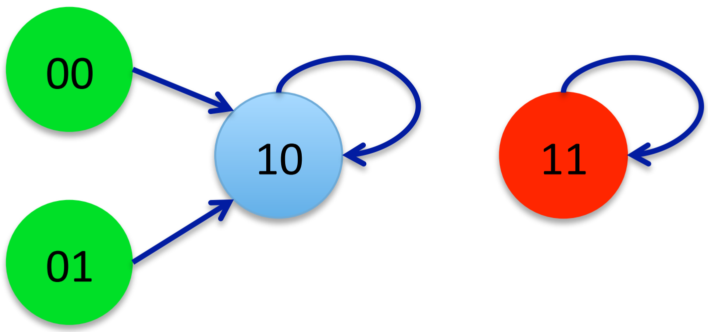
System model



Modeling

- System is modeled as (V,INIT,T), where:
  - V is a finite set of variables
    - **S – set of states – all valuations of V**
  - INIT $\in 2^V$ is the set of initial states
  - T $\subseteq 2^V \times 2^V$ is the set of transitions
- A safety property of the form *AG P*
  - "P holds in every reachable state of the system"
  - P is a formula over V

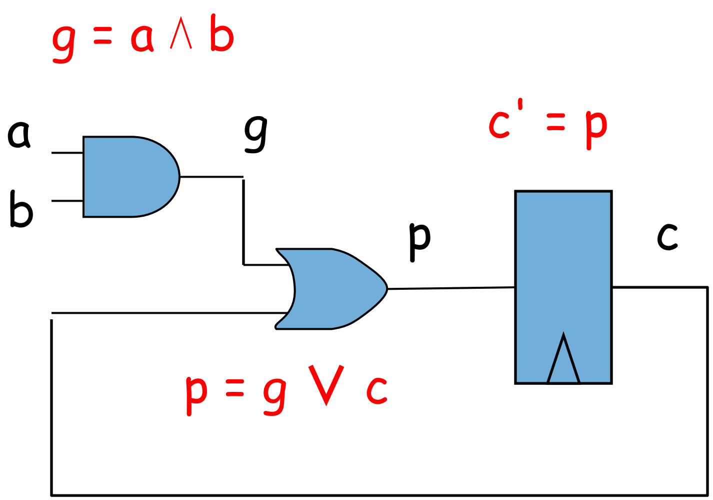
Translation to Propositional Formulas



- Four states:
  - Two Boolean variables: v_1 v_2
- INIT: $\neg v_1$
- T :
  - $v_1' = \neg v_1 \vee (v_1 \wedge v_2)$
  - $v_2' = (v_1 \wedge v_2)$
- P: $\neg v_1 \vee \neg v_2$ (Bad = $\neg P = v_1 \wedge v_2$)

Example

T is a conjunction of constraints, one per component.



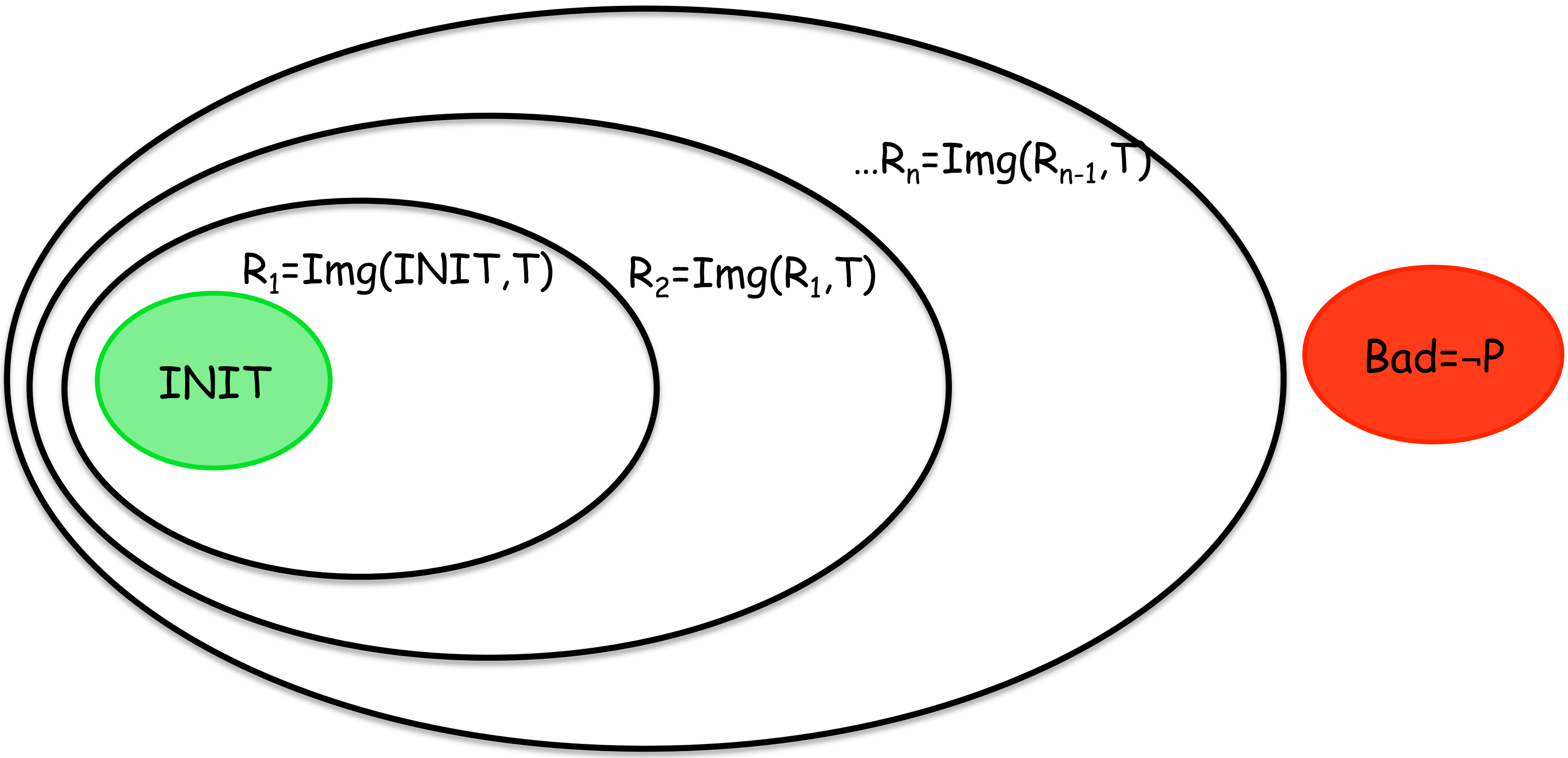
$T = \wedge\{$
$g = a \wedge b,$
$p = g \vee c,$
$c' = p$
$\}$

Reachability Analysis

- Problem definition:
  - Does the transition system have a finite run ending in a state satisfying $\neg P$?
  - More precisely, is there a sequence of states $s_0 \ldots s_k$ s.t.:
    - $s_0 \in I$ and $s_k \in \neg P$
    - for all $0 \leq i < k$, $(s_i, s_{i+1}) \in T$
- Using automata-theoretic methods, model checking safety properties reduces to reachability analysis.

Forward Reachability Analysis

Does AG P hold?



Termination when

- either a bad state satisfying $\neg \mathbf{p}$ is found
- or a **fixpoint** is reached: $\mathbf{R_j} \subseteq \cup_{i=0,j-1} \mathbf{R_i}$

Main limitation

The state explosion problem:

Space and time requirements grow with the size of the model

SAT-based Model Checking

Main idea

- Translate the model and the specification to propositional formulas
- Use efficient tools (SAT solvers) for solving the satisfiability problem
- At the beginning it was mainly used for finding CEXs

DPLL-style SAT solvers

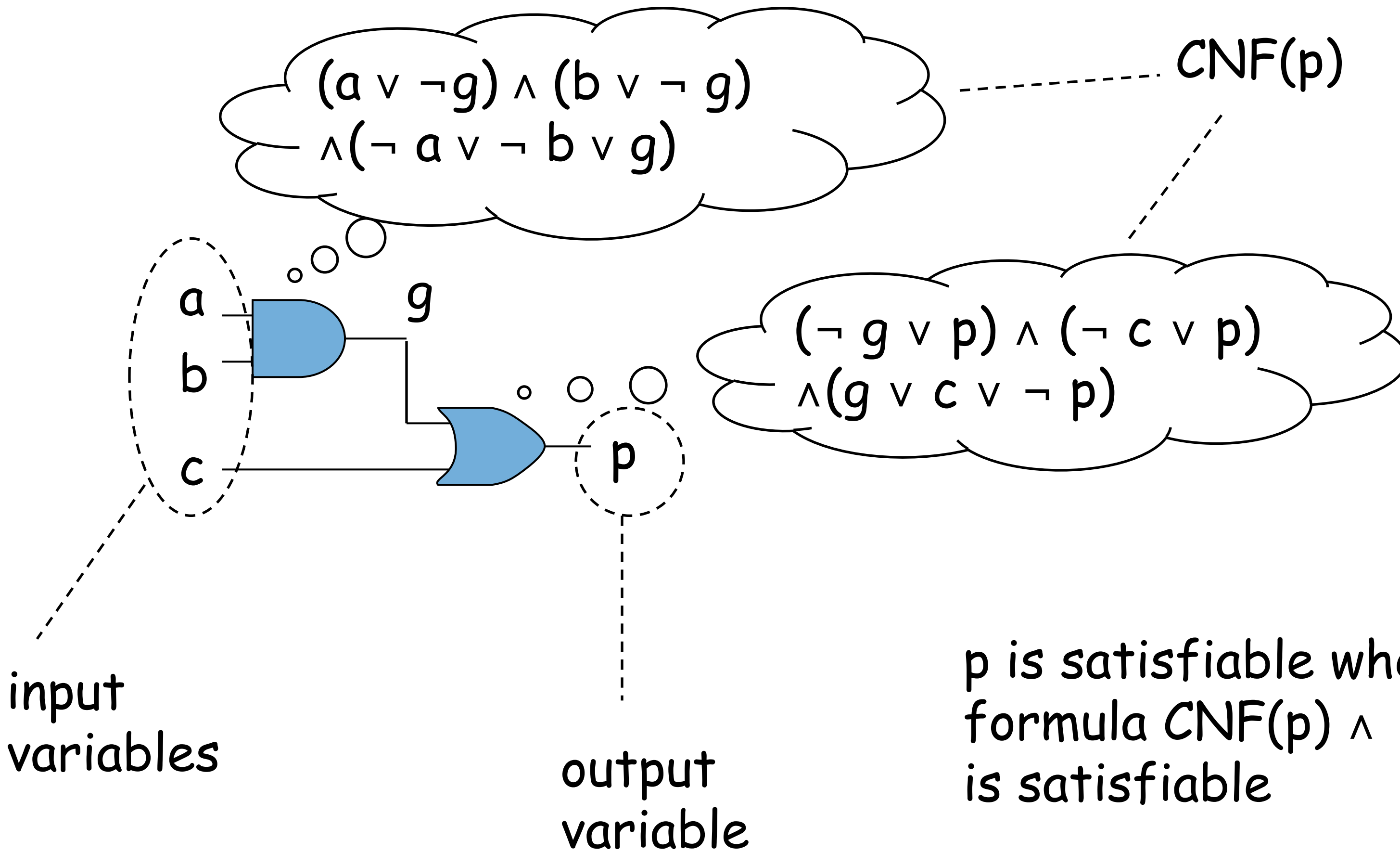
GRASP, CHAFF, MiniSAT, Glucose

- Objective:
  - Check whether a CNF formula is satisfiable or not
    - Either return a satisfying assignment
    - Or "UNSAT" and a refutation proof

- Approach:
  - Decision: Choose arbitrary variable+value for an unassigned variable
  - Propagate implications
  - Add conflict clauses to avoid rechecking assignments

Circuit to SAT

Can the circuit output be 1?



p is satisfiable when the formula CNF(p) ∧ p is satisfiable

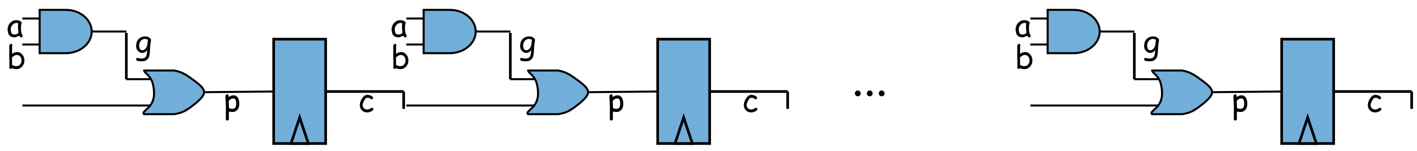
Bounded model checking

Biere, et al. TACAS99

- Unfold the model k times:

$$U = T^{\langle 0 \rangle} \wedge T^{\langle 1 \rangle} \wedge \ldots \wedge T^{\langle k-1 \rangle}$$

$I^{\langle 0 \rangle}$



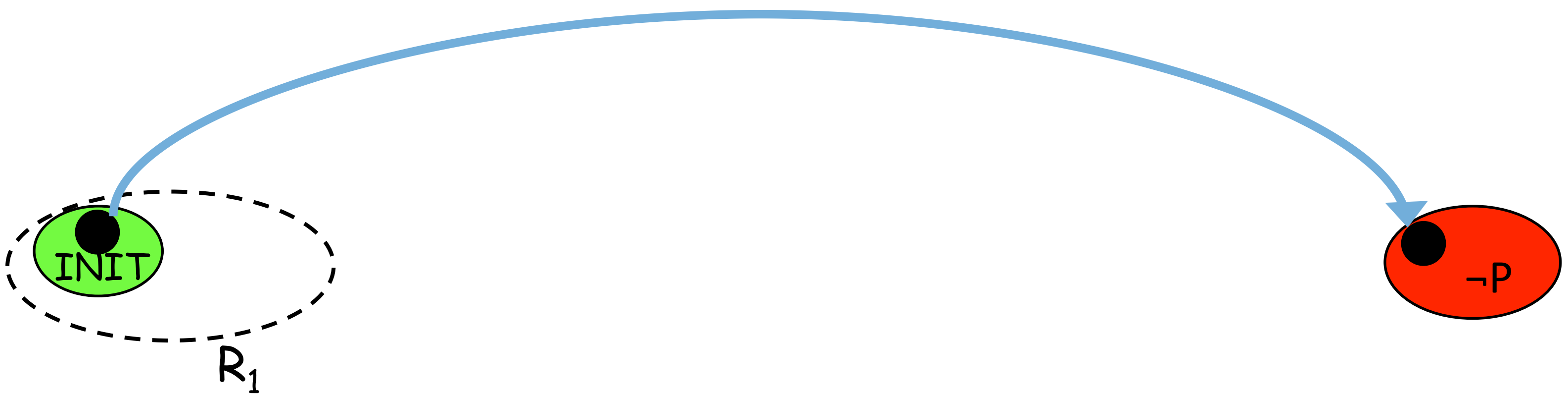
$\neg P^{\langle k \rangle}$

- Use SAT solver to check satisfiability of

$$I^{\langle 0 \rangle} \wedge U \wedge \neg P^{\langle k \rangle}$$

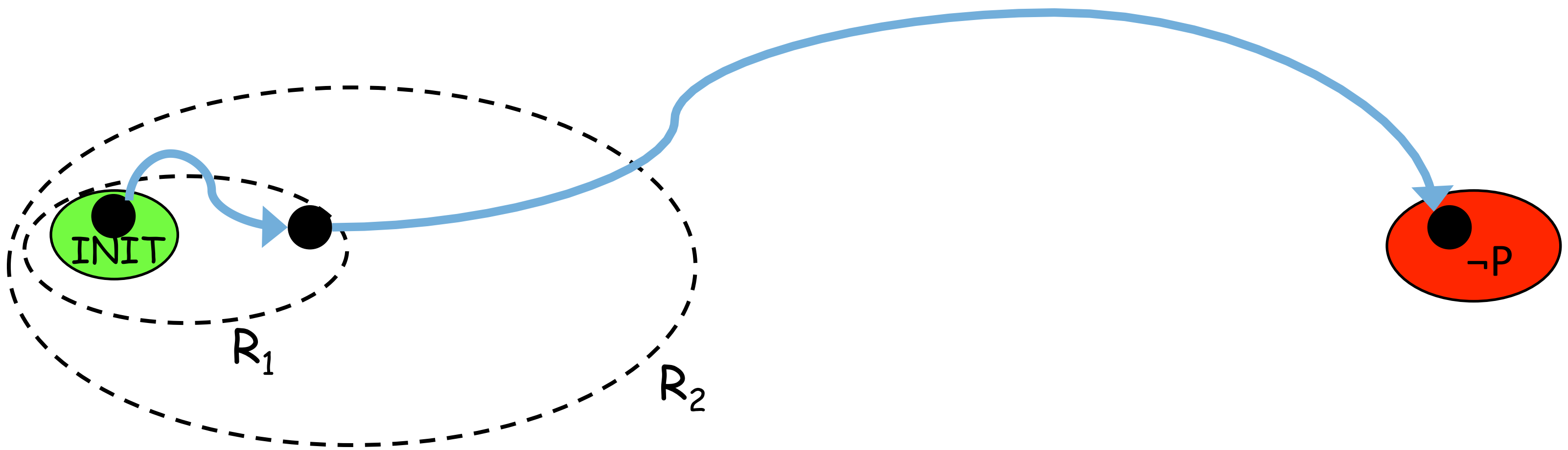
- If unsatisfiable:
  - property has no counterexample of length k
  - can produce a refutation proof

Bounded Model Checking



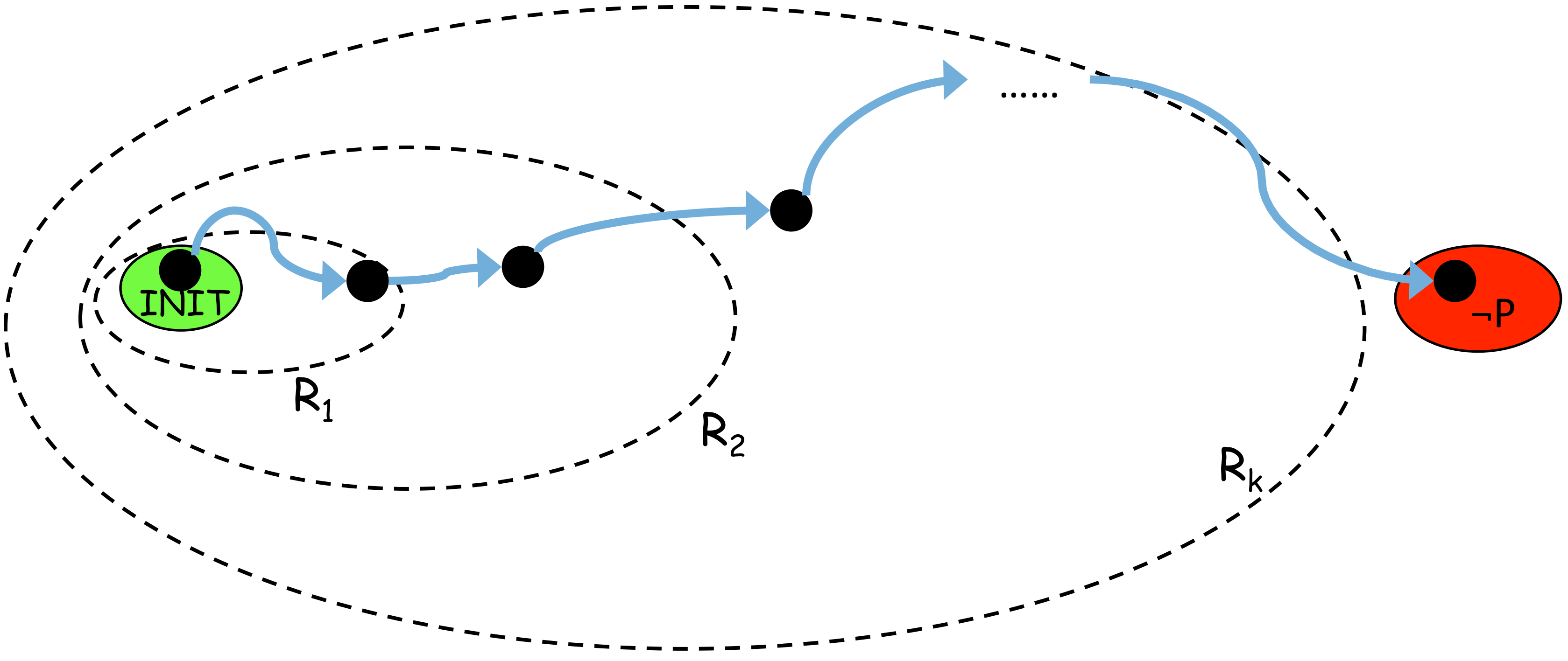
$$INIT(V^0) \wedge T(V^0, V^1) \wedge \neg P(V^1)$$

Bounded Model Checking



$$INIT(V^0) \wedge T(V^0, V^1) \wedge T(V^1, V^2) \wedge \neg P(V^2)$$

Bounded Model Checking



$$INIT(V^0) \wedge T(V^0,V^1) \wedge \ldots \wedge T(V^{k-1},V^k) \wedge \neg P(V^k)$$

Bounded Model Checking

Terminates

- with a counterexample or
- with time- or memory-out

The method is suitable for **falsification**, not verification

Outline

- Background on model checking
- SAT-based model checking with interpolation
- Model checking with interpolation sequence
- Model checking with backward and forward interpolations

SAT-Based Verification

unbounded model checking

- Uses BMC for falsification
- Simulates forward reachability analysis for verification
- Identifies a termination condition
  - all reachable states has been found: "fixpoint"

Interpolants

Craig 57

- Given an unsatisfiable pair (A,B) of propositional formulas
  - $A(X,Y) \wedge B(Y,Z)$ is unsatisfiable
- There exists a formula I such that:
  - $A \Rightarrow I$
  - $I \wedge B$ is unsatisfiable
  - I is over Y, the common variables of A and B

Interpolation (cont.)

Interpolants from proofs

- When $A \wedge B$ is unsatisfiable, SAT solvers return a proof of unsatisfiability in the form of a resolution graph

- Given a resolution graph, **I** can be derived in linear time

Pudlak,Krajicek 97, McMillan 03

ITP - Interpolation-based MC

McMillan, CAV 2003

A | B

$$\underbrace{INIT(V) \wedge T(V,V^1)}_{A} \wedge \underbrace{T(V^1,V^2) \wedge T(V^2,V^3) \wedge (\neg P(V^1) \vee \ldots \vee \neg P(V^3))}_{B}$$

I

- I over-approximates the states reachable from INIT in one transition
  - It satisfies P and cannot reach a bad state in two transitions or less

ITP - Interpolation-based MC

McMillan, CAV 2003

A | B

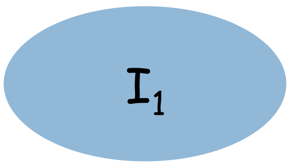
$$\underbrace{I(V) \wedge T(V,V^1)}_{A} \;\wedge\; \underbrace{T(V^1,V^2) \wedge T(V^2,V^3) \wedge (\neg P(V^1) \vee \ldots \vee \neg P(V^3))}_{B}$$

I'

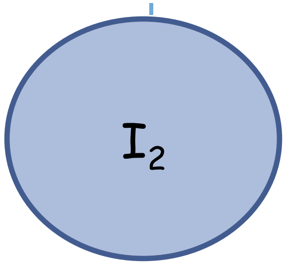
- I is fed back to the formula
  - A new interpolant I' is computed
  - Iterative process

Using Interpolation (i=1)

$$INIT(V_0) \wedge T(V_0, V_1) \wedge \neg p(V_1)$$



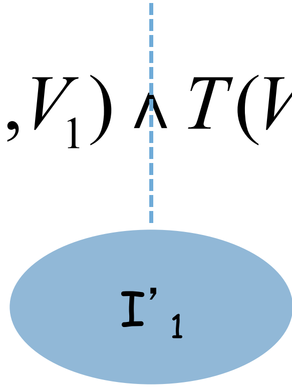
$$I_1(V_0) \wedge T(V_0, V_1) \wedge \neg p(V_1)$$



$$I_2(V_0) \wedge T(V_0, V_1) \wedge \neg p(V_1)$$

Using Interpolation (i=2)

$$INIT(V_0) \wedge T(V_0,V_1) \wedge T(V_1,V_2) \wedge (\neg q(V_1) \vee \neg q(V_2))$$



$$I_1'(V_0) \wedge T(V_0,V_1) \wedge T(V_1,V_2) \wedge (\neg q(V_1) \vee \neg q(V_2))$$

.

.

.

$$I_k'(V_0) \wedge T(V_0,V_1) \wedge T(V_1,V_2) \wedge (\neg q(V_1) \vee \neg q(V_2))$$

- In ITP, short BMC formulas can prove the nonexistence of long CEXs
  - INIT is replaced by I_k which over-approximates S_k
- If a satisfying assignment is found, the counterexample might be spurious
  - Since INIT is over-approximated
- Increase k and start with the original INIT

- A fixpoint is checked whenever a new interpolant is computed

- For iteration i, every new interpolant is checked for inclusion in all previously computed interpolants for the same i
  - $I_n \Rightarrow INIT \vee V_{j=1,n-1}\, I_j$

- In ITP, a computed interpolant is fed back into the BMC problem
- BMC problem is solved with a SAT solver

Problems:

1. "Big" interpolant causes the BMC problem to be hard to solve
2. Non-CNF interpolant needs to be translated to CNF

Outline

- Background on model checking
- SAT-based model checking with interpolation
- Model checking with interpolation sequence
- Model checking with backward and forward interpolations

Interpolation-Sequence

- If $A_1 \wedge A_2 \wedge \dots \wedge A_k$ is unsatisfiable, then there exists an *interpolation-sequence* $I_0, I_1, \dots, I_{k+1}$ for $(A_1, \dots, A_k)$ such that:

$$I_0 = T \quad \text{and} \quad I_{k+1} = F$$

$$I_j \wedge A_{j+1} \Rightarrow I_{j+1}$$

  I_j - over common variables of $A_1, \dots, A_j$ and $A_{j+1}, \dots, A_k$

- Each I_j can be computed as the interpolant of $A = A_1 \wedge \dots \wedge A_j$ and $B = A_{j+1} \wedge \dots \wedge A_k$
  - **All I_j's should be computed on the same resolution graph**

Reachability with Interpolation-Sequence

Vizel , Grumberg, FMCAD 2009

- Unsatisfiable BMC formula partitioned in the following manner:

$$\underbrace{INIT(V^0) \wedge T(V^0,V^1)}_{A_1} \wedge \underbrace{T(V^1,V^2)}_{A_2} \wedge \underbrace{T(V^2,V^3)}_{A_3} \wedge \cdots \wedge \underbrace{T(V^{k-1},V^k)}_{A_k} \wedge \underbrace{\neg P(V^k)}_{A_{k+1}}$$

$T \quad I_1 \quad I_2 \quad I_3 \quad \ldots \quad I_k \quad F$

Reachability with Interpolation-Sequence

$$\overbrace{INIT(V^0) \wedge T(V^0,V^1)}^{A_1} \;\big|\; \wedge \overbrace{T(V^1,V^2)}^{A_2} \;\big|\; \wedge \overbrace{T(V^2,V^3)}^{A_3} \;\big|\; \wedge \cdots \wedge \overbrace{T(V^{k-1},V^k)}^{A_k} \;\big|\; \overbrace{\wedge \neg P(V^k)}^{A_{k+1}}$$

$$T \qquad I_1 \qquad I_2 \qquad I_3 \qquad I_k \qquad F$$

$I_0=T$ and $I_{k+1}=F$

$I_j \wedge A_{j+1} \Rightarrow I_{j+1}$

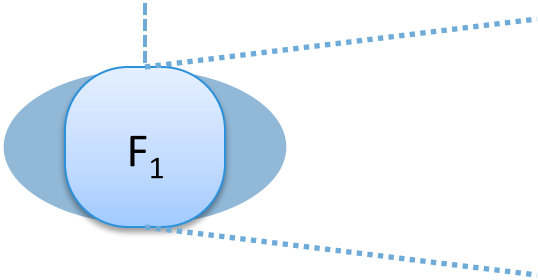
I_j - over common variables of $A_1, \ldots, A_j$ and $A_{j+1}, \ldots, A_k$

Reachability with Interpolation-Sequence

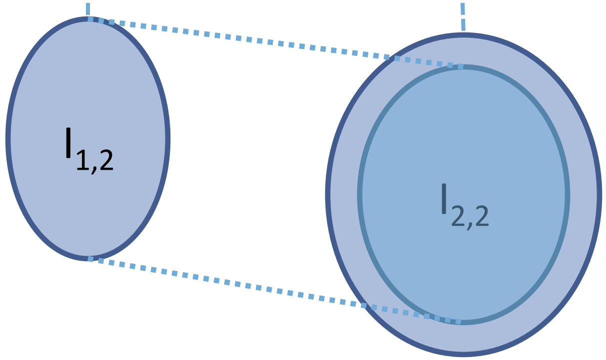
- Compute a sequence of reachable states from BMC formulas
  - Forward Sequence: $\langle F_0, F_1, \ldots, F_n \rangle$
- Sequence is over-approximated
  - $F_i(V) \wedge T(V,V') \Rightarrow F_{i+1}(V')$
  - $F_i \Rightarrow P$
- Integrated into the BMC loop to detect termination

Using Interpolation-Sequence

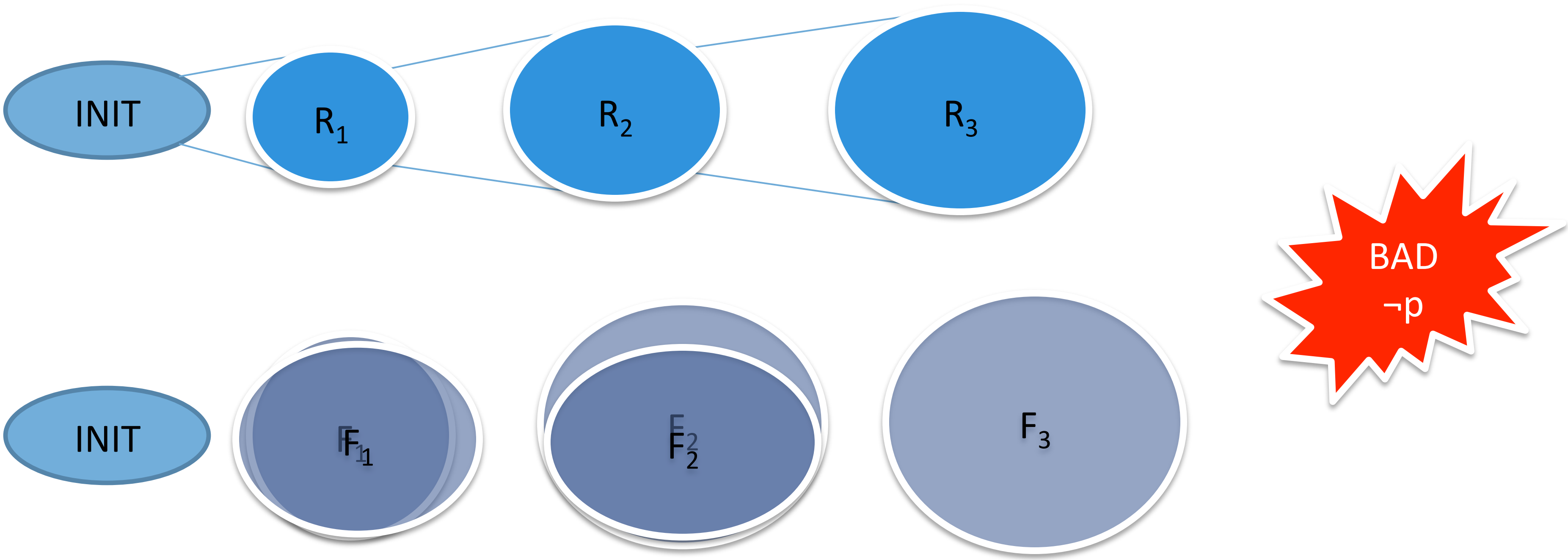
$$INIT(V_0) \wedge T(V_0,V_1) \wedge \neg p(V_1)$$



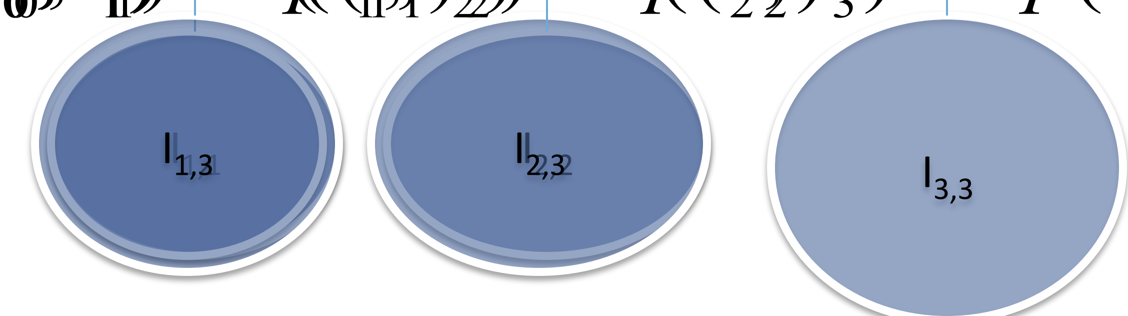
$$INIT(V_0) \wedge T(V_0,V_1) \wedge T(V_1,V_2) \wedge \neg p(V_2)$$



The Analogy to Forward Reachability Analysis



$$INIT(V_0) \wedge T(V_0,V_1) \wedge T(V_1,V_2) \wedge T(V_2,V_3) \wedge \neg p(V_3)$$



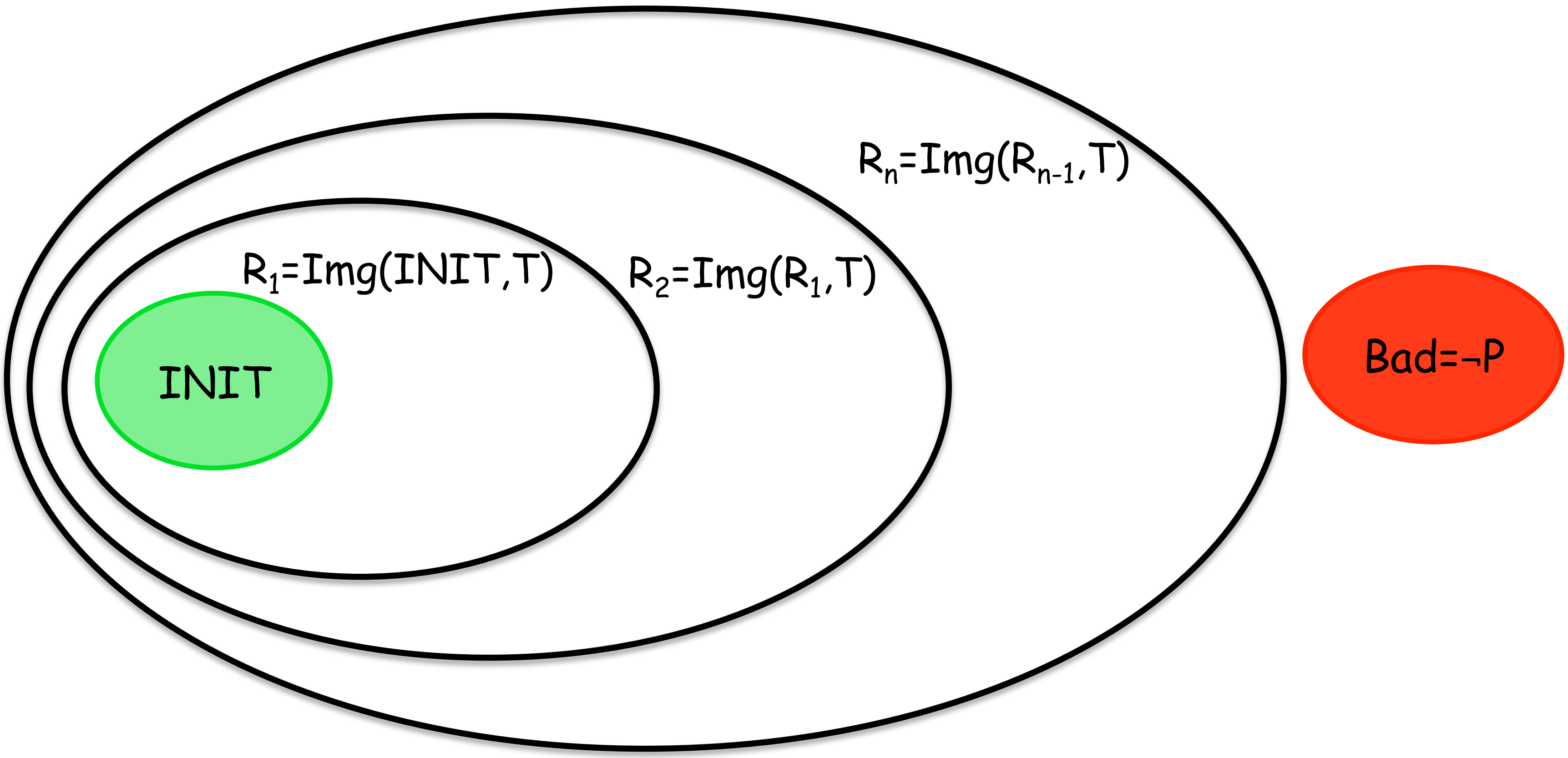
Checking if a "fixpoint" has been reached

- $F_n \Rightarrow \vee_{j=1,n-1} F_j$
- Similar to checking fixpoint in forward reachability analysis :
  $R_k \subseteq \cup_{j=1,k-1} R_j$
- But here we check inclusion for every $2 \leq k \leq n$
  - No monotonicity because of the approximation
- "Fixpoint" is checked with a SAT solver

Outline

- Background on model checking
- SAT-based model checking with interpolation
- Model checking with interpolation sequence
- Model checking with backward and forward interpolations

Forward Reachability Analysis



Interpolants

- Given an unsatisfiable pair (A,B) of propositional formulas
- Then, there exists a formula I such that:
  - $A \Rightarrow I$
  - $I \wedge B$ is unsatisfiable
  - I is over the common variables of A and B
- $I = Itp(A,B)$

Approximated Forward Reachability

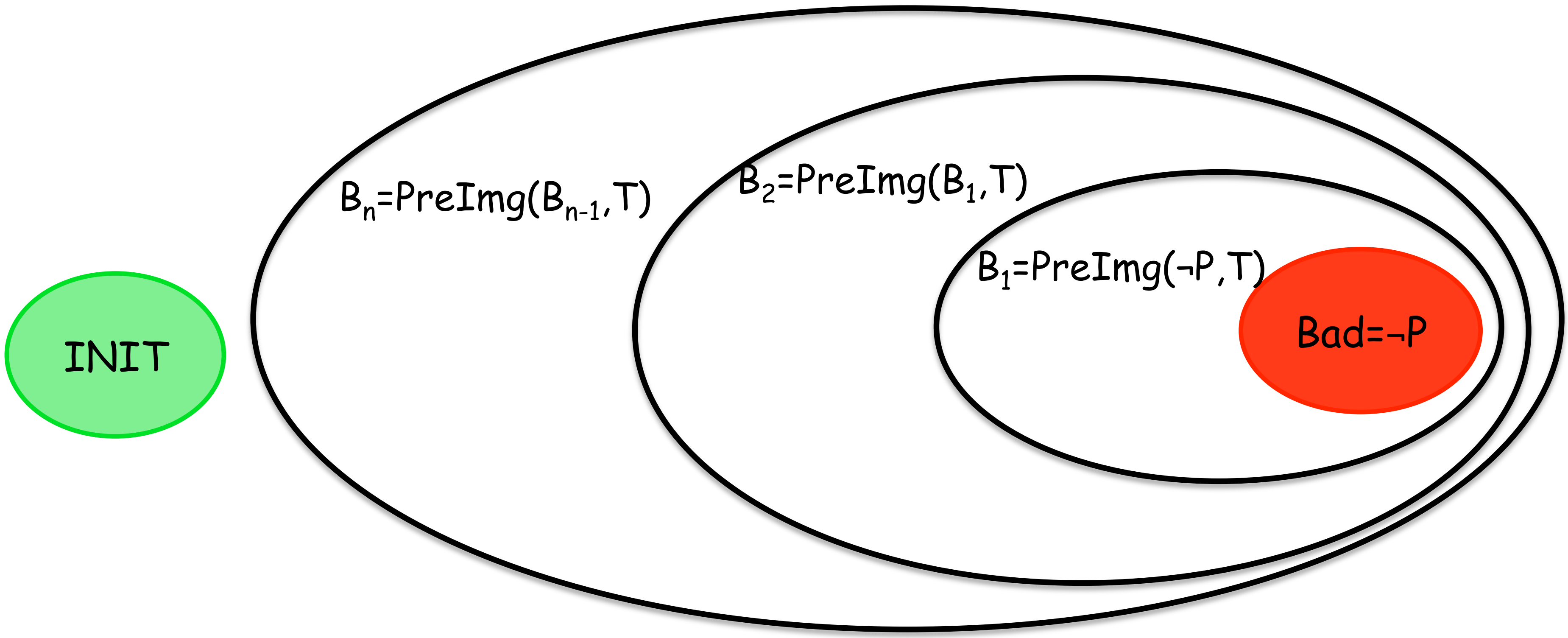
- F(V) – a set of states
- For the unsatisfiable formula $F(V) \wedge T(V,V') \wedge \neg P(V')$, define:

$$A = F(V) \wedge T(V,V')$$
$$B = \neg P(V')$$

- Approximated forward reachability

Backward Reachability Analysis

Does AGp hold?



Duality In a SAT Query

- $INIT(V) \wedge T(V,V') \wedge \neg P(V')$

Do we reach the bad states?

- We tend to read it "Forward"
  - From left to right

Duality In a SAT Query

- $INIT(V) \land T(V,V') \land \neg P(V')$

Do we reach the initial states?

- We tend to read it "Forward"
  - From left to right
- We can also read it "Backward"
  - From right to left
  - Does the pre-image of the bad states intersect the initial states

Approximated Backward Reachability

- B(V) – a set of states
- For the unsatisfiable formula
  $INIT(V) \wedge T(V,V') \wedge B(V')$, define:

$$A = T(V,V') \wedge B(V')$$

$$B = INIT(V)$$

- Approximated backward reachability

Dual Approximated Reachability (DAR)

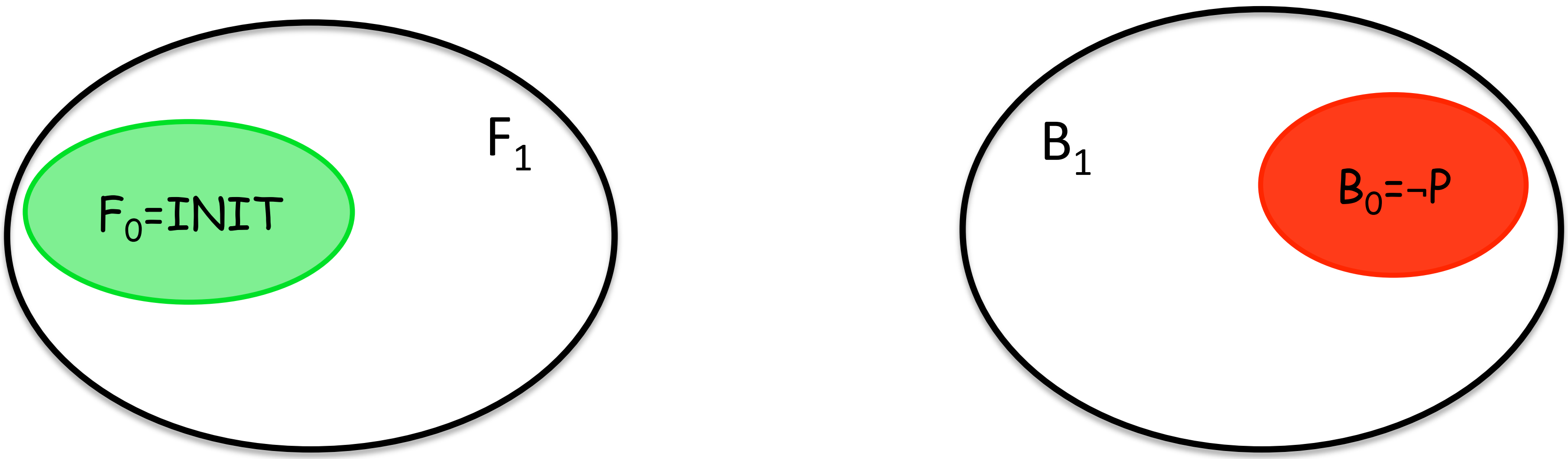
(Vizel, Grumberg and Shoham,TACAS 2013)

- Compute two sequences of reachable states
  - Forward Sequence: $\langle F_0, F_1, \ldots, F_n \rangle$
  - Backward Sequence: $\langle B_0, B_1, \ldots, B_n \rangle$
- Sequences are over-approximations
  - For the forward sequence:
    - $F_i(V) \land T(V,V') \Rightarrow F_{i+1}(V')$
    - $F_i \Rightarrow P$
  - For the backward sequence
    - $B_{i+1}(V) \Leftarrow T(V,V') \land B_i(V')$
    - $B_i \Rightarrow \neg INIT$

Dual Approximated Reachability (DAR)

- Two main phases during the computation
  - Local Strengthening
    - No unrolling
  - Global Strengthening
    - Limited unrolling
    - In case the Local Strengthening fails

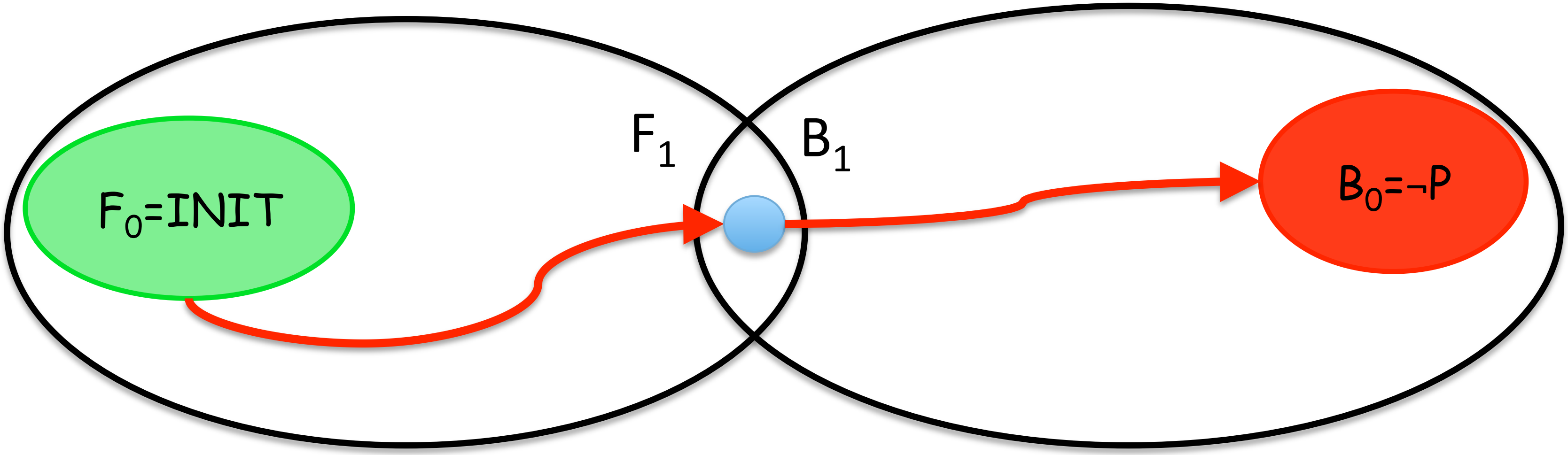
Dual Approximated Reachability



Local Strengthening

What if F_1 and B_1 intersect each other?

There might be a counterexample

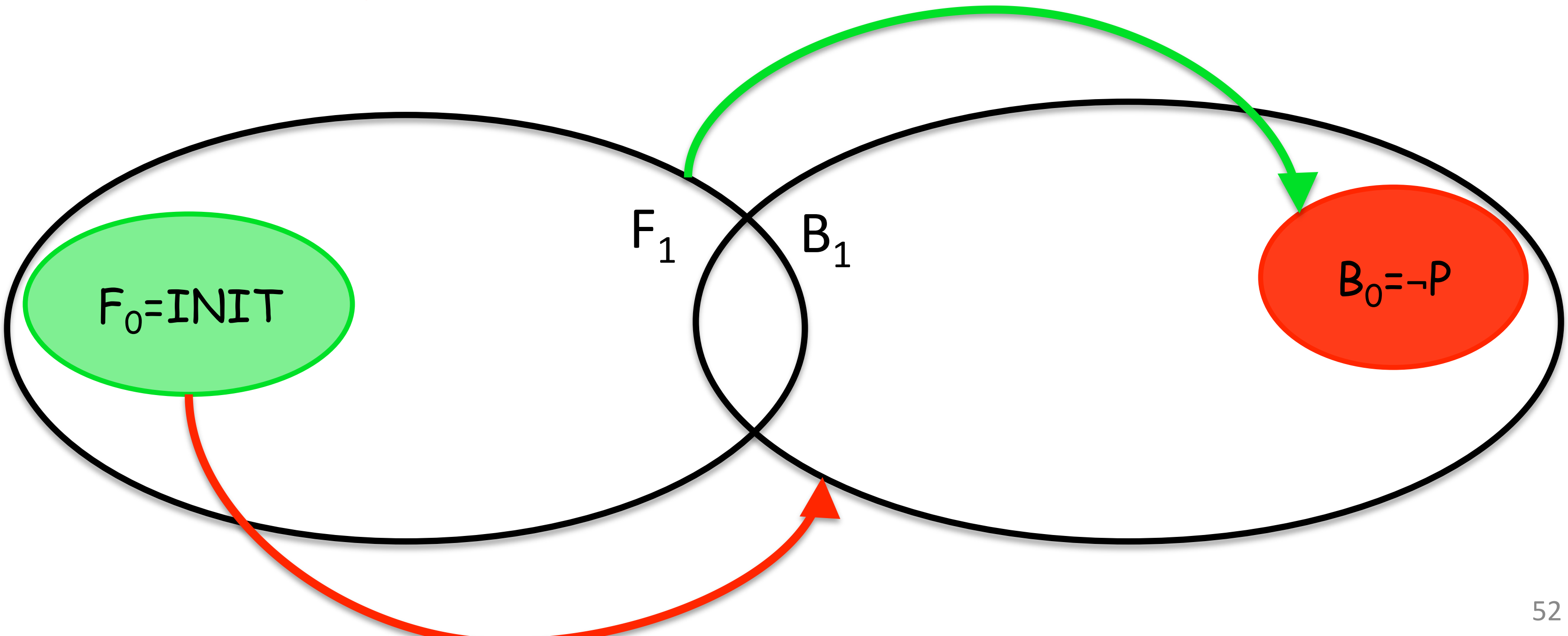


Local Strengthening

What if F_1 and B_1 intersect each other?

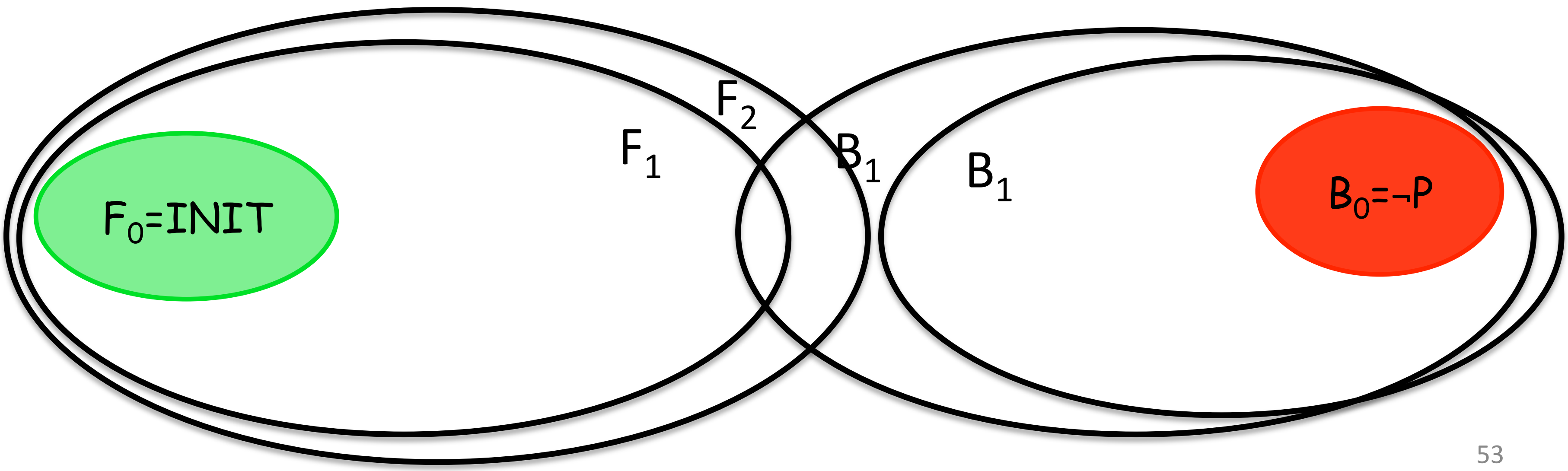
$F_1(V) \wedge T(V,V') \wedge B_0(V')$

$F_0(V) \wedge T(V,V') \wedge B_1(V')$



Local Strengthening

- Compute forward and backward interpolants
  - F_2 is the forward interpolant
  - Backward interpolant strengthens the already existing B_1



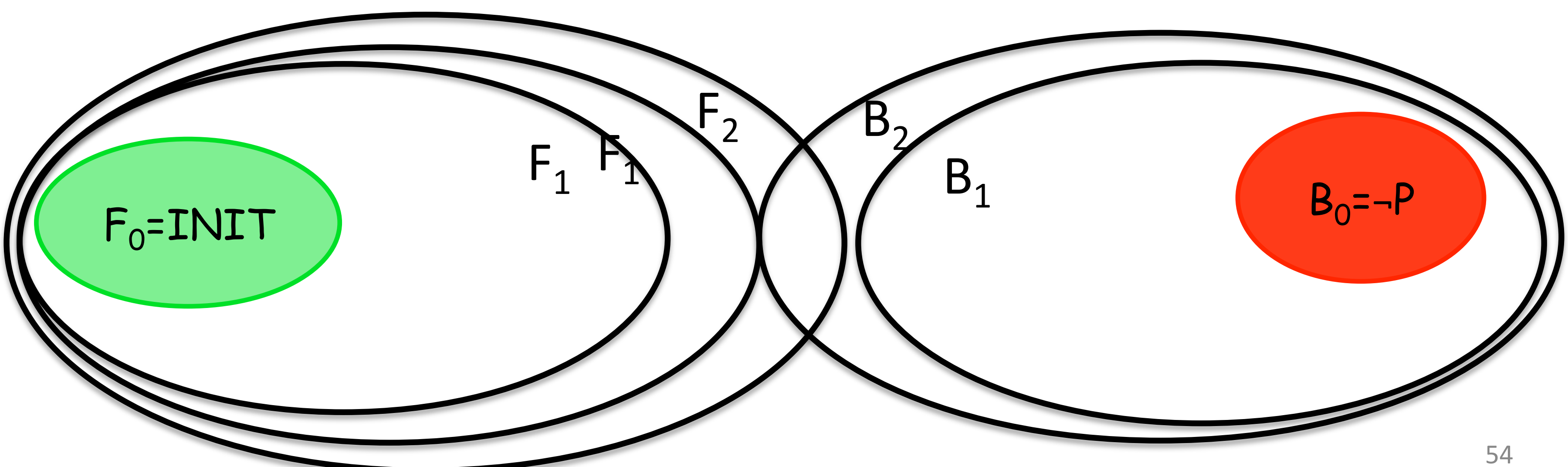
Local Strengthening

B A A B

$INIT(V) \wedge T(V,V') \wedge B_1(V')$

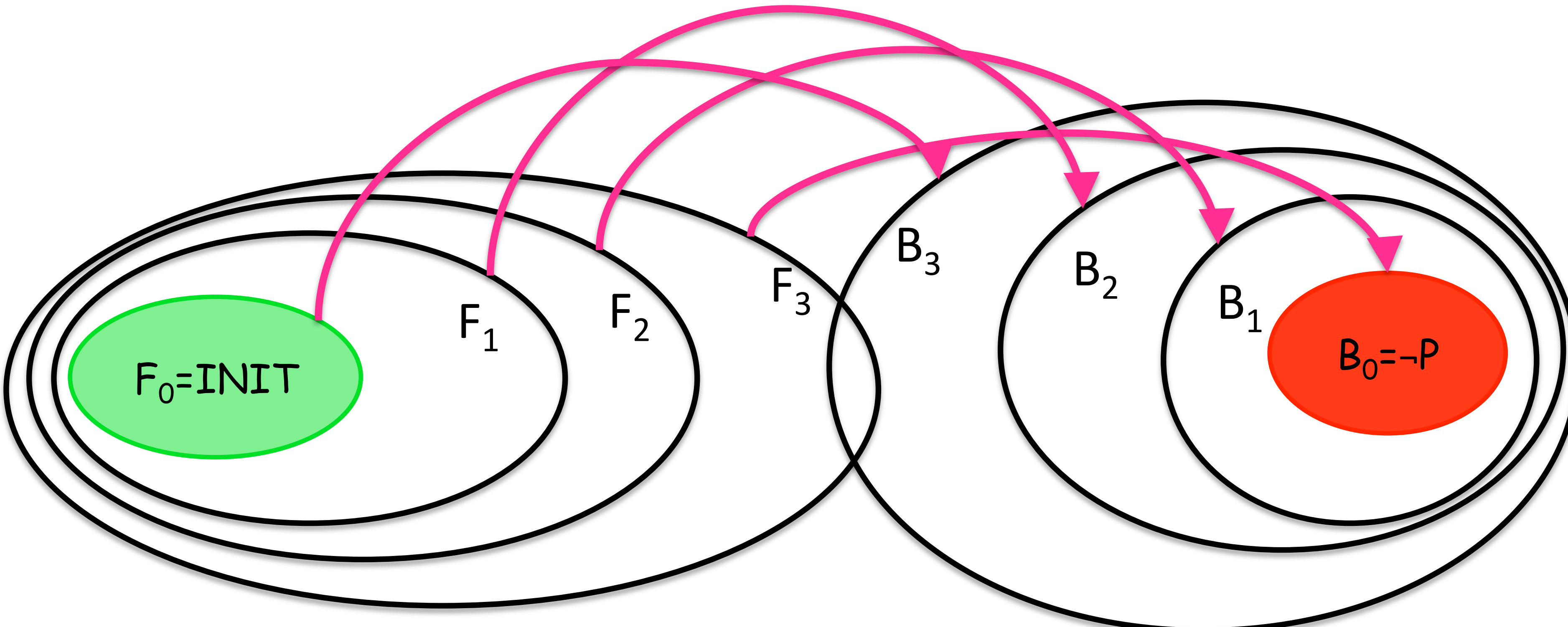
Must **be UnSAT**

- Compute forward and backward interpolants
  - B_2 is the backward interpolant
  - F'_1 is strengthening the already existing F_1



Local Strengthening Fails

$F_0(V) \land T(V,V') \land B_3(V')$

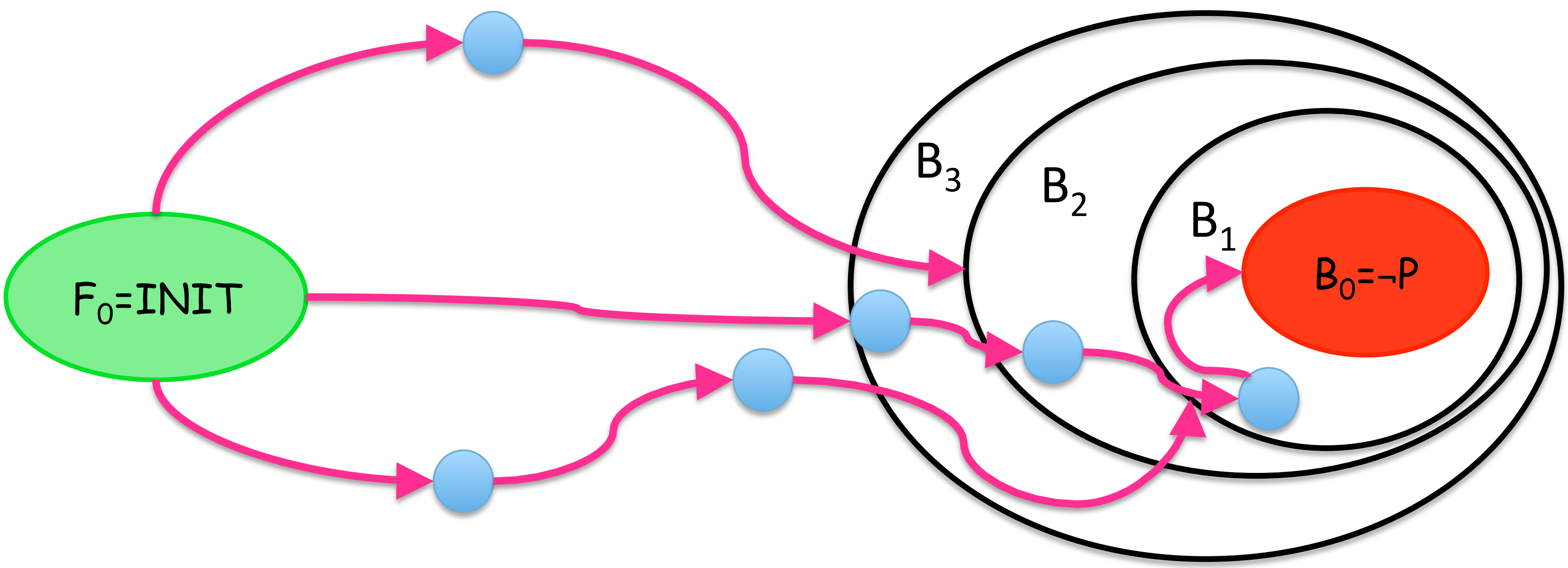


Global Strengthening

- Apply unrolling gradually
  - Start from the initial states
  - Try to reach the backward sequence using an increasing number of T's

Global Strengthening

$F_0(V) \wedge T(V,V') \wedge T(V',V'') \wedge T(V'',V''') \wedge B_2(V''') \; B_1(V'''') \; P(V'''')$



Global Strengthening

Interpolation sequence for UNSAT formula

$$\underbrace{F_0(V) \wedge T(V,V')}_{A_1} \wedge \underbrace{T(V',V'')}_{A_2} \wedge \underbrace{T(V'',V''')}_{A_3} \wedge \underbrace{B_1(V''')}_{A_4}$$

$T \qquad I_1 \qquad I_2 \qquad I_3 \qquad F$

Global Strengthening

$F_0(V) \wedge T(V,V') \wedge T(V',V'') \wedge T(V'',V''') \wedge B_1(V''')$

- Formula is unsatisfiable
  - Extract an interpolation-sequence: I_1, I_2, I_3
  - I_j over-approximates states reachable in j steps
- Use I_j to strengthen F_j
  - Example: $F_3' = F_3 \wedge I_3$
  - $F_3' \wedge B_1$ is unsatisfiable
- Re-Apply Local Strengthening
  - $F_3(V) \wedge T(V,V') \wedge \neg P(V')$ is unsatisfiable

Global Strengthening

- If a CEX exists - Full unrolling
- Otherwise, gradually unroll the model
  - Try to reach the Backward sequence
- When the backward sequence is not reachable
  - Extract interpolation sequence
  - Strengthen forward sequence
  - Reapply Local Strengthening

Checking if a "fixpoint" has been reached

- $F_k \Rightarrow \vee_{j=1,n-1} F_j$
- But we also have the backward sequence
  $B_k \Rightarrow \vee_{j=1,n-1} B_j$
- Same principle applies here check inclusion for every $2 \leq k \leq n$

(Local) Summary

- Use both Forward and Backward traversals in a tight manner
- Mostly local - No unrolling
  - Inspired by IC3/PDR
- When unrolling is used, it is restricted
  - Experiments confirm

(Global) Summary

We presented several methods for SAT-based (unbounded) model checking

- Over-approximate the (forward) reachability analysis
- Apply different methods for making the over-approximation more precise
  - Reduce number of spurious counterexamples
  - (Hopefully) help termination (fixpont)

Thank You