Towards a Theoretical Understanding of IC3/PDR

Sharon Shoham

TEL AVIV UNIVERSITY אוניברסיטת תל אביב

Supervised Verification of Infinite-State Systems

Yotam Feldman

TEL AVIV UNIVERSITY אוניברסיטת תל אביב

Neil Immerman

UNIVERSITY OF MASSACHUSETTS AMHERST 1863

Mooly Sagiv

TEL AVIV UNIVERSITY אוניברסיטת תל אביב

CERTORA

James R. Wilcox

UNIVERSITY of WASHINGTON

SAT-Based Invariant Inference

- predicate abstraction
  [CAV'97, POPL'02]
- symbolic abstraction
  [VMCAI'04,'16]
- interpolation
  [CAV'03, TACAS'06]
- IC3/PDR
  [VMCAI'11, FMCAD'11]
- abduction
  [OOPSLA'13]
- SyGuS
  [FMCAD'13,...]
- ICE learning
  [CAV'14, POPL'15]
- ...

Why do they succeed?

Why do they fail?

(How can we make them better?)

Goal

Shed light on the principles underlying IC3/PDR

[POPL'20] Complexity and information in invariant inference. Feldman, Immerman, Sagiv, Shoham
[POPL'21] Learning the boundary of inductive invariants. Feldman, Sagiv, Shoham, Wilcox
[POPL'22] Property-directed reachability as abstract interpretation in the monotone theory. Feldman, Sagiv, Shoham, Wilcox
[SAS'22] Invariant Inference With Provable Complexity From the Monotone Theory. Feldman, Shoham

Safety of Transition Systems

No bad state is reachable from the initial states

Init:

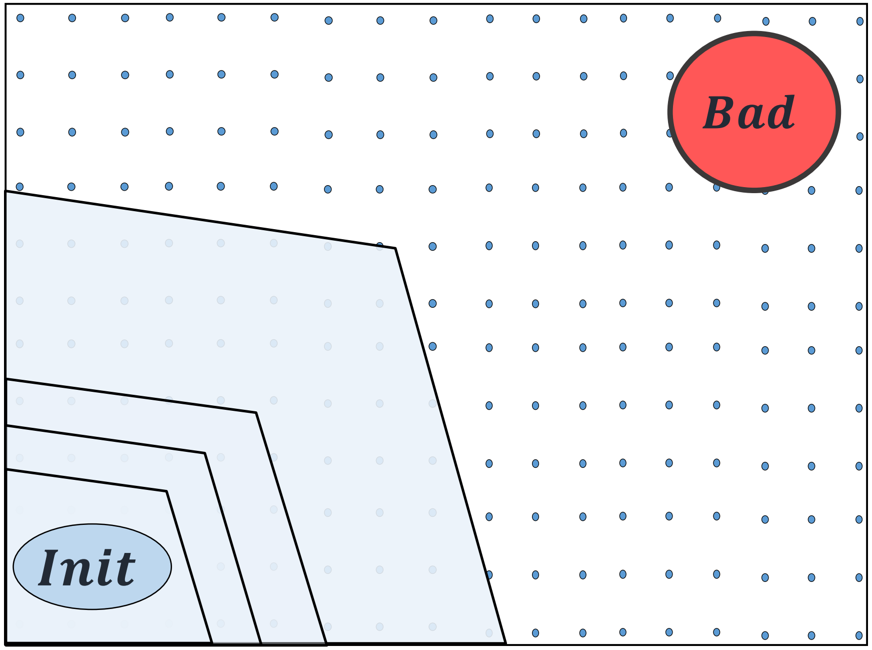
$(x_n, \ldots, x_0) \coloneqq 00 \ldots 00$

δ:

$(x_n, \ldots, x_0) \coloneqq (x_n, \ldots, x_0) + 00 \ldots 10$

Bad:

$(x_n, \ldots, x_0) = 10 \ldots 01$



Inductive Invariants

No bad state is reachable from the initial states

Init:

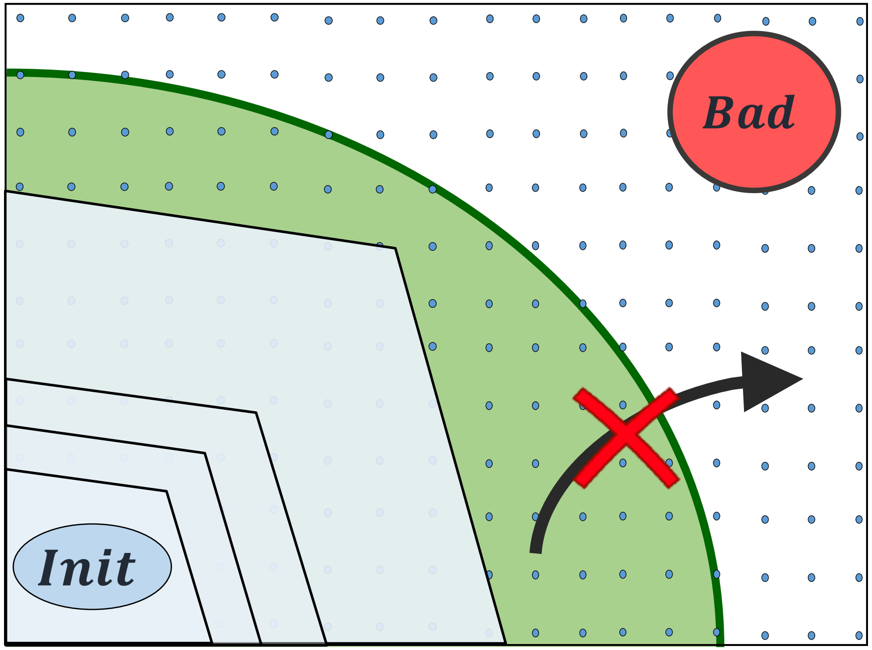
$(x_n, \dots, x_0) := 00 \dots 00$

δ:

$(x_n, \dots, x_0) := (x_n, \dots, x_0) + 00 \dots 10$

Bad:

$(x_n, \dots, x_0) = 10 \dots 01$



Initiation: Init $\subseteq I$

Safety: $I \cap \text{Bad} = \emptyset$

Consecution: $\delta(I) \subseteq I$

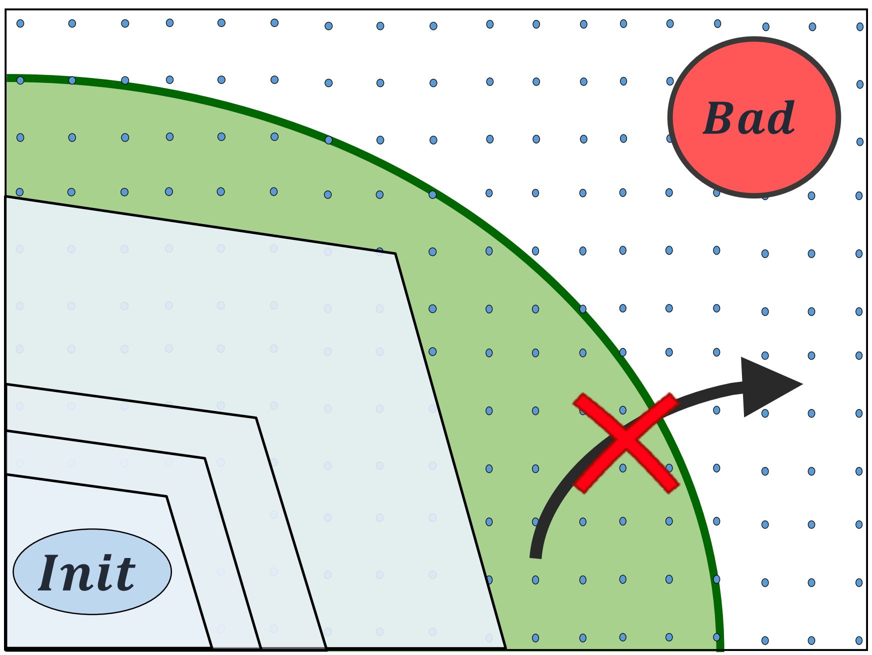
Inductive Invariants

No bad state is reachable from the initial states

Goal:

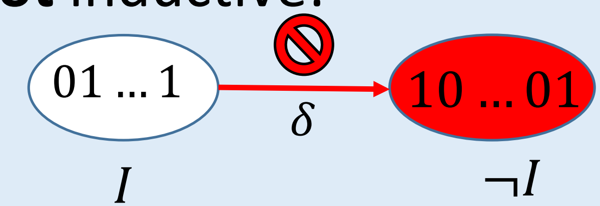
Find inductive invariants automatically

$(x_n, \dots, x_0) \; = \; 10 \dots 01$



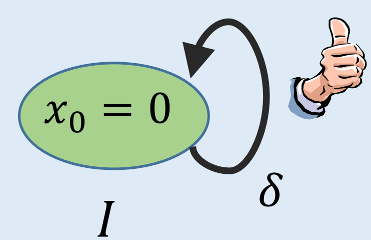
I: $(x_n, \dots, x_0) \neq 10 \dots 01$

Not inductive:



I: $x_0 = 0$

Inductive:



Invariant Inference with IC3/PDR

Prof. Aaron R. Bradley

The HVC 2012 Award Committee, chaired by Prof. Daniel Kroening, has decided to give this year's award to **Prof. Aaron R. Bradley of CU Boulder** for the invention of the IC3 algorithm.

IC3 is an algorithm for verifying reachability properties on finite-state transition systems, and has been proposed by Aaron Bradley, CU Boulder. There are independent reports of leading performance on hardware-verification problems, which is impressive given the decades of research on competing techniques, including BDDs and recent innovations such as Craig interpolation with propositional SAT. The technique has been picked up by others, as evidenced by a variant of the algorithm for software presented at CAV this year. The committee believes that IC3 has innovated this mature research area, and produced new impulses for numerous adjacent research problems.

[VMCAI’11] SAT-Based Model Checking Without Unrolling. Bradley
[FMCAD’11] Efficient Implementation of Property Directed Reachability. Ean, Mishchenko, Brayton

The HVC 2012 Award Committee, chaired by Prof. Daniel Kroening, ha decided to give this year's award to **Prof. Aaron R. Bradley of CU Boulder** for the invention of the IC3 algorithm.

IC3 is an algorithm for verifying reachability properties on finite-state transition systems, and has been proposed by Aaron Bradley, CU Boul There are independent reports of leading performance on hardware-verification problems, which is impressive given the decades of researc nniques, including BDDs and recent innovations such as Craig interpolati SAT. The technique has been picked up by others, as evidenced by a orithm for software presented at CAV this year. The committee believes th this mature research area, and produced new impulses for numerous problems.

[VMCAI'11] SAT-Based Model Checking Without Unrolling. Bradley
[FMCAD'11] Efficient Implementation of Property Directed Reachability. Ean, Mishchenko, Brayton

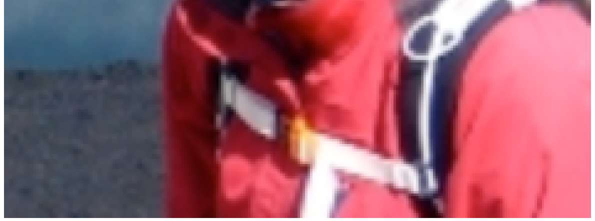
Invariant Inference with IC3/PDR

Prof. Aaron R. Bradley

The HVC 2012 Award Committee, chaired by Prof. Daniel Kroening, has decided to give this year's award to **Prof. Aaron R. Bradley of CU Boulder** for the invention of the IC3 algorithm.

IC3 is an algorithm for verifying reachability properties on finite-state transition systems, and has been proposed by Aaron Bradley, CU Boulder. There are independent reports of leading performance on hardware-verification problems, which is impressive given the decades of research on competing techniques, including BDDs and recent innovations such as Craig interpolation with propositional SAT. The technique has been picked up by others, as evidenced by a variant of the algorithm for software presented at CAV this year. The committee believes that IC3 has innovated this mature research area, and produced new impulses for numerous adjacent research problems.

[VMCAI'11] SAT-Based Model Checking Without Unrolling. Bradley
[FMCAD'11] Efficient Implementation of Property Directed Reachability. Ean, Mishchenko, Brayton



Prof. Aaron R. Bradley

IC3 is an algorithm for verifying r
transition systems, and has been
There are independent reports of
verification problems, which is im
on competing techniques, including BDDs and recen
with propositional SAT. The technique has been pick
variant of the algorithm for software presented at CA
IC3 has innovated this mature research area, and pr
adjacent research problems.

[VMCAI'11] SAT-Based Model Checking Without Unrolling. Bradley
[FMCAD'11] Efficient Implementation of Property Directed Reachability. Ean, Mishchenko, Brayton

Invariant Inference with IC3/PDR

Prof. Aaron R. Bradley

The HVC 2012 Award Committee, chaired by Prof. Daniel Kroening, has decided to give this year's award to **Prof. Aaron R. Bradley of CU Boulder** for the invention of the IC3 algorithm.

IC3 is an algorithm for verifying reachability properties on finite-state transition systems, and has been proposed by Aaron Bradley, CU Boulder. There are independent reports of leading performance on hardware-verification problems, which is impressive given the decades of research on competing techniques, including BDDs and recent innovations such as Craig interpolation with propositional SAT. The technique has been picked up by others, as evidenced by a variant of the algorithm for software presented at CAV this year. The committee believes that IC3 has innov … nerous adjacent rese…

What makes IC3/PDR so effective?

[VMCAI'11] SAT-Based Model Checking Without Unrolling. Bradley
[FMCAD'11] Efficient Implementation of Property Directed Reachability. Ean, Mishchenko, Brayton

IC3/PDR

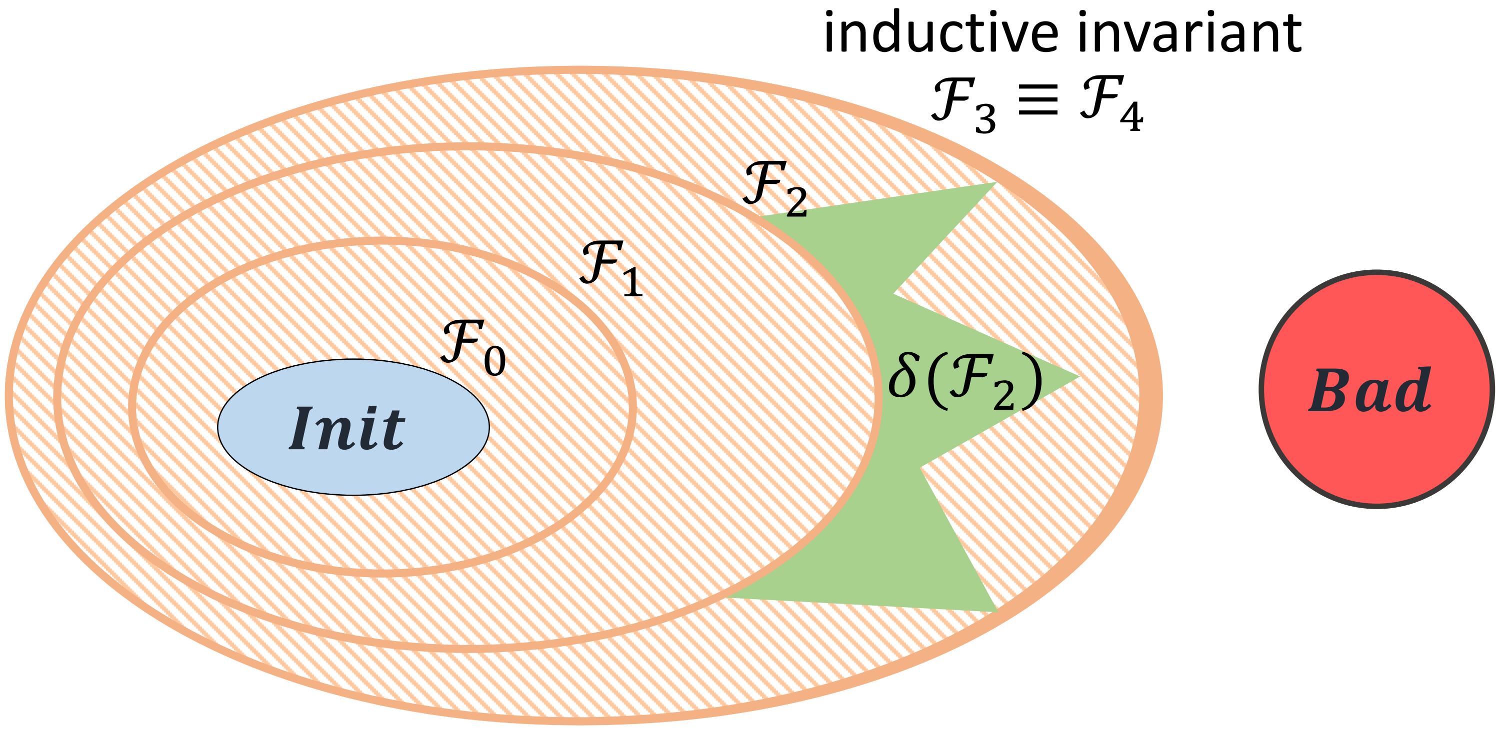
A sequence of frames (formulas) $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \ldots$

(1) $\mathcal{F}_0 = \mathrm{Init}$

(2) $\mathcal{F}_i \Longrightarrow \mathcal{F}_{i+1}$

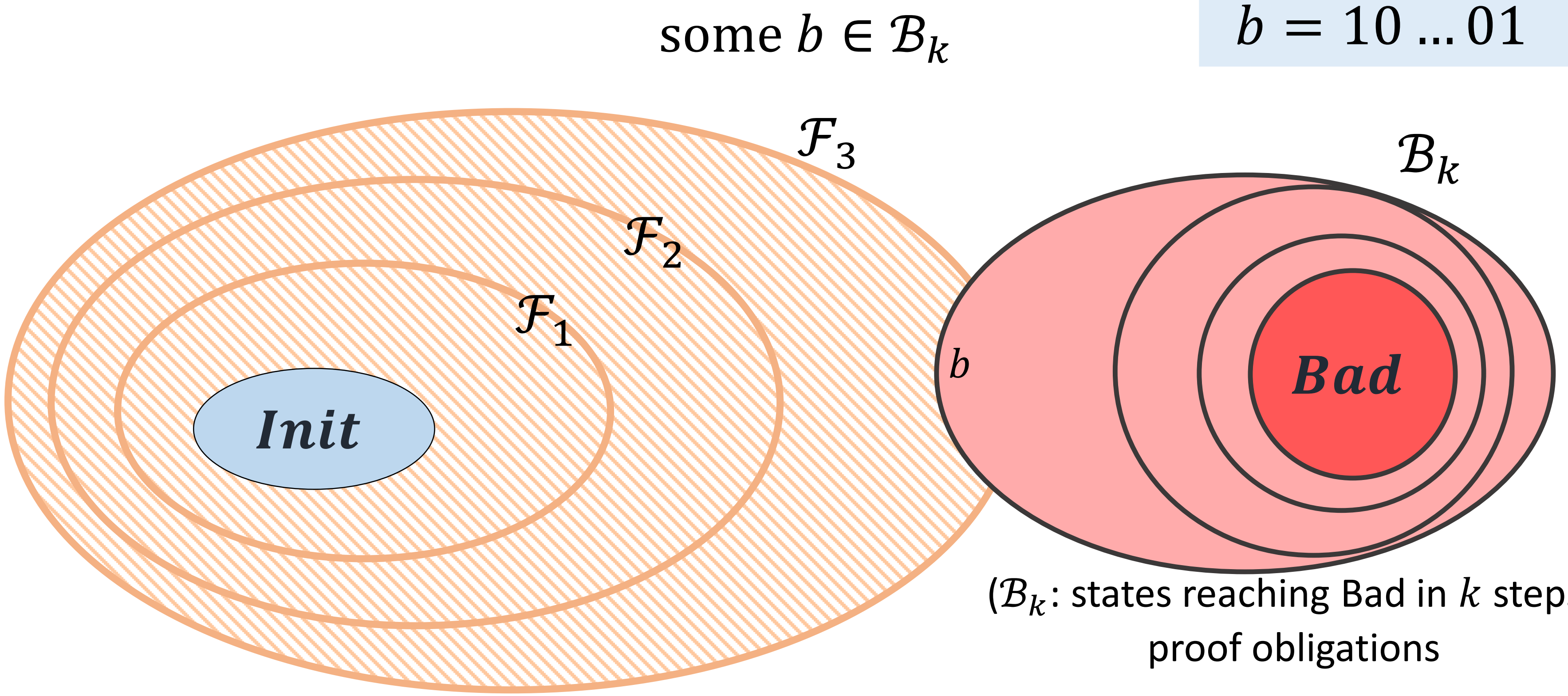
(3) $\delta(\mathcal{F}_i) \Longrightarrow \mathcal{F}_{i+1}$

(4) $\mathcal{F}_i \Longrightarrow \neg\mathrm{Bad}$



Lemmas Block Counterexamples

Example:
$b = 10 \dots 01$



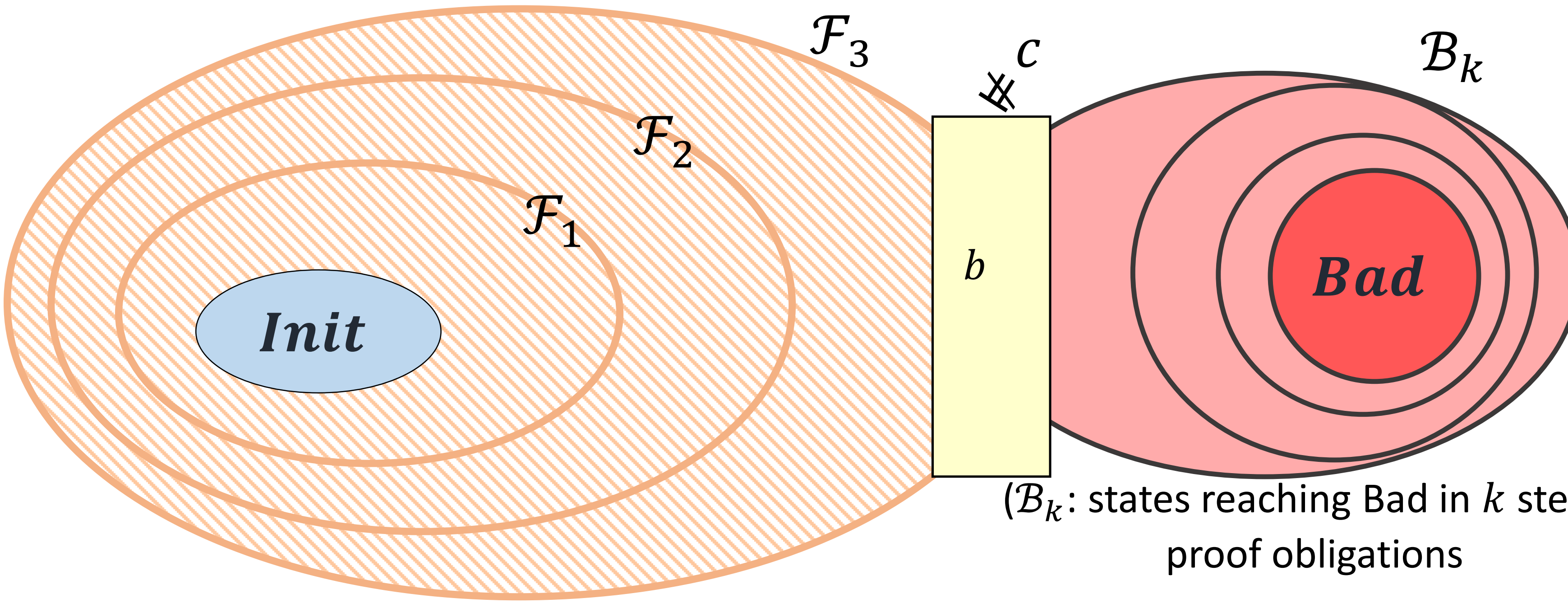
($\mathcal{B}_k$: states reaching Bad in k steps)
proof obligations

Lemmas Block Counterexamples

$\mathcal{F}_{i+1} \coloneqq \mathcal{F}_{i+1} \wedge c$ for some c s.t.

$b \not\models c$ for some $b \in \mathcal{B}_k$

Example:
$b = 10 \ldots 01$



($\mathcal{B}_k$: states reaching Bad in k steps)
proof obligations

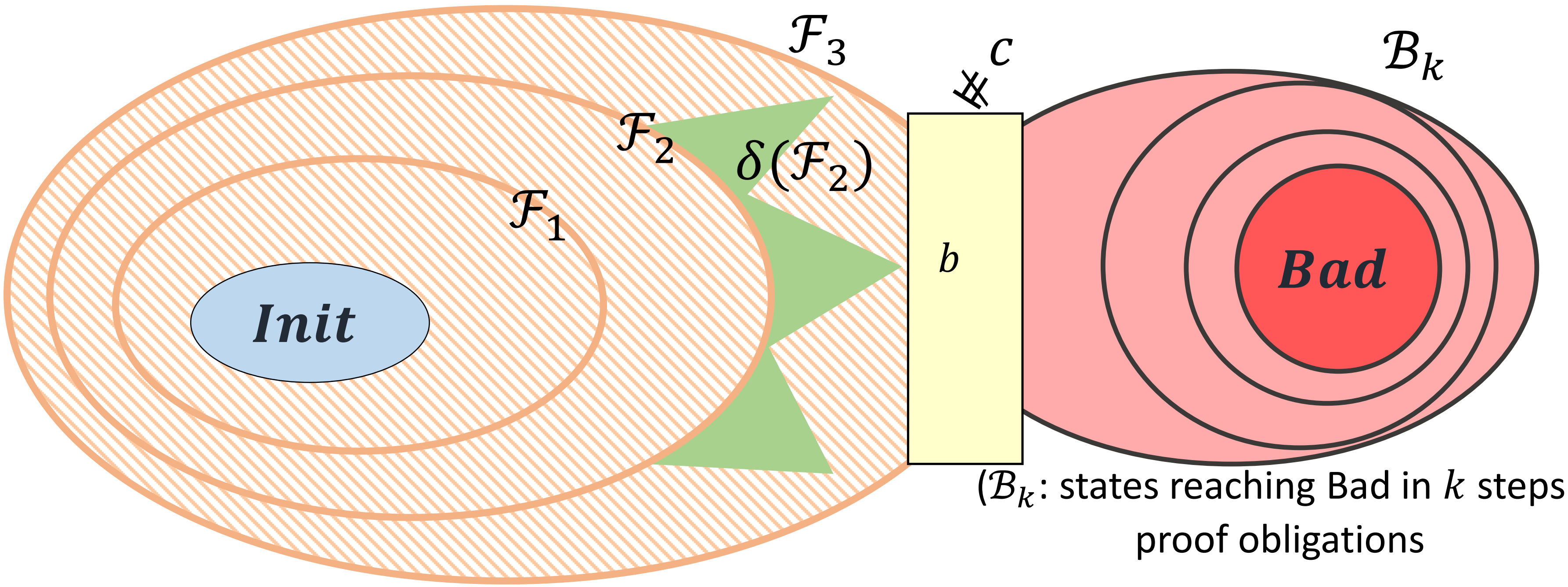
Lemmas Block Counterexamples

$\mathcal{F}_{i+1} \coloneqq \mathcal{F}_{i+1} \wedge c$ for some c s.t.

$\delta(\mathcal{F}_i) \subseteq c$ and $Init \subseteq c$ and

$b \not\models c$ for some $b \in \mathcal{B}_k$

Example:
$b = 10 \ldots 01$



Lemmas Block Counterexamples

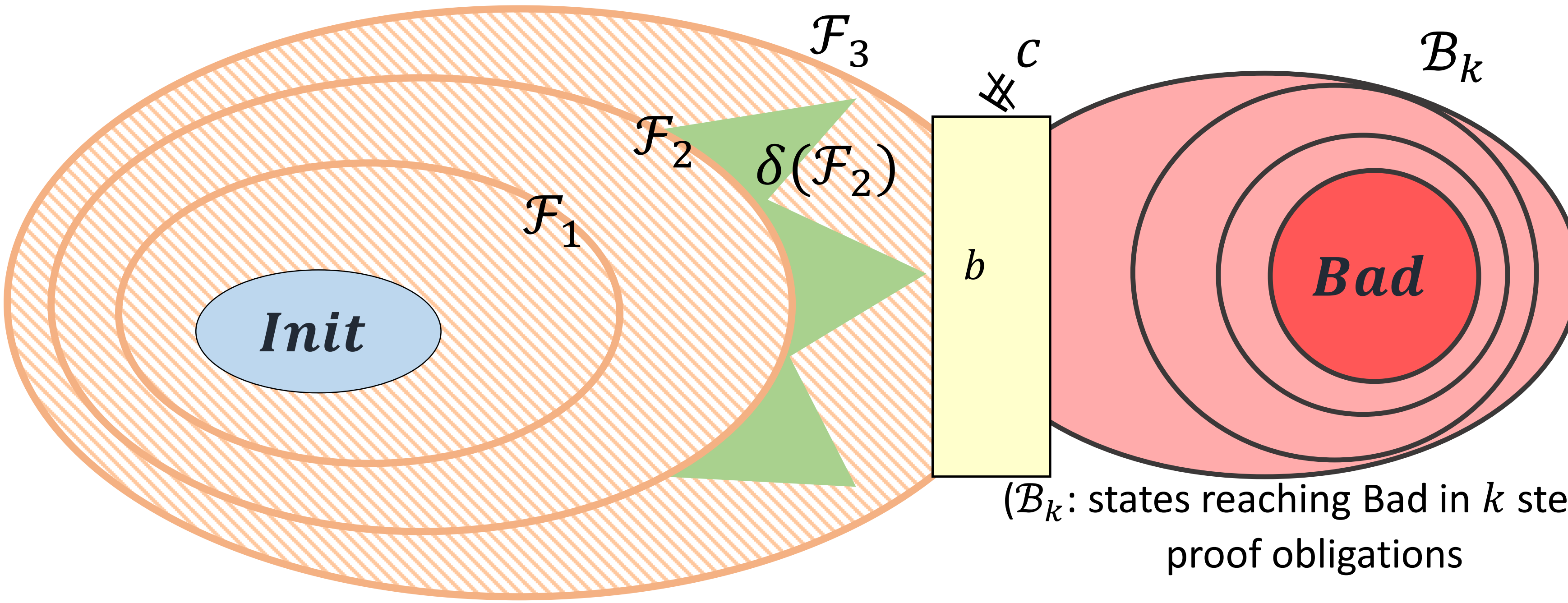
c: $(x_n = 0) \vee (x_{n-1} = 1) \vee \cdots \vee (x_1 = 1) \vee (x_0 = 0)$

$\mathcal{F}_{i+1} \coloneqq \mathcal{F}_{i+1} \wedge c$ for some c s.t.

$\delta(\mathcal{F}_i) \subseteq c$ and $Init \subseteq c$ and

$b \nvDash c$ for some $b \in \mathcal{B}_k$

Example:
$b = 10 \ldots 01$



($\mathcal{B}_k$: states reaching Bad in k steps)
proof obligations

Lemmas Block Counterexamples

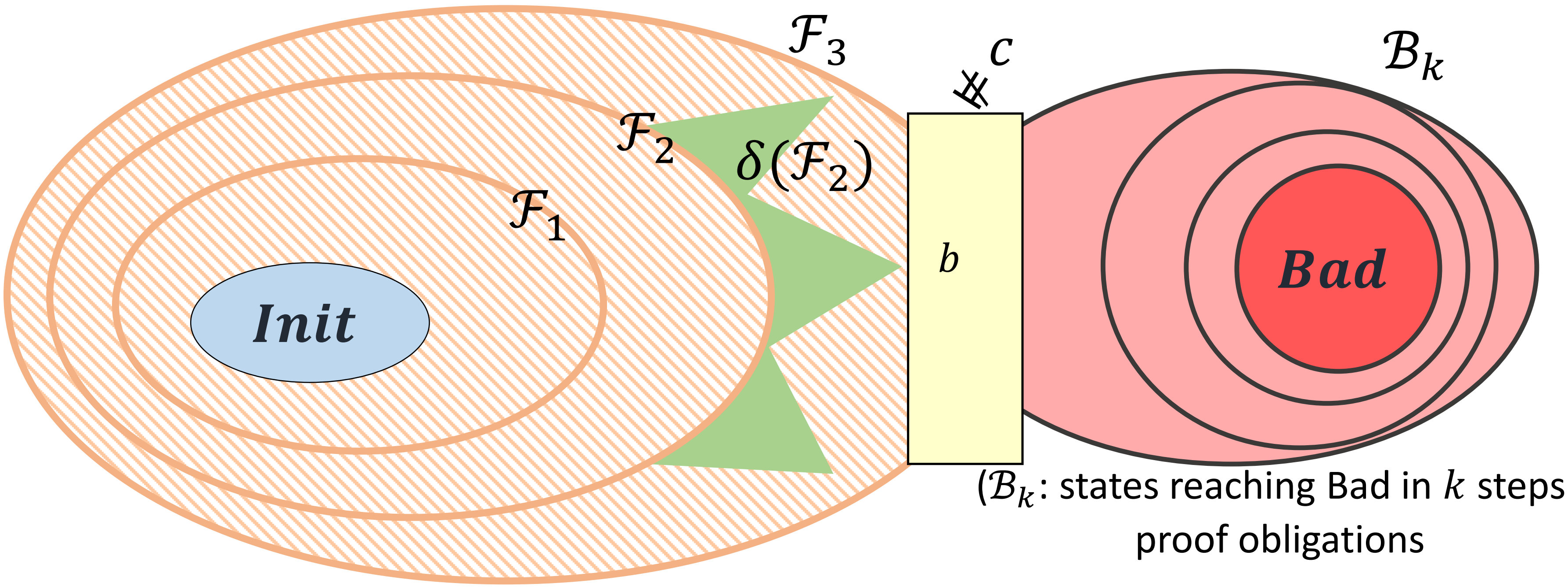
c: $(x_n = 0) \vee \cancel{(x_{n-1} = 1)} \vee \cancel{\cdots} \vee (x_1 = 1) \vee (x_0 = 0)$

$\mathcal{F}_{i+1} \coloneqq \mathcal{F}_{i+1} \wedge c$ for some c s.t.

$\delta(\mathcal{F}_i) \subseteq c$ and $Init \subseteq c$ and

$b \not\models c$ for some $b \in \mathcal{B}_k$

Example:
$b = 10 \ldots 01$



Lemmas Block Counterexamples

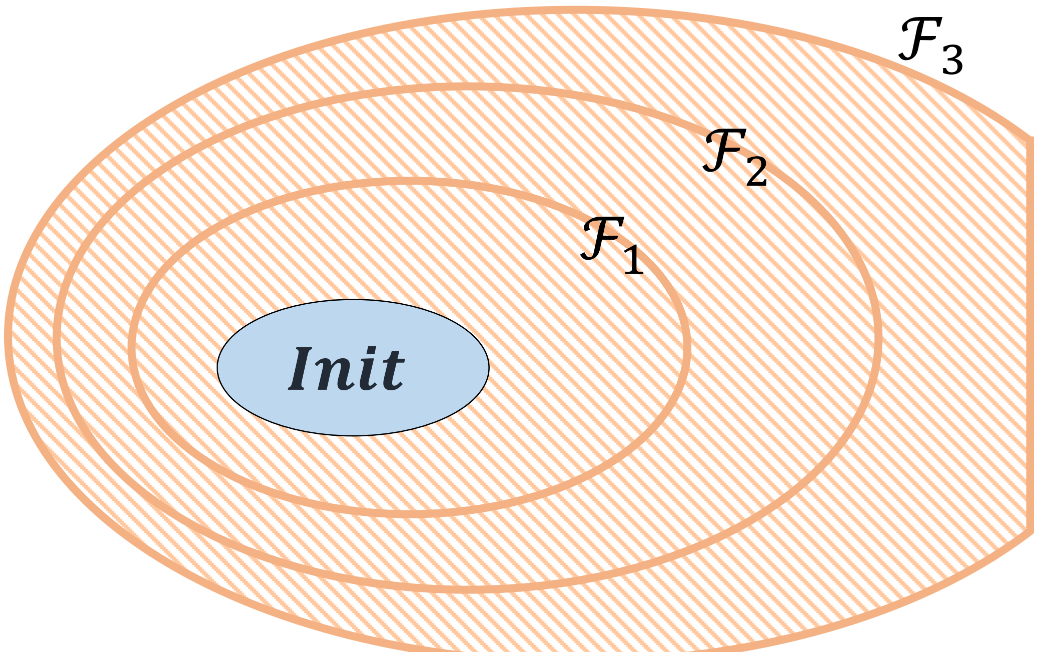
c: $(x_n = 0) \vee (x_{n-1} = 1) \vee (x_0 = 0)$

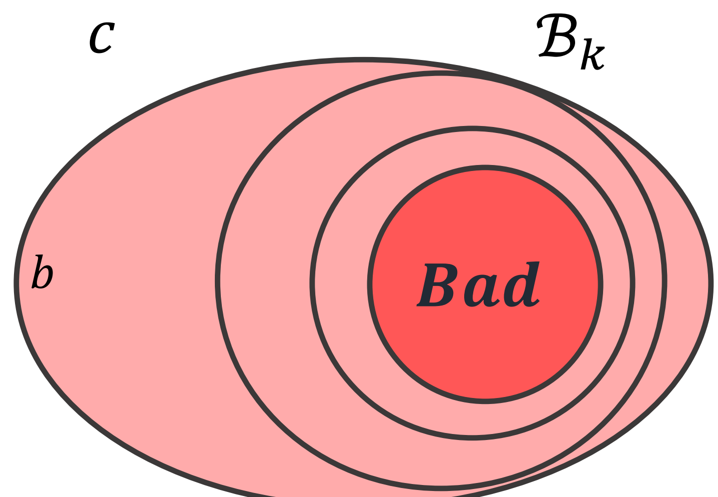
$\mathcal{F}_{i+1} \coloneqq \mathcal{F}_{i+1} \wedge c$ for some c s.t.

$\delta(\mathcal{F}_i) \subseteq c$ and $Init \subseteq c$ and

$b \not\models c$ for some $b \in \mathcal{B}_k$

Example:
$b = 10 \ldots 01$





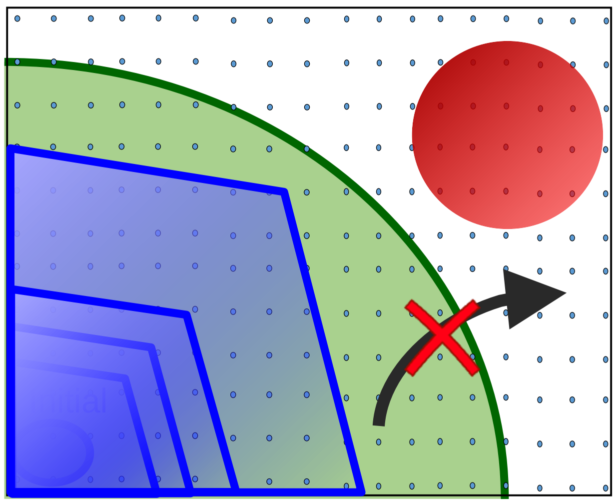
($\mathcal{B}_k$: states reaching Bad in k steps)
proof obligations

This Talk

- SAT queries performed by IC3/PDR
- Generalization/overapproximation in IC3/PDR

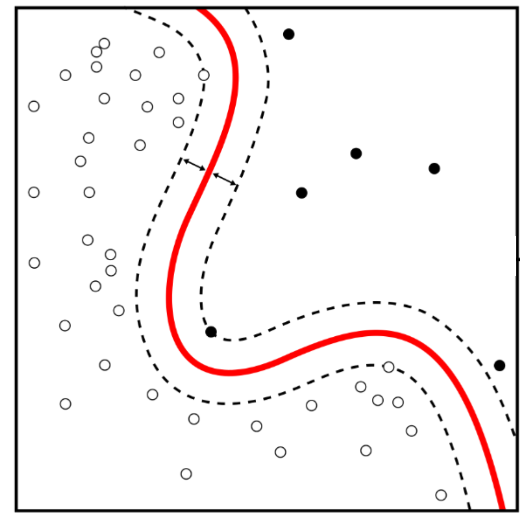
Part I: PDR's SAT Queries

Invariant Inference



as

Exact Concept Learning



- Query-based learning models for SAT-based invariant inference
- Exponential complexity gap between different query models

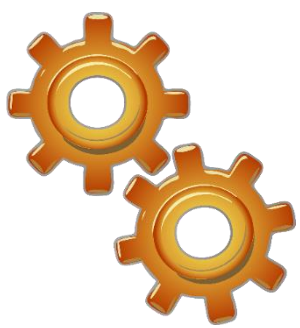
The power of PDR's SAT queries

[POPL'20] Complexity and information in invariant inference. Feldman, Immerman, Sagiv, Shoham

Exact Concept Learning with Equivalence & Membership Queries

Goal: learn an unknown concept φ

learning algorithm



is it ψ_1?

✓ / ✗+counterexample

is it ψ_2?

✓ / ✗+counterexample

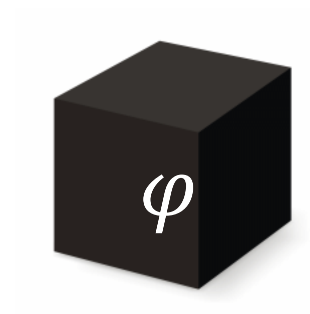
does $\sigma_3 \models$?

✓ / ✗

…

Membership Equivalence

oracle

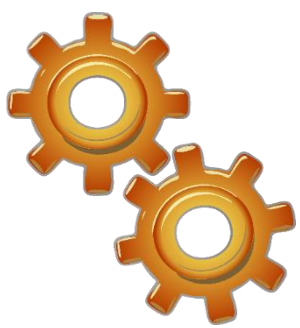


[ML'87] Queries and Concept Learning. Angluin

SAT-Based Invariant Inference as Inference with Queries

Goal: infer an unknown inductive invariant I

learning algorithm $\mapsto$ inference algorithm



Q

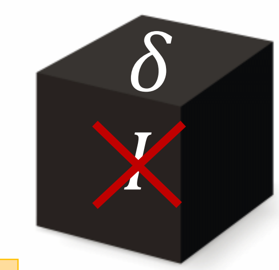
A

Q

A

Q

oracle $\mapsto$ SAT-solver

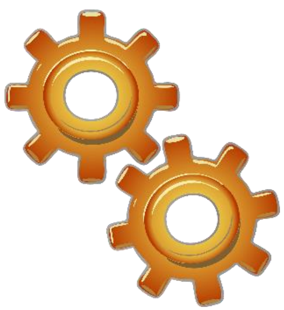


Which SAT queries?

Algorithms cannot access the transition relation directly, only through SAT queries

Inductiveness-Query Model

inference algorithm



α_1 inductive?

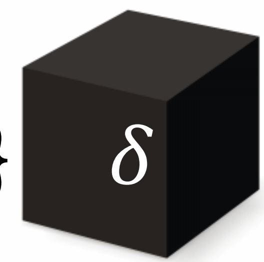
✓ / ✗+counterexample

…

α_m inductive?

✓ / ✗+counterexample

inductiveness-query oracle

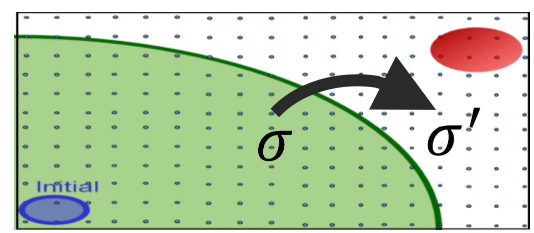
$\{\alpha_i\}$  $\{\alpha_i\}$

$\boldsymbol{\alpha_i \wedge \delta \wedge \neg\alpha_i'}$ unsat?

Cex to Induction (CTI):
Transition (σ, σ') of δ s.t.
$\sigma \vDash \alpha_i,\quad \sigma' \vDash \neg\alpha_i$

<u>ICE framework</u> <u>- Learn from examples:</u>

Positive : $\sigma \vDash I$ (e.g., initial)
Negative: $\sigma \nvDash I$ (e.g., bad)
Implication: $\sigma \vDash I$ implies $\sigma' \vDash I$ (CTI)

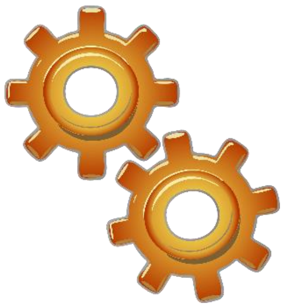


* α_i'=$\alpha_i[V \mapsto V']$

[CAV'14] ICE: A Robust Framework for Learning Invariants. Garg, Löding, Madhusudan, Neider

Inductiveness-Query Model

inference algorithm



α_1 inductive?

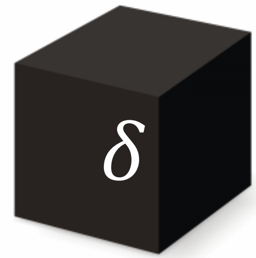
✓ / ✗+counterexample

…

α_m inductive?

✓ / ✗+counterexample

inductiveness-query oracle

$\{\alpha_i\}$ 

$\{\alpha_i\}$

$\boldsymbol{\alpha_i \wedge \delta \wedge \neg\alpha_i'}$ unsat?

Cex to Induction (CTI):
Transition (σ, σ') of δ s.t.
$\sigma \vDash \alpha_i, \quad \sigma' \vDash \neg\alpha_i$

ICE framework - Learn from examples:

Positive : $\sigma \vDash I$ (e.g., initial)

Negative: $\sigma \nvDash I$ (e.g., bad)

Implication: $\sigma \vDash I$ implies $\sigma' \vDash I$ (CTI)

Is it sufficient to capture IC3/PDR?

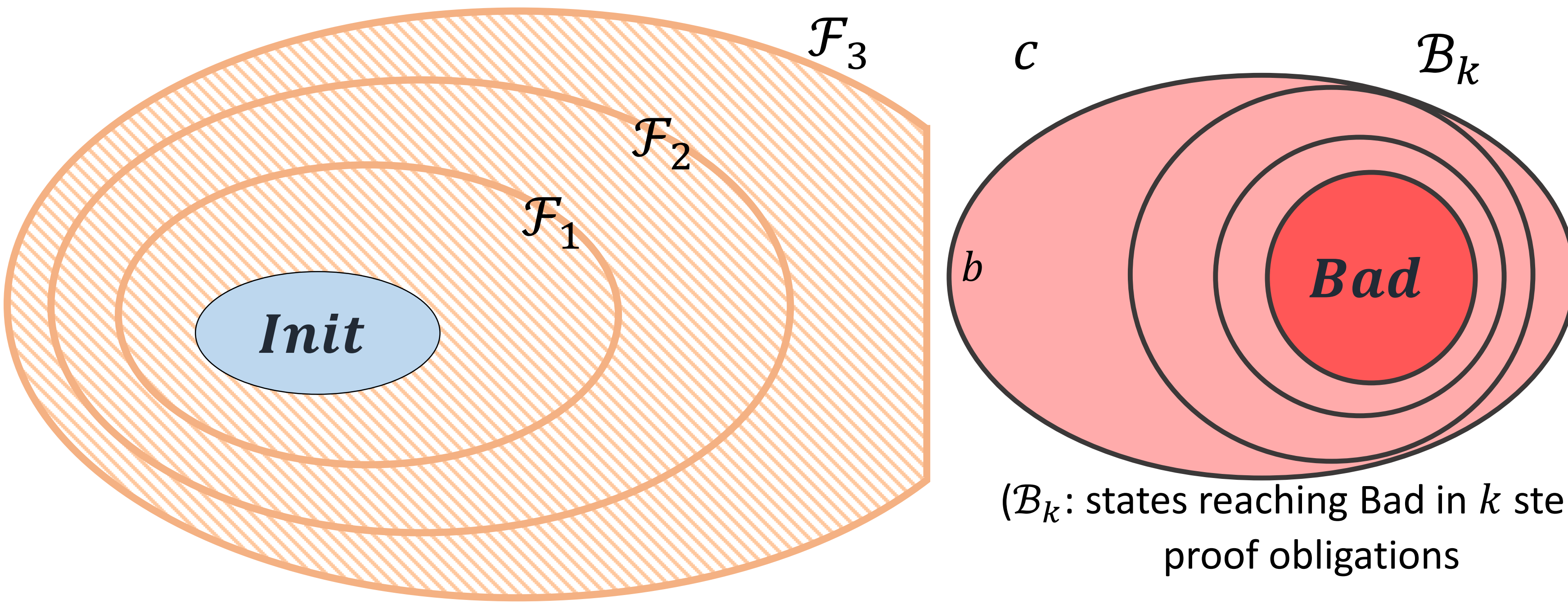
* $\boldsymbol{\alpha_i'}$=$\boldsymbol{\alpha_i}[V \mapsto V']$

[CAV'14] ICE: A Robust Framework for Learning Invariants. Garg, Löding, Madhusudan, Neider

PDR

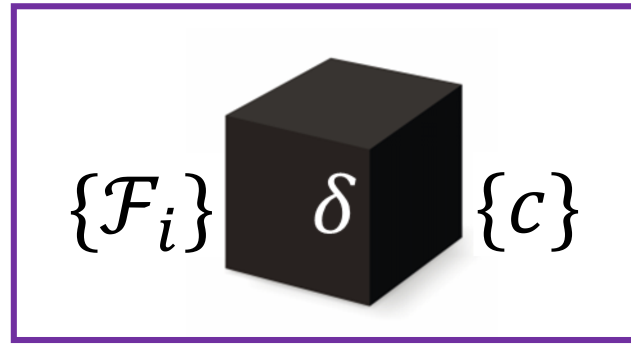
Is it captured by the inductiveness query model?

$$\mathcal{F}_{i+1} \coloneqq \mathcal{F}_{i+1} \wedge c \ \text{ for some } c \text{ s.t.}$$
$$\boxed{\delta(\mathcal{F}_i) \subseteq c} \text{ and } Init \subseteq c \ \text{ and}$$
$$b \nvDash c \ \text{ for some } b \in \mathcal{B}_k$$



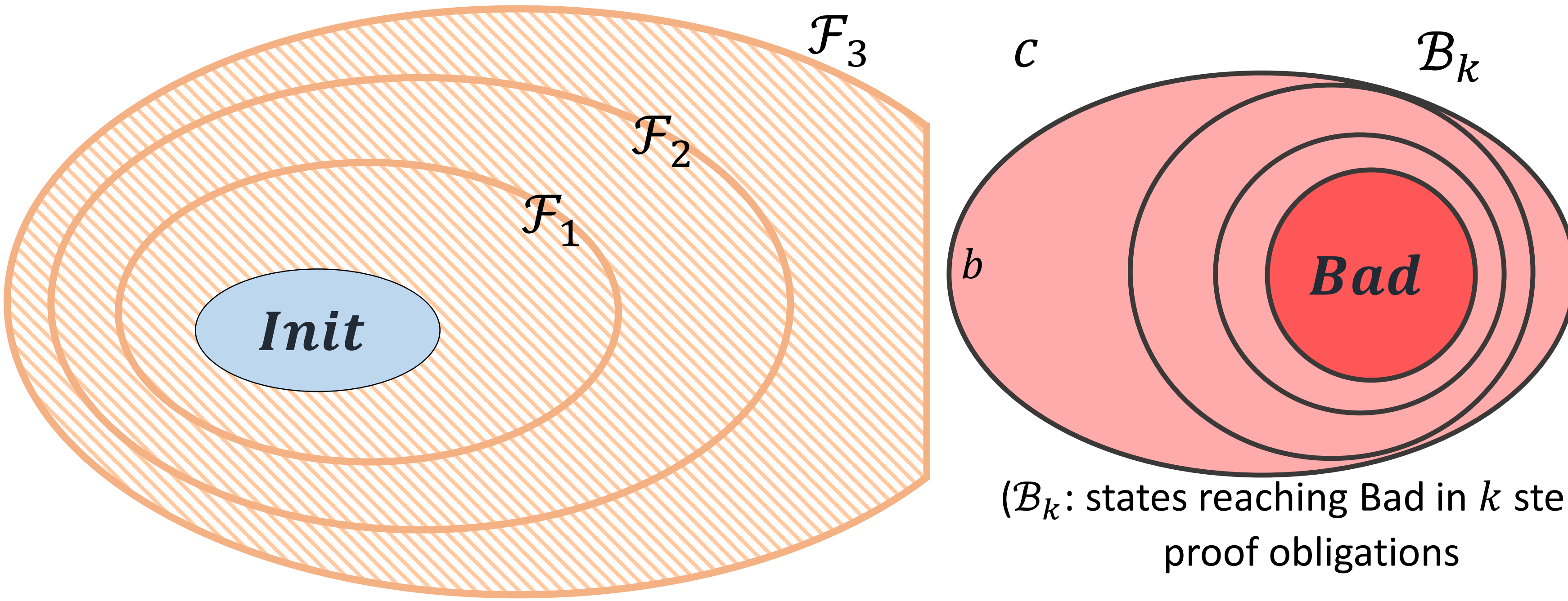
($\mathcal{B}_k$: states reaching Bad in k steps)
proof obligations

PDR



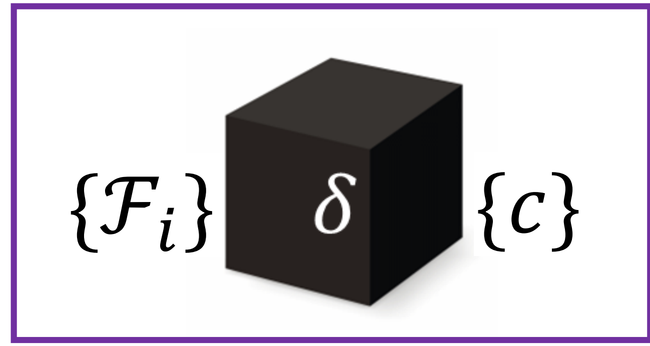
Is it captured by the inductiveness query model?

$$\mathcal{F}_{i+1} \coloneqq \mathcal{F}_{i+1} \wedge c \ \text{ for some } c \text{ s.t.}$$
$$\boxed{\delta(\mathcal{F}_i) \subseteq c} \text{ and } \ Init \subseteq c \ \text{ and}$$
$$b \nvDash c \ \text{ for some } b \in \mathcal{B}_k$$



($\mathcal{B}_k$: states reaching Bad in k steps)
proof obligations

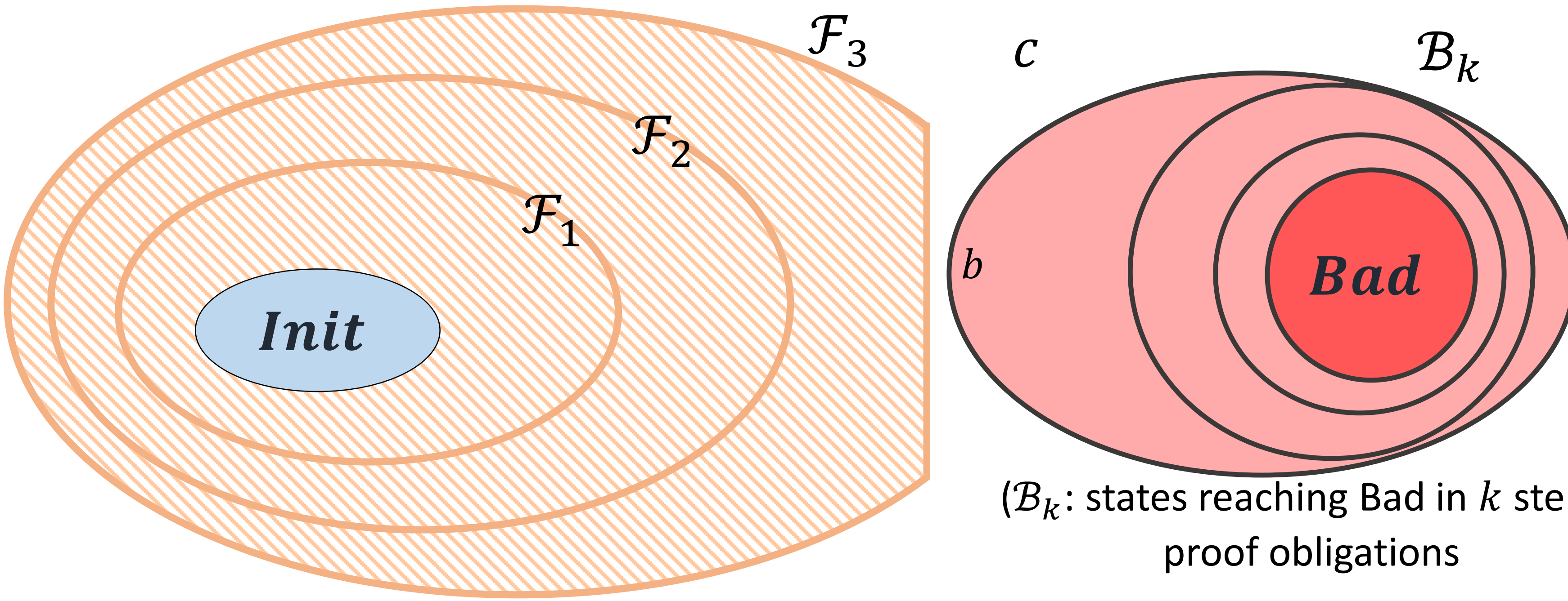
PDR



Is it captured by the inductiveness query model?

$$\mathcal{F}_{i+1} \coloneqq \mathcal{F}_{i+1} \wedge c \ \text{ for some } c \text{ s.t.}$$
$$\boxed{\delta(\mathcal{F}_i) \subseteq c} \text{ and } Init \subseteq c \text{ and}$$
$$b \nvDash c \ \text{ for some } b \in \mathcal{B}_k$$

[VMCAI'17]
extended ICE -
Is it necessary?

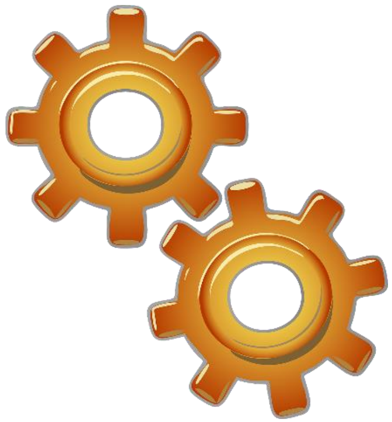


($\mathcal{B}_k$: states reaching Bad in k steps)
proof obligations

[VMCAI'17] IC3 - Flipping the E in ICE. Vizel, Gurfinkel, Shoham, Malik.

Hoare-Query Model

inference algorithm



Hoare-query oracle

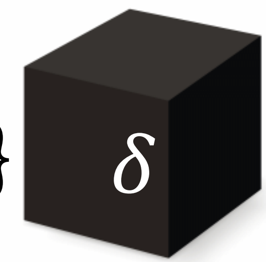
$\{\alpha_1\}\ \delta\ \ \{\beta_1\}?$

✓ / ✗ +counterexample

…

$\{\alpha_m\}\ \delta\ \ \{\beta_m\}?$

✓ / ✗ +counterexample

$\{\alpha_i\}$ 

$\{\beta_i\}$

$\boldsymbol{\alpha_i \wedge \delta \wedge \neg\beta_i'}$ unsat?

Transition (σ, σ') of δ s.t.
$\sigma \vDash \alpha_i, \sigma' \vDash \neg\beta_i$

Capable of modeling PDR, Interpolation-based inference and more

Hoare > Inductiveness

<u>Thm</u>: There exists a class of transition systems $\mathcal{P}$, so that for solving inference:

1. ∃ Hoare-query algorithm with $\text{poly}(n)$ queries
2. ∀ inductiveness-query algorithm requires $2^{\Omega(n)}$ queries

[POPL'20] Complexity and Information in Invariant Inference. Feldman, Immerman, Shoham, Sagiv

Hoare > Inductiveness

<u>Thm</u>: There exists a class of transition systems $\mathcal{P}$, so that for solving inference:

1. $\exists$ Hoare-query algorithm with $\mathrm{poly}(n)$ queries
2. $\forall$ inductiveness-query algorithm requires $2^{\Omega(n)}$ queries

a simple case of IC3/PDR

$\Rightarrow$ ICE cannot model PDR,

and the extension of [VMCAI'17] is necessary

[POPL'20] Complexity and Information in Invariant Inference. Feldman, Immerman, Shoham, Sagiv
[VMCAI'17] IC3 - Flipping the E in ICE. Vizel, Gurfinkel, Shoham, Malik.

Hoare > Inductiveness

Thm: There exists a class of transition systems $\mathcal{P}$, so that for solving inference:

1. $\exists$ Hoare-query algorithm with $\mathrm{poly}(n)$ queries
2. $\forall$ inductiveness-query algorithm requires $2^{\Omega(n)}$ queries

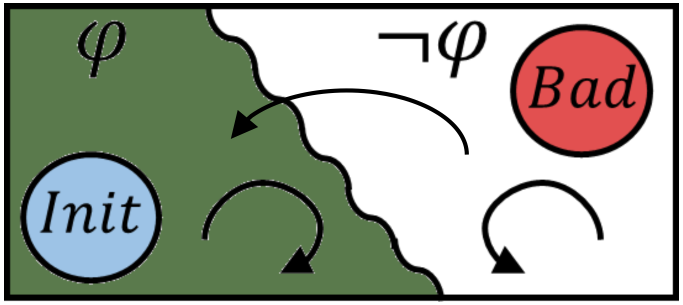
Proof:

$\mathcal{P}$ = maximal transition systems for monotone CNF with n cubes

propositions appear only positively

$$\varphi = x_1 \wedge (x_2 \vee x_3)$$

Maximal system for φ:



Hoare > Inductiveness

<u>Upper bound</u>:

PDR takes $\boldsymbol{O}(\boldsymbol{n}^2)$ queries

$$\mathcal{F}_1 \ := \mathcal{F}_1 \ \wedge c \ \text{ for some } c \text{ s.t.}$$

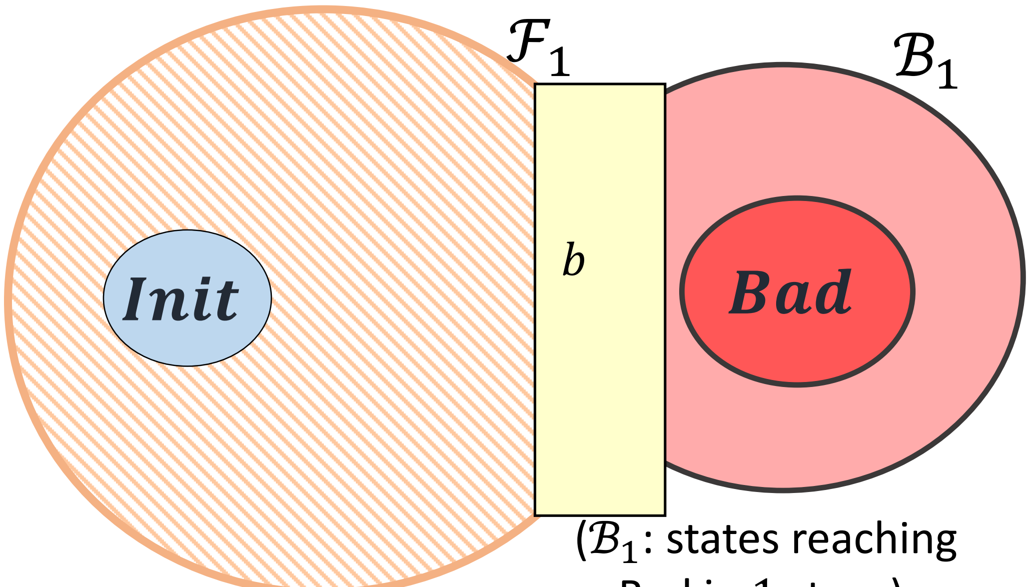
$$\delta(Init) \ \subseteq \ c \ \text{ and } \ Init \subseteq c \ \text{ and}$$

$$b \nvDash c \ \text{ for some } b \in \mathcal{B}_1$$

generalize$(\neg b)$:

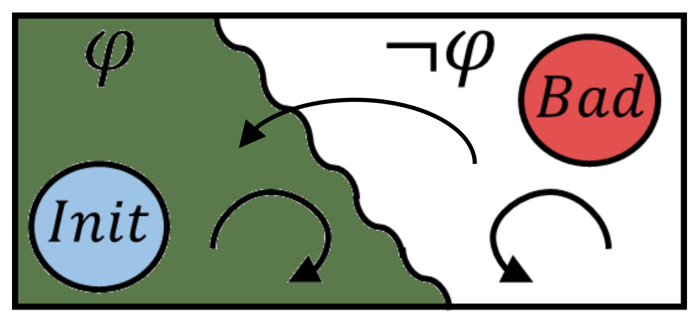
drop literals from $\neg b$

while $\{Init\}\ \delta\ \{\neg b\}$



($\mathcal{B}_1$: states reaching Bad in 1 steps)

$$\varphi \ = x_1 \wedge (x_2 \vee x_3)$$



Hoare > Inductiveness

Upper bound:

PDR takes $\boldsymbol{O(n^2)}$ queries

$\mathcal{F}_1 := \mathcal{F}_1 \wedge c$ for some c s.t.

$\delta(Init) \subseteq c$ and $Init \subseteq c$ and

$b \not\models c$ for some $b \in \mathcal{B}_1$

generalize$(\neg b)$:

drop literals from $\neg b$

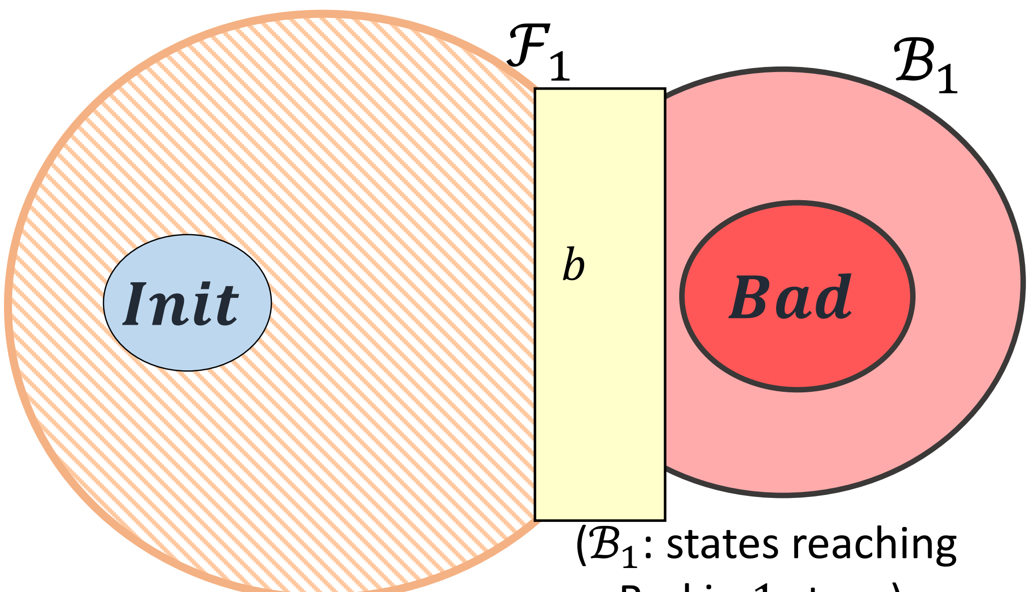
while $\{Init\}\ \delta\ \{\neg b\}$

minimal

$\varphi \Rightarrow \neg b$

φ is monotone

$$\varphi = \underbrace{x_1}_{\text{1 iteration}} \wedge \underbrace{(x_2 \vee x_3)}_{\text{1 iteration}}$$



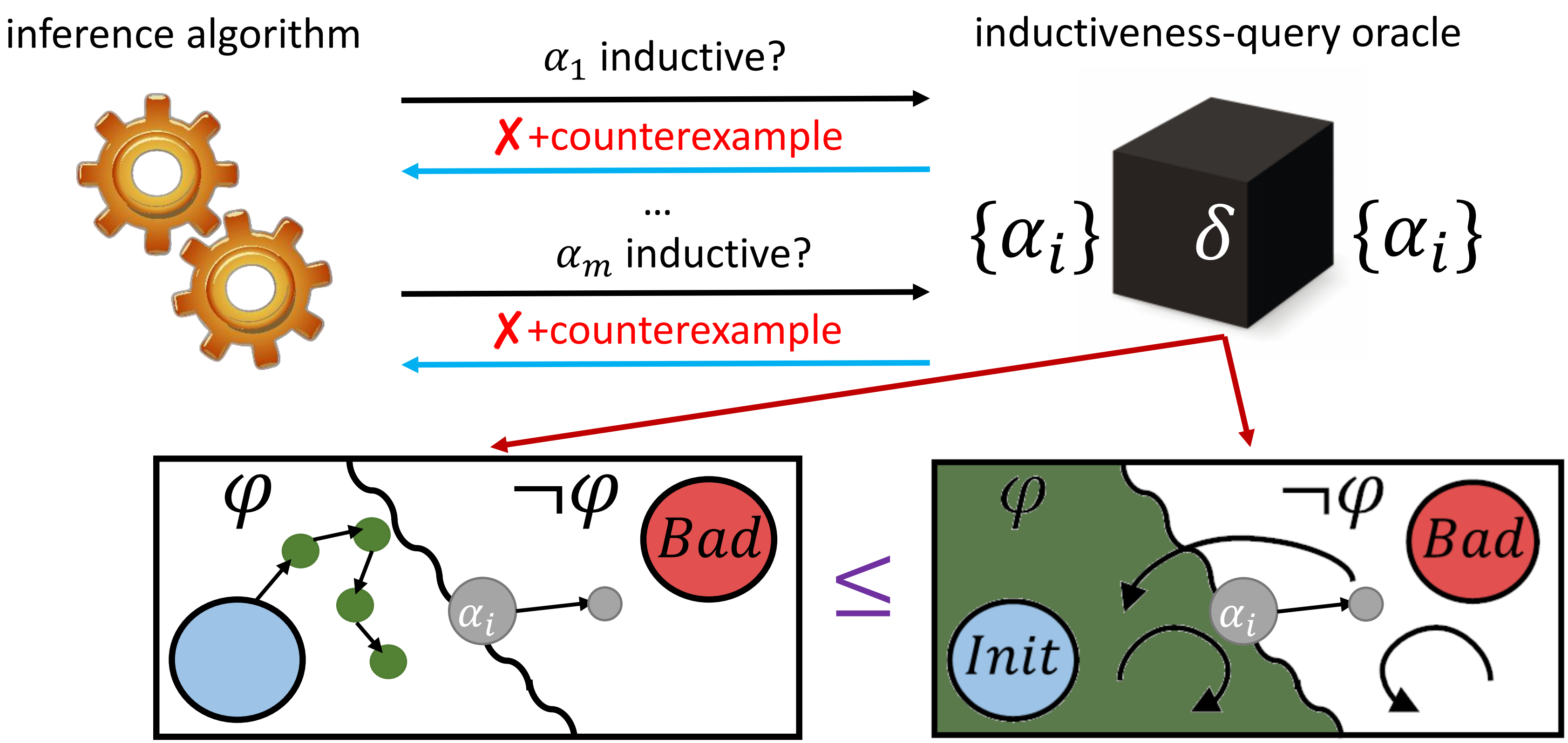
($\mathcal{B}_1$: states reaching Bad in 1 steps)

Hoare > Inductiveness

Lower bound:

∀ inductiveness-query algorithm requires $2^{\Omega(n)}$ queries

Proof:



Hoare > Inductiveness

<u>Lower bound</u>:

∀ inductiveness-query algorithm requires $2^{\Omega(n)}$ queries

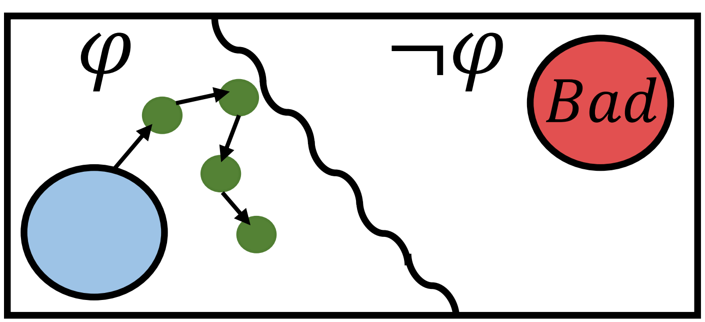
<u>Proof:</u>

Thm:
invariants in monotone CNF

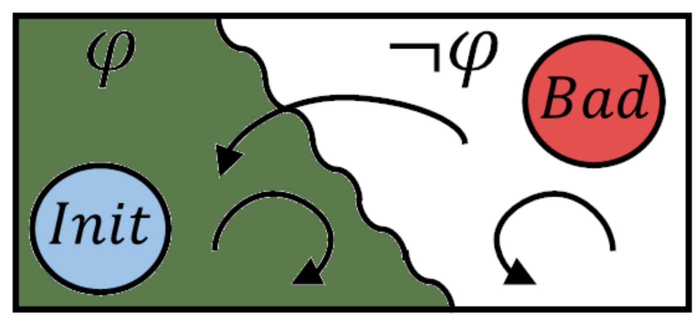
general systems
monotone CNF invariants

maximal systems
monotone CNF invariants

$2^{\Omega(n)} \leq$



$\leq$



Hoare > Inductiveness

Thm: There exists a class of transition systems $\mathcal{P}$, so that for solving inference:

1. ∃ Hoare-query algorithm with $\mathrm{poly}(n)$ queries
2. ∀ inductiveness-query algorithm requires $2^{\Omega(n)}$ queries

More generally:

Condition on (δ, Inv) that allows efficient inference of:

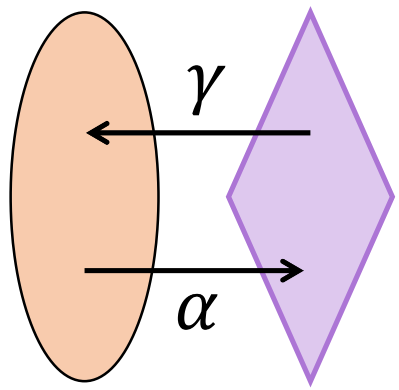
- Monotone CNF invariants -- by Model-based ITP
- Almost-monotone CNF invariants
- CDNF invariants

[POPL'20] Complexity and Information in Invariant Inference. Feldman, Immerman, Shoham, Sagiv
[POPL'21] Learning the boundary of inductive invariants. Feldman, Sagiv, Shoham, Wilcox
[SAS'22] Invariant Inference With Provable Complexity From the Monotone Theory. Feldman, Shoham

Part II: PDR's Generalization

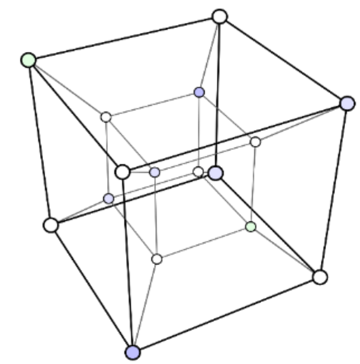
PDR is (roughly) an abstract interpretation
based on Bshouty's monotone theory from exact learning

PDR $\approx$



Abstract
Interpretation

+



Monotone
Theory

Principle of how PDR achieves overapproximation

[POPL'22] PDR as abstract interpretation in the monotone theory. Feldman, Sagiv, Shoham, Wilcox
[SAS'22] Invariant Inference With Provable Complexity From the Monotone Theory. Feldman, Shoham

PDR's Frames

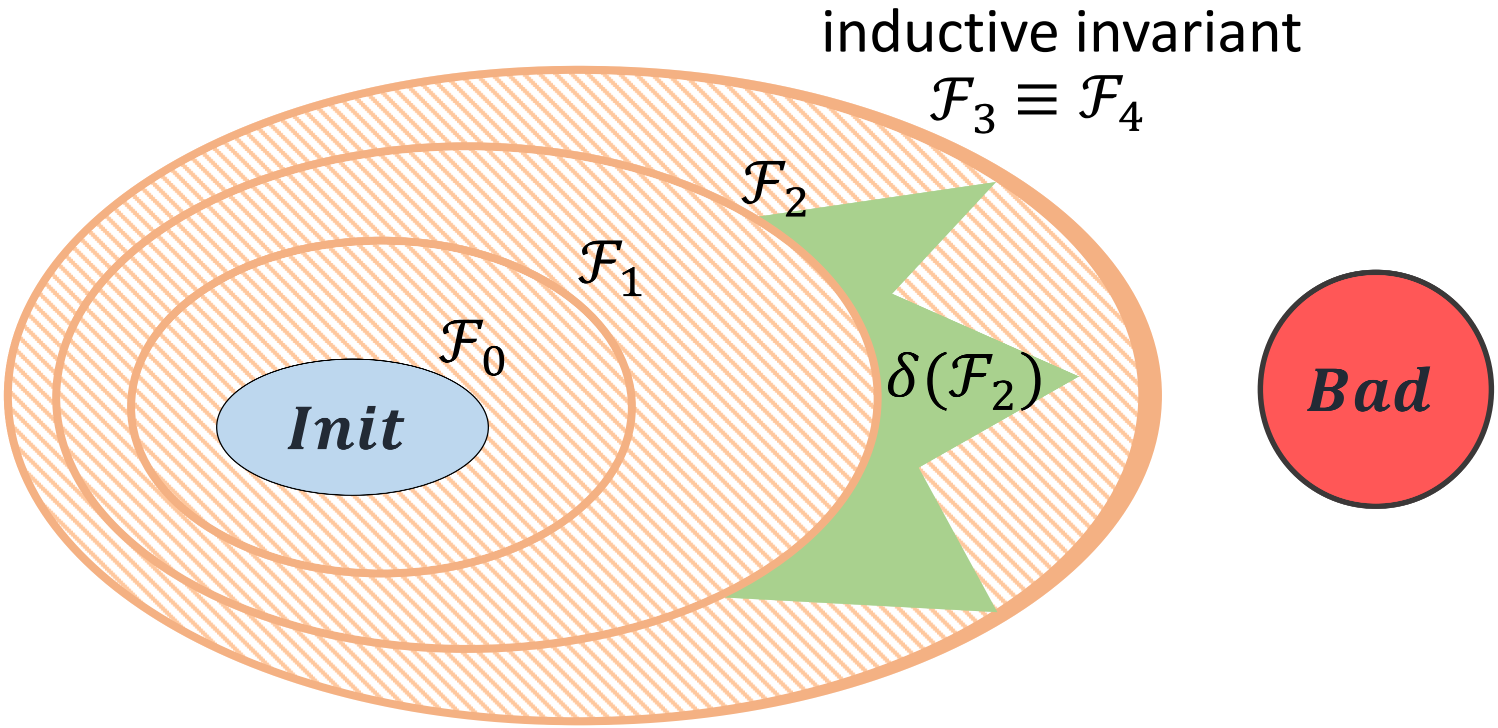
A sequence of formulas $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \ldots$

(1) $\mathcal{F}_0 = \mathrm{Init}$

(2) $\mathcal{F}_i \Longrightarrow \mathcal{F}_{i+1}$

(3) $\delta(\mathcal{F}_i) \Longrightarrow \mathcal{F}_{i+1}$

(4) $\mathcal{F}_i \Longrightarrow \neg\mathrm{Bad}$



PDR's Frames

A sequence of formulas $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \ldots$

$\mathcal{R}_i \subseteq \mathcal{F}_i$

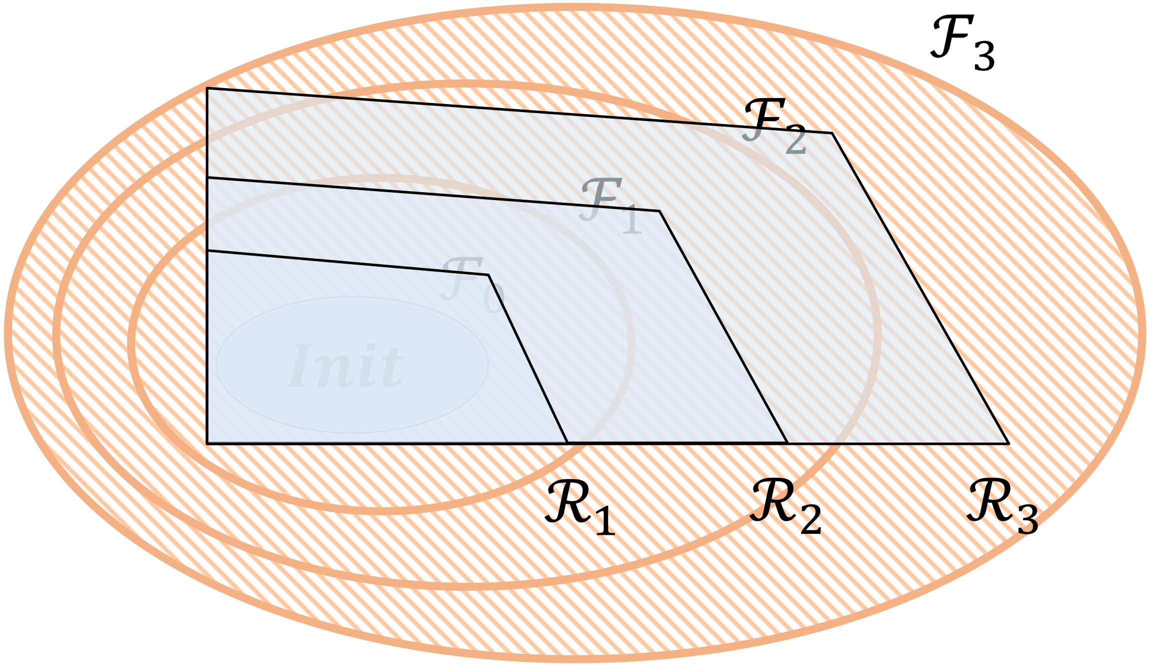
($\mathcal{R}_i$: states reachable in i steps)

(1) $\mathcal{F}_0 = \text{Init}$

(2) $\mathcal{F}_i \Longrightarrow \mathcal{F}_{i+1}$

(3) $\delta(\mathcal{F}_i) \Longrightarrow \mathcal{F}_{i+1}$

(4) $\mathcal{F}_i \Longrightarrow \neg\text{Bad}$



PDR's Frames

A sequence of formulas $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \ldots$

$\mathcal{R}_i \subseteq \mathcal{F}_i$

($\mathcal{R}_i$: states reachable in i steps)

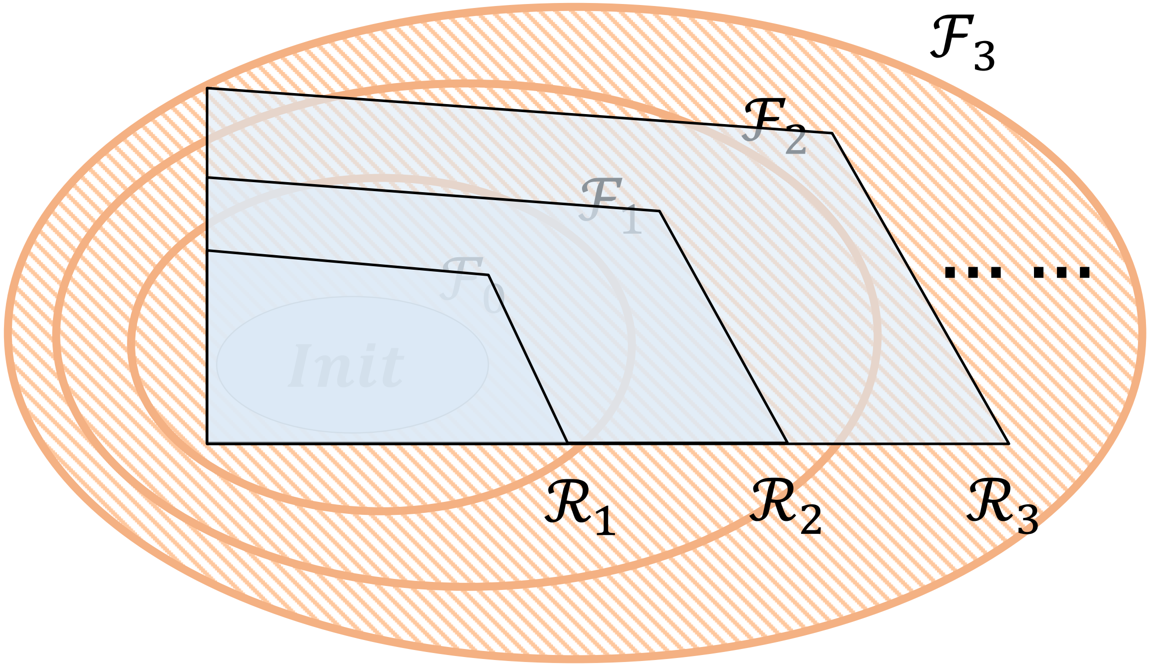
(1) $\mathcal{F}_0 = \mathrm{Init}$

(2) $\mathcal{F}_i \Longrightarrow \mathcal{F}_{i+1}$

(3) $\delta(\mathcal{F}_i) \Longrightarrow \mathcal{F}_{i+1}$

(4) $\mathcal{F}_i \Longrightarrow \neg\mathrm{Bad}$

$\mathcal{R}_i \ll \mathcal{F}_i$



PDR's Frames

Init:

$(x_n, \ldots, x_0) \coloneqq$ 00 … 00

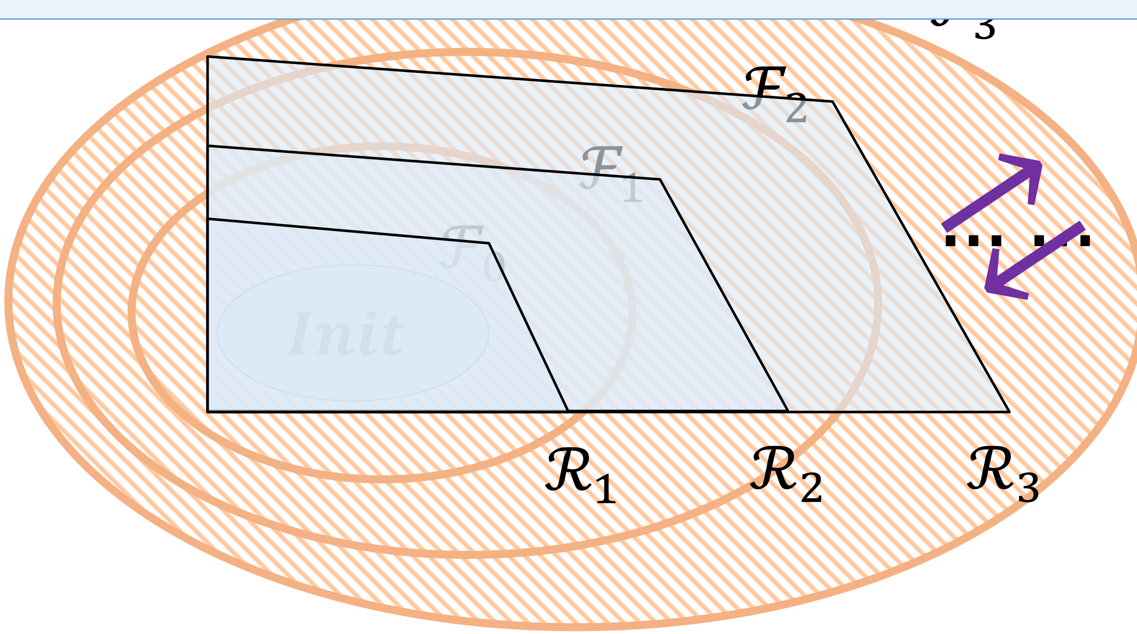
δ:

$(x_n, \ldots, x_0) \coloneqq (x_n, \ldots, x_0) +$ 00 … 10

Bad:

$(x_n, \ldots, x_0) =$ 10 … 01

$$\mathcal{R}_i : \ (x_n, \ldots, x_0) \leq 2i \wedge x_0 = 0$$



$$\mathcal{R}_i \ll \mathcal{F}_i$$

PDR's Frames

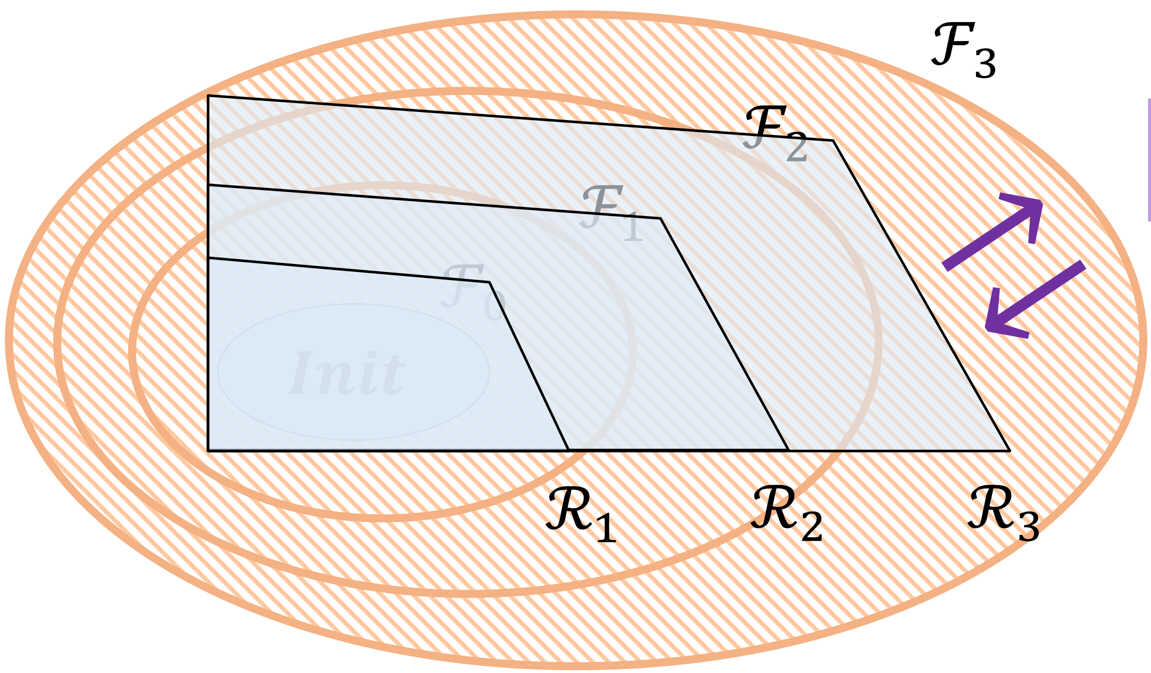
A sequence of formulas $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \ldots$

How is overapproximation guaranteed?

($\mathcal{R}_i$: states reachable in i steps)

(3) $\delta(\mathcal{F}_i) \Longrightarrow \mathcal{F}_{i+1}$

(4) $\mathcal{F}_i \Longrightarrow \neg\mathrm{Bad}$

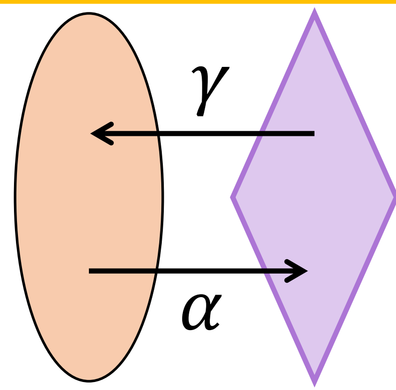


$\mathcal{R}_i \ll \mathcal{F}_i$

Key Ideas and Results

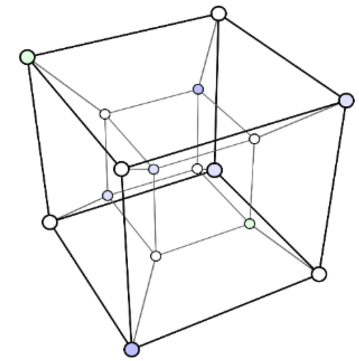
Λ-PDR: an algorithm that lower bounds the overapproximation of PDR

Λ-PDR =



Abstract Interpretation

+



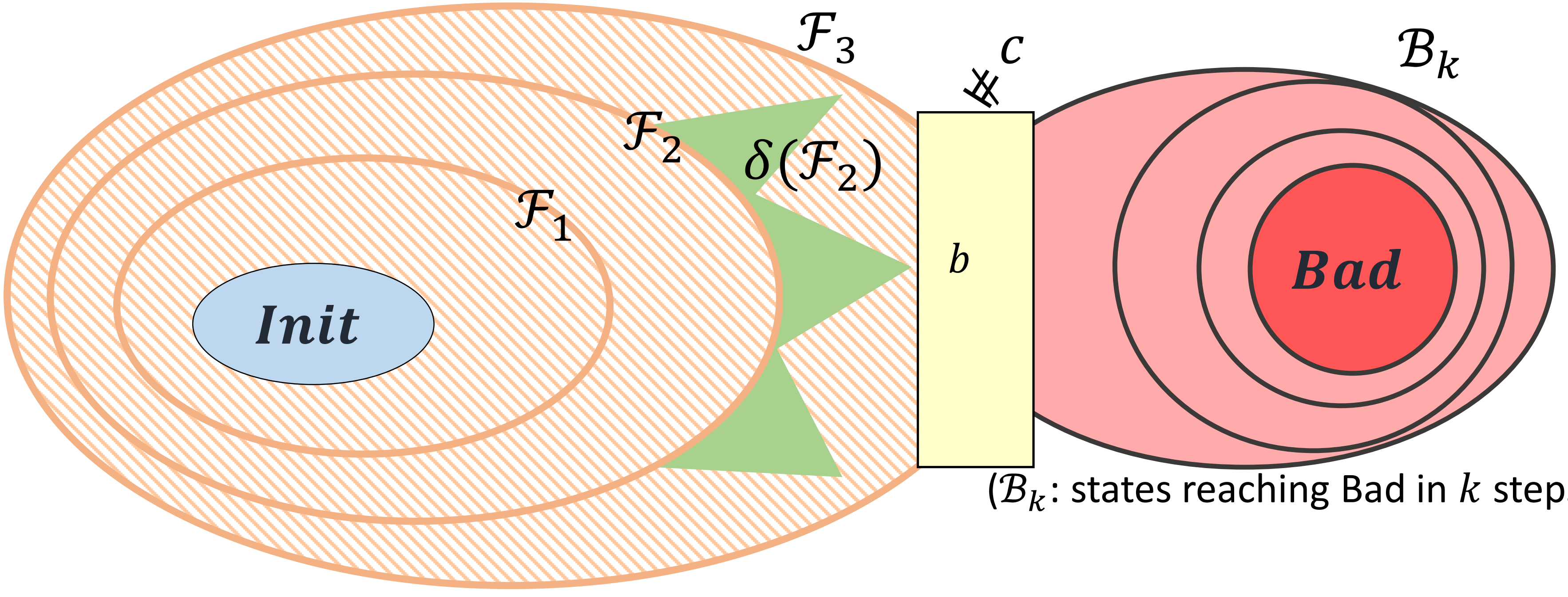
Monotone Theory

- Exponential gap between #frames in Λ-PDR and exact forward reachability, interpolation

The same overapproximation is present in PDR

PDR's Frames

$$\mathcal{F}_{i+1} \coloneqq \text{conjunction of } \textbf{some clauses } c \text{ s.t.}$$
$$\delta(\mathcal{F}_i) \cup \mathcal{F}_i \subseteq c \text{ and}$$
$$b \not\models c \text{ for } \textbf{some } b \in \mathcal{B}_k$$



($\mathcal{B}_k$: states reaching Bad in k steps)

Λ-PDR's Frames

Eager PDR

$\mathcal{F}_0^{\Lambda} \coloneqq Init;\ i \coloneqq 0$

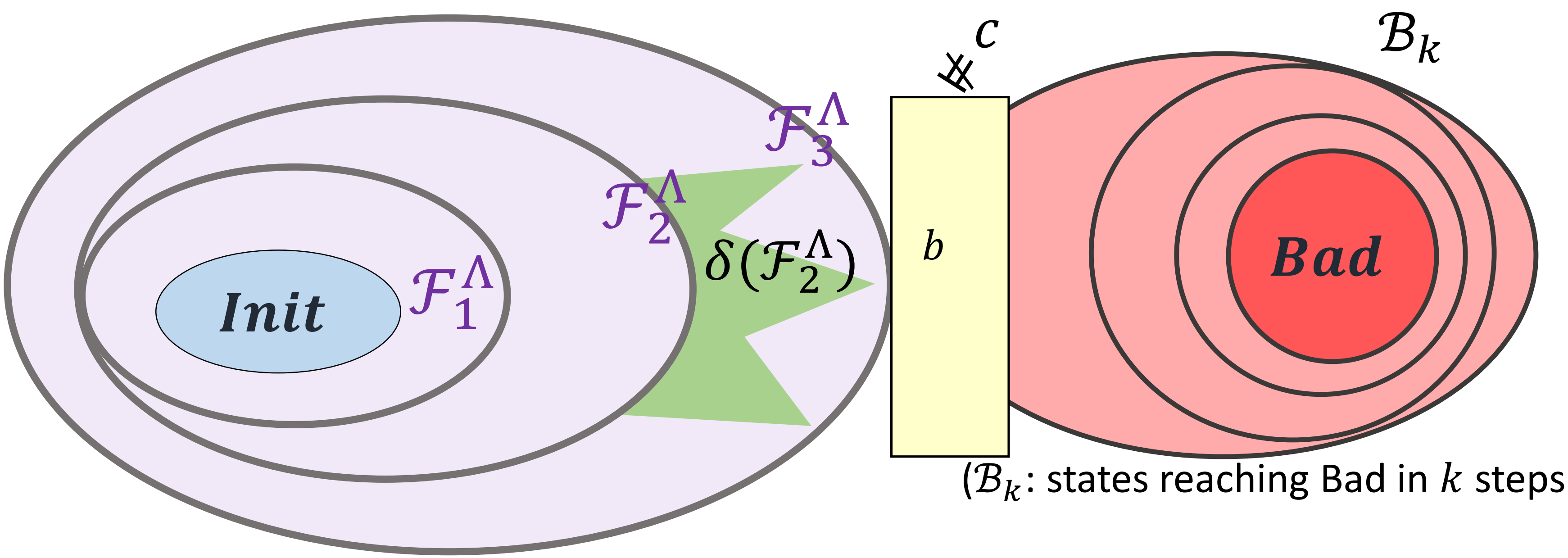
`while` $\mathcal{F}_{i+1}^{\Lambda} \neq \mathcal{F}_i^{\Lambda}$:

$\mathcal{F}_{i+1}^{\Lambda} \coloneqq$ conjunction of **all** clauses c s.t.

$\delta(\mathcal{F}_i^{\Lambda}) \cup \mathcal{F}_i^{\Lambda} \subseteq c$ and

$b \nvDash c$ for **any** $b \in \mathcal{B}_k$

k is a parameter



($\mathcal{B}_k$: states reaching Bad in k steps)

Λ-PDR's Frames

Eager PDR

$\mathcal{F}_0^{\Lambda} \coloneqq Init;\ i \coloneqq 0$

`while` $\mathcal{F}_{i+1}^{\Lambda} \neq \mathcal{F}_i^{\Lambda}$:

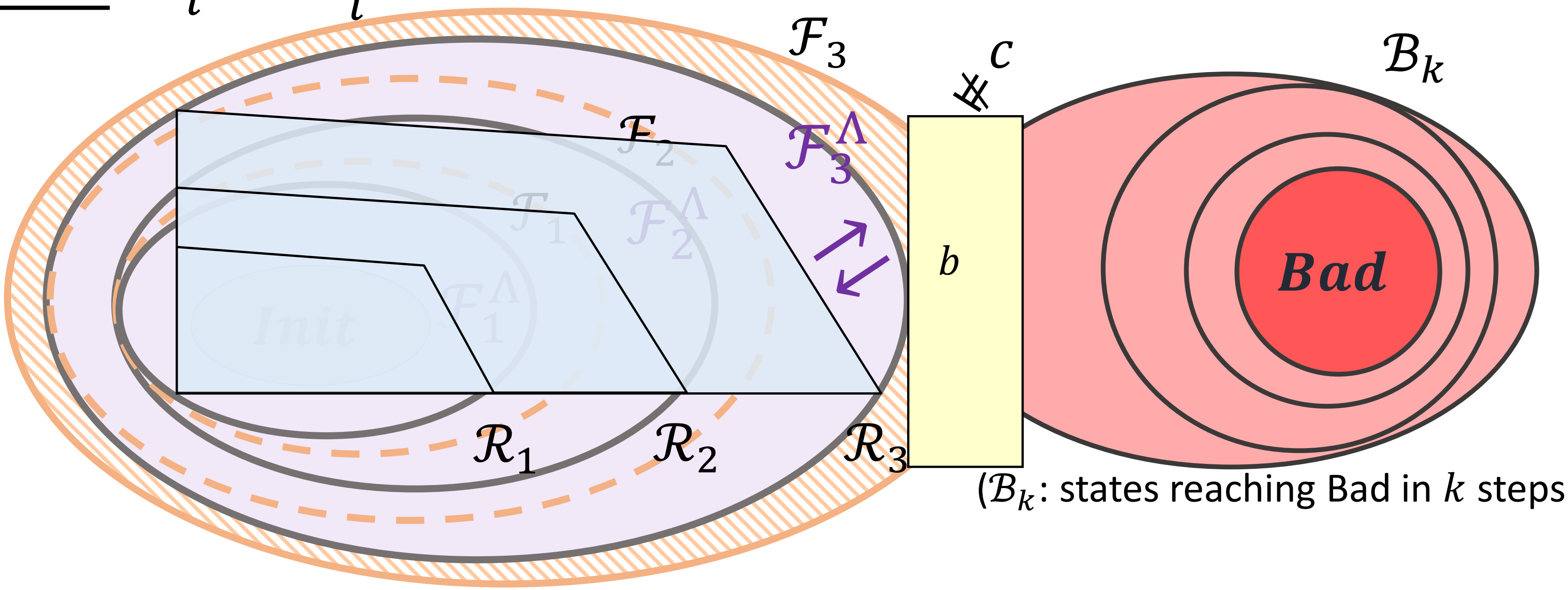
$\mathcal{F}_{i+1}^{\Lambda} \coloneqq$ conjunction of **all** clauses c s.t.

$\delta(\mathcal{F}_i^{\Lambda}) \cup \mathcal{F}_i^{\Lambda} \subseteq c$ and

$b \not\models c$ for **any** $b \in \mathcal{B}_k$

k is a parameter

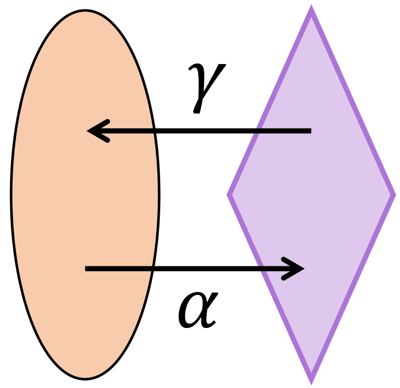
Claim: $\mathcal{F}_i^{\Lambda} \subseteq \mathcal{F}_i^{\mathrm{pdr}}$



Key Ideas and Results

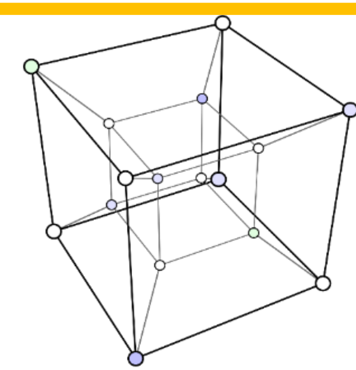
Λ-PDR: an algorithm that lower bounds the overapproximation of PDR

Λ-PDR =



Abstract Interpretation

\+



Monotone Theory

- Exponential gap between #frames in Λ-PDR and exact forward reachability, interpolation

The same overapproximation is present in PDR

Monotone Formulas

Def: ψ is *monotone* $\iff$

ψ has a DNF representation where all variables appear positively

$$v \vDash \psi \implies v[x_i \mapsto 1] \vDash \psi$$

b-Monotone Formulas

<u>Def</u>: ψ is b-*monotone* $\iff$

ψ has a DNF representation where all variables are in polarity opposite to b

$b = 00\ldots00$

$((x_n = \boxed{1}) \wedge (x_1 = \boxed{1})) \vee (x_0 = \boxed{1})$ $00\ldots00$-monotone

$((x_n = 1) \wedge (x_1 = \boxed{0})) \vee (x_0 = 1)$

not $00\ldots00$-monotone

[Inf. Comput. '95] Exact Learning Boolean Function via the Monotone Theory. Bshouty

b-Monotone Formulas

<u>Def</u>: ψ is b-*monotone* $\iff$

ψ has a DNF representation where all variables are in polarity opposite to b

$b = 00 \dots 10$

$((x_n = 1) \wedge (x_1 = 1)) \vee (x_0 = 1)$

not $00 \dots 10$-monotone

$((x_n = 1) \wedge (x_1 = 0)) \vee (x_0 = 1)$ $00 \dots 10$-monotone

[Inf. Comput. '95] Exact Learning Boolean Function via the Monotone Theory. Bshouty

b-Monotone Formulas

Def: ψ is *b-monotone* $\iff$

ψ has a DNF representation where all variables are in polarity opposite to b

$$\boxed{v \vDash \psi \quad \Longrightarrow \quad v[x_i \mapsto \neg b[x_i]] \vDash \psi}$$

ψ is closed under "walking away" from b in the Hamming cube

[Inf. Comput. '95] Exact Learning Boolean Function via the Monotone Theory. Bshouty

Monotone Span

Def: ψ is b-*monotone* $\iff$

ψ has a DNF representation where

all variables are in polarity $\text{opposite to } b$

Def:

$$\psi \in \mathrm{MSpan}(\mathcal{B}_k) \iff \psi \equiv c_1 \wedge \cdots \wedge c_t,$$

$$\text{for some clauses } c_1, \ldots, c_t \text{ s.t.}$$

$$\forall i.\ c_i \text{ is } b_i\text{-monotone for some } b_i \in \mathcal{B}_k$$

Monotone Span

Def: ψ is b-*monotone* $\iff$

ψ has a DNF representation where all variables are in polarity $\text{opposite to } b$

In particular: $\mathrm{MSpan}(\{b\}) = \{\psi \mid \psi$ is b–monotone$\}$

$\mathrm{MSpan}(\mathcal{B}_k) = \bigotimes_{b\in\mathcal{B}_k} \mathrm{MSpan}(\{b\})$

Def:

$$\psi \in \mathrm{MSpan}(\mathcal{B}_k) \iff \psi \equiv c_1 \wedge \cdots \wedge c_t,$$

$\text{for some clauses } c_1, \ldots, c_t \text{ s.t.}$

$\forall i.\ \ c_i \text{ is } b_i\text{-monotone for some } b_i \in \mathcal{B}_k$

Monotone Span and PDR

Def: ψ is b-*monotone* $\iff$

ψ has a DNF representation where all variables are in polarity opposite to b

Observation: a **clause** c is b-monotone iff $b \nvDash$ c

$\Rightarrow$ clause c inferred by PDR/Λ-PDR is b-monotone w.r.t. the proof obligation $b \in \mathcal{B}_k$

Def:

$$\psi \in \mathrm{MSpan}(\mathcal{B}_k) \iff \psi \equiv c_1 \wedge \cdots \wedge c_t,$$

$$\text{for some clauses } c_1, \ldots, c_t \text{ s.t.}$$

$$b_i \nvDash c_i \qquad \forall i. \boxed{c_i \text{ is } b_i\text{-monotone}} \text{ for some } b_i \in \mathcal{B}_k$$

$\Rightarrow$ $\mathcal{F}_i^{\mathrm{pdr}}, \mathcal{F}_i^{\Lambda} \in \mathrm{MSpan}(\mathcal{B}_k)$

(Λ-)PDR's Frames

A sequence of formulas $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \ldots$

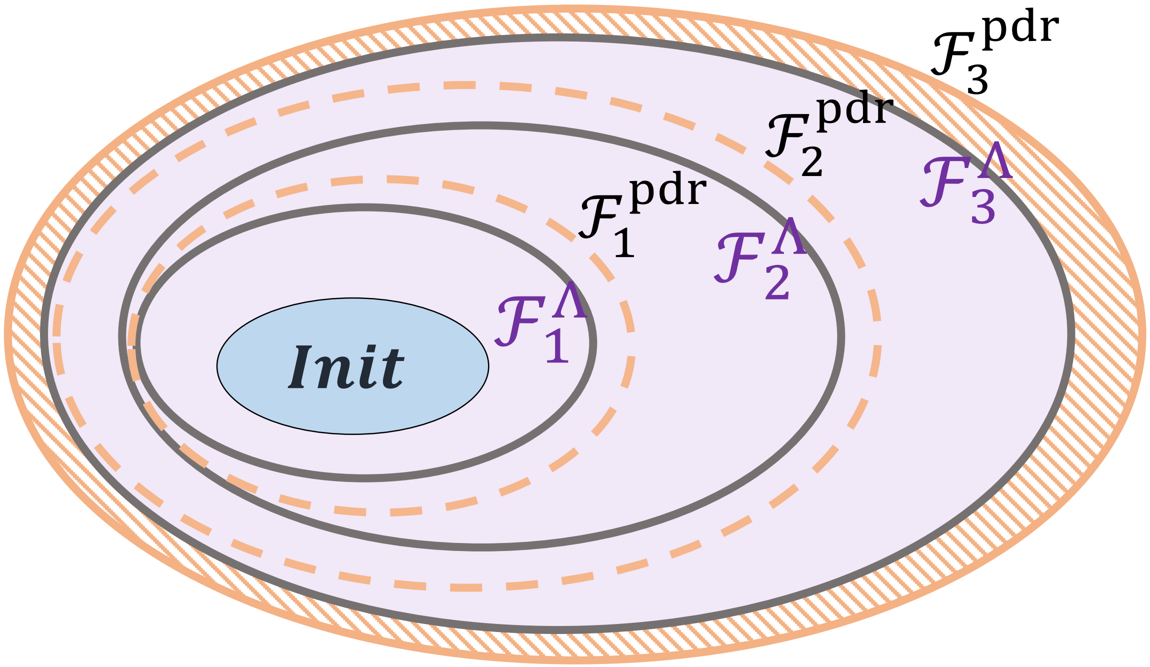
(1) $\mathcal{F}_0 = \mathrm{Init}$

(2) $\mathcal{F}_i \Longrightarrow \mathcal{F}_{i+1}$

(3) $\delta(\mathcal{F}_i) \Longrightarrow \mathcal{F}_{i+1}$

(4) $\mathcal{F}_i \Longrightarrow \neg\mathrm{Bad}$

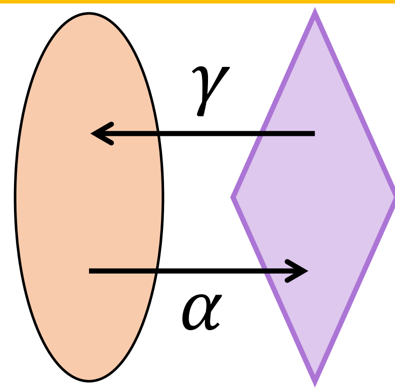
(5) $\mathcal{F}_i \in \mathrm{MSpan}(\mathcal{B}_k)$



Key Ideas and Results

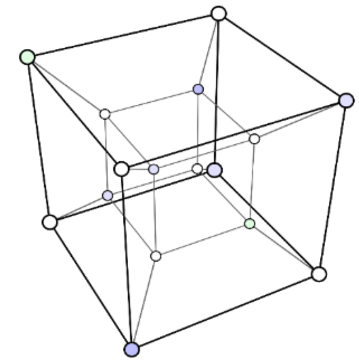
Λ-PDR: an algorithm that lower bounds the overapproximation of PDR

Λ-PDR =



Abstract Interpretation

+



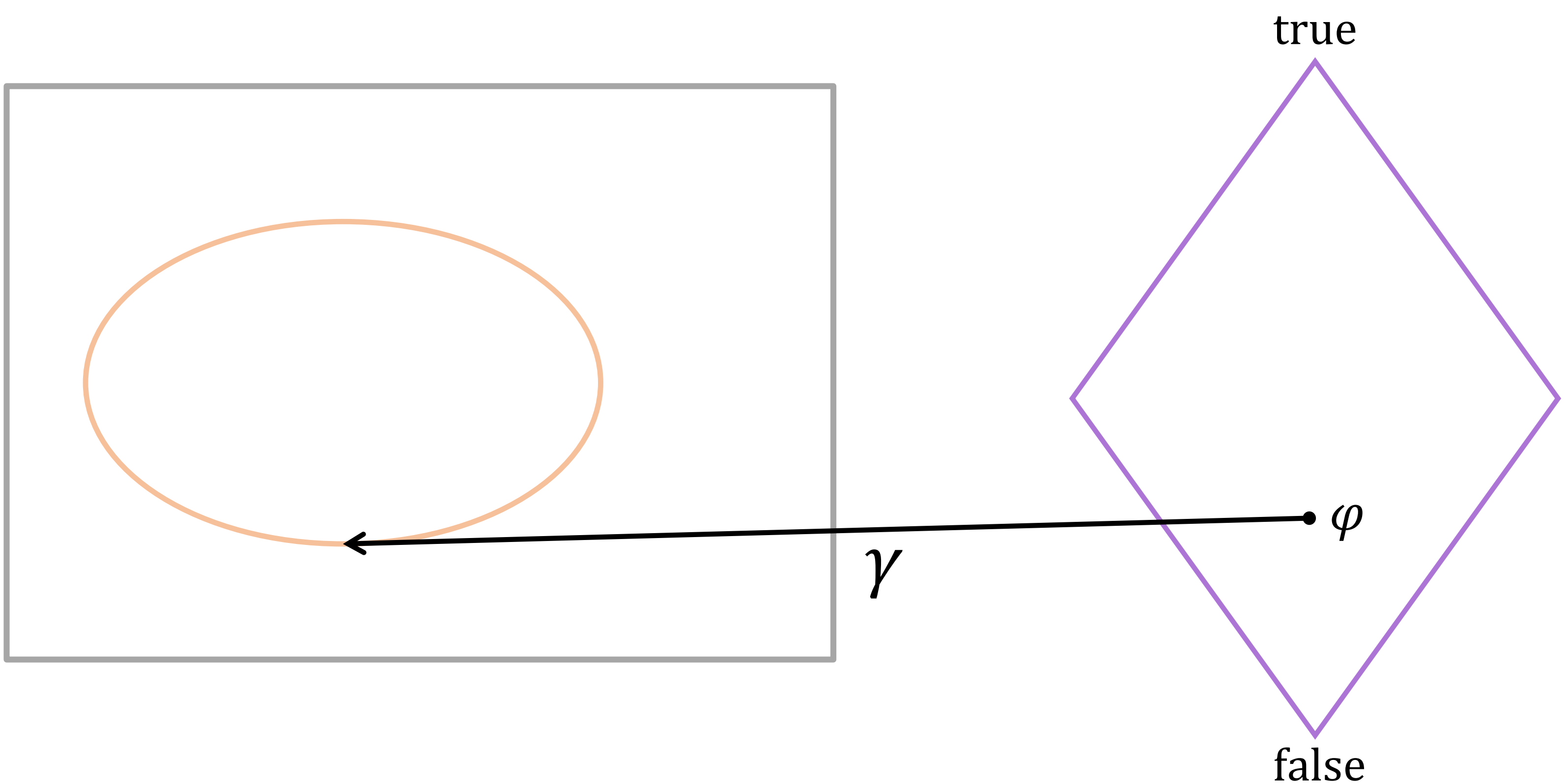
Monotone Theory

- Exponential gap between #frames in Λ-PDR and exact forward reachability, interpolation

The same overapproximation is present in PDR

Monotone Span Abstract Domain

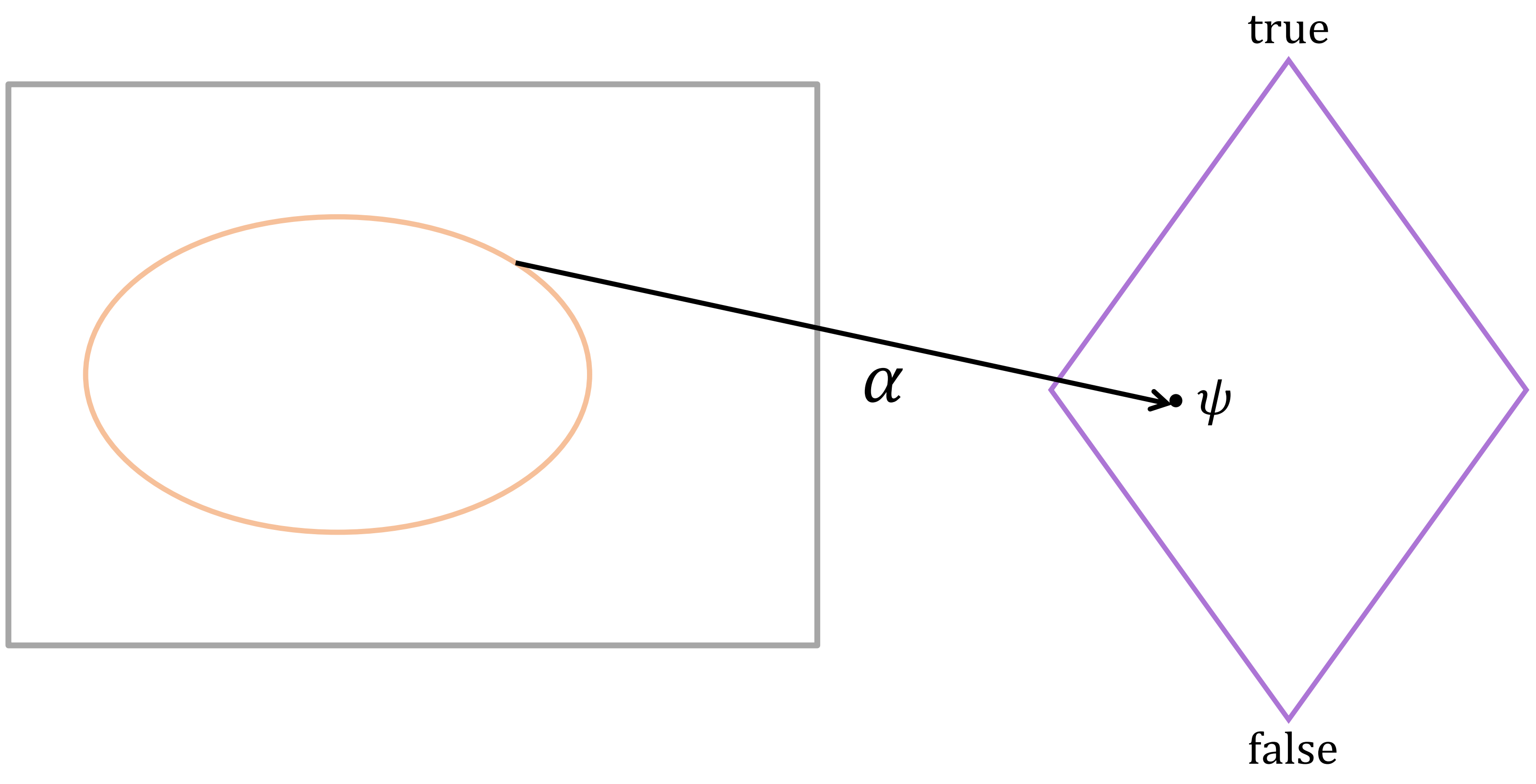
$$\langle C, \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\rightleftarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$



Monotone Span Abstract Domain

$$\langle C, \subseteq \rangle \underset{\boxed{\alpha}}{\overset{\gamma}{\rightleftarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$

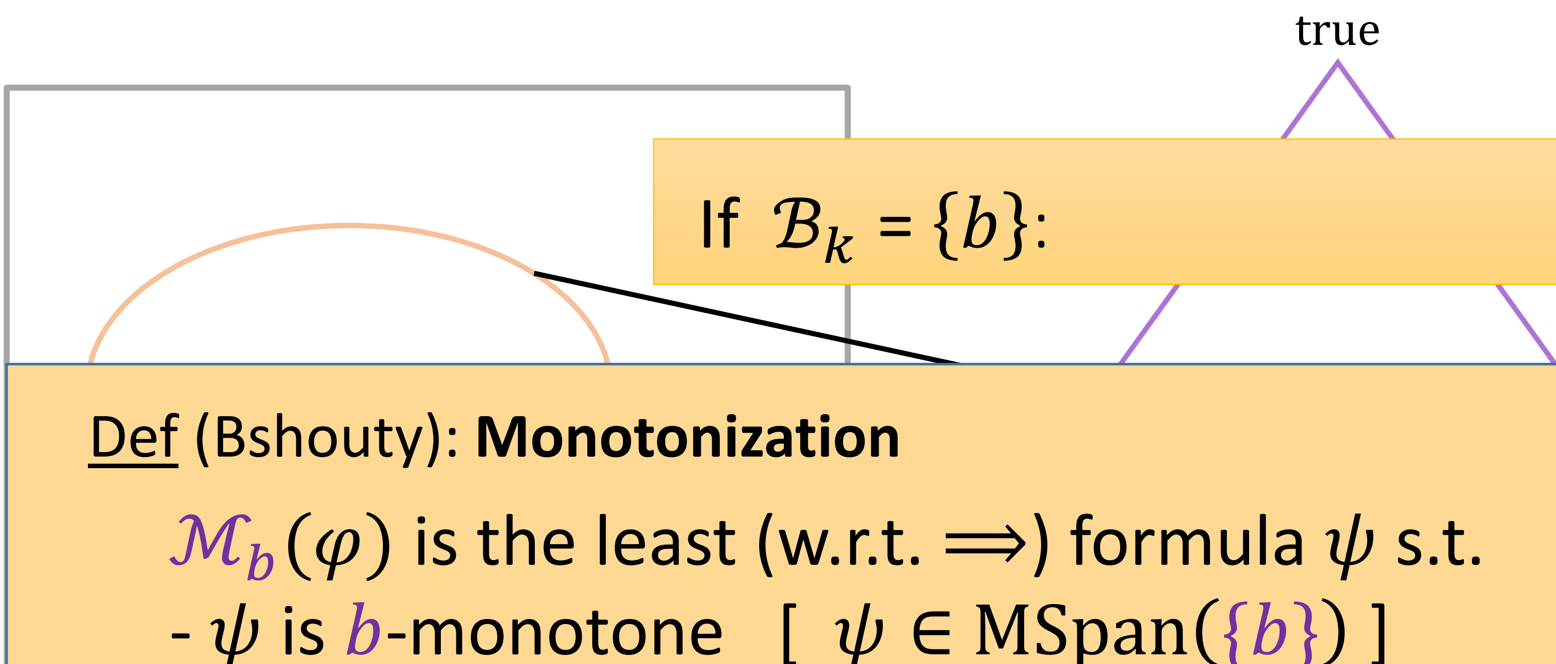
$\alpha(S) =$ least (w.r.t. $\Longrightarrow$) formula $\psi \in \mathrm{MSpan}(\mathcal{B}_k)$ s.t. $S \Longrightarrow \psi$



Monotone Span Abstract Domain

$$\langle C, \subseteq \rangle \underset{\boxed{\alpha}}{\overset{\gamma}{\leftrightarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$

$\alpha(S) =$ least (w.r.t. $\Longrightarrow$) formula $\psi \in \mathrm{MSpan}(\mathcal{B}_k)$ s.t. $S \Longrightarrow \psi$



If $\mathcal{B}_k = \{b\}$:

Def (Bshouty): **Monotonization**

$\mathcal{M}_b(\varphi)$ is the least (w.r.t. $\Longrightarrow$) formula ψ s.t.

- ψ is b-monotone [$\psi \in \mathrm{MSpan}(\{b\})$]
- $\varphi \Longrightarrow \psi$.

Monotone Span Abstract Domain

$$\langle C, \subseteq \rangle \underset{\boxed{\alpha}}{\overset{\gamma}{\leftrightarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$

$\alpha(S) =$ least (w.r.t. $\Longrightarrow$) formula $\psi \in \mathrm{MSpan}(\mathcal{B}_k)$ s.t. $S \Longrightarrow \psi$

true

If $\mathcal{B}_k = \{b\}$: $\alpha(S) = \mathcal{M}_b(S)$

<u>Def</u> (Bshouty): **Monotonization**

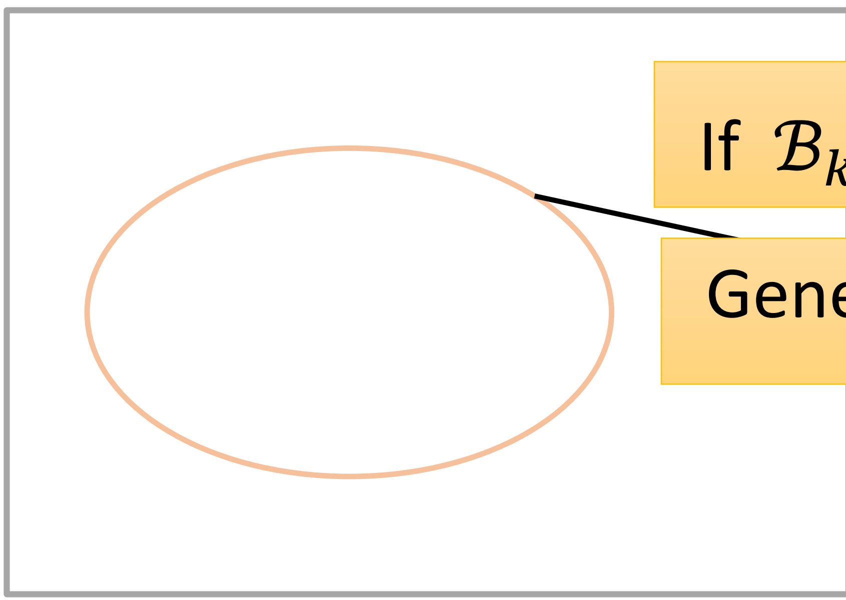
$\mathcal{M}_b(\varphi)$ is the least (w.r.t. $\Longrightarrow$) formula ψ s.t.

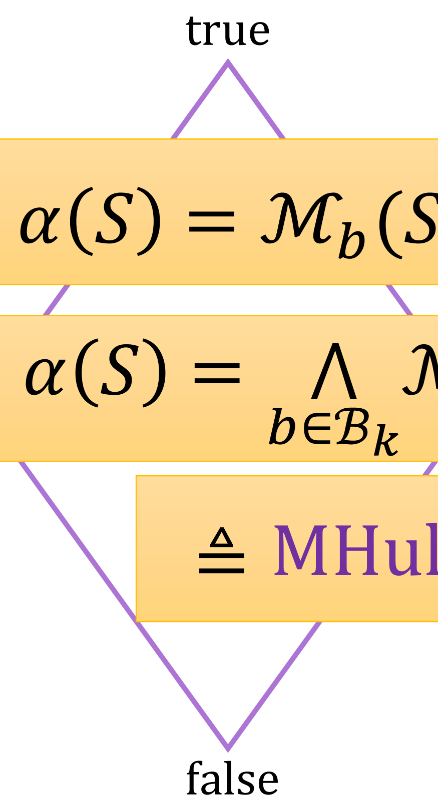
- ψ is b-monotone [$\psi \in \mathrm{MSpan}(\{b\})$]
- $\varphi \Longrightarrow \psi$.

Monotone Span Abstract Domain

$$\langle C, \subseteq \rangle \underset{\boxed{\alpha}}{\overset{\gamma}{\leftrightarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$

$\alpha(S) =$ least (w.r.t. $\Longrightarrow$) formula $\psi \in \mathrm{MSpan}(\mathcal{B}_k)$ s.t. $S \Longrightarrow \psi$





If $\mathcal{B}_k = \{b\}$: $\alpha(S) = \mathcal{M}_b(S)$

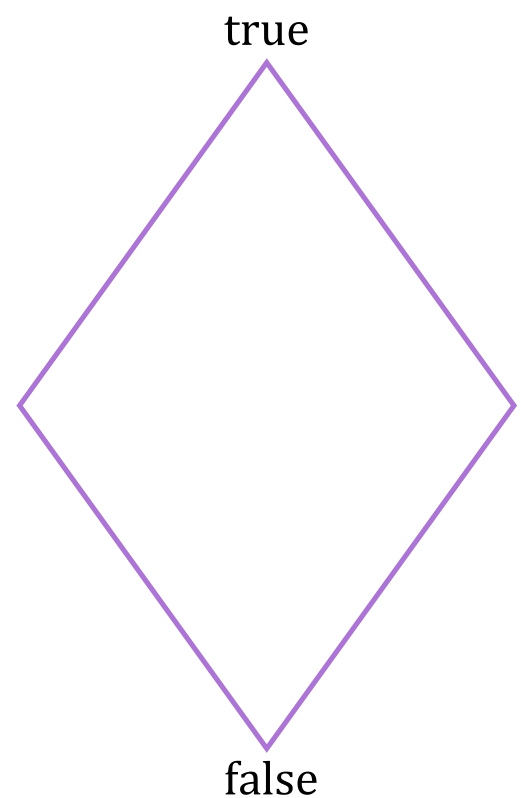
General $\mathcal{B}_k$: $\alpha(S) = \bigwedge_{b \in \mathcal{B}_k} \mathcal{M}_b(S)$

$\triangleq \mathrm{MHull}_{\mathcal{B}_k}(S)$

Abstract Interpretation in MSpan

$$\langle C, \subseteq \rangle \underset{\alpha = \mathrm{MHull}_{\mathcal{B}_k}}{\overset{\gamma}{\leftrightarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$

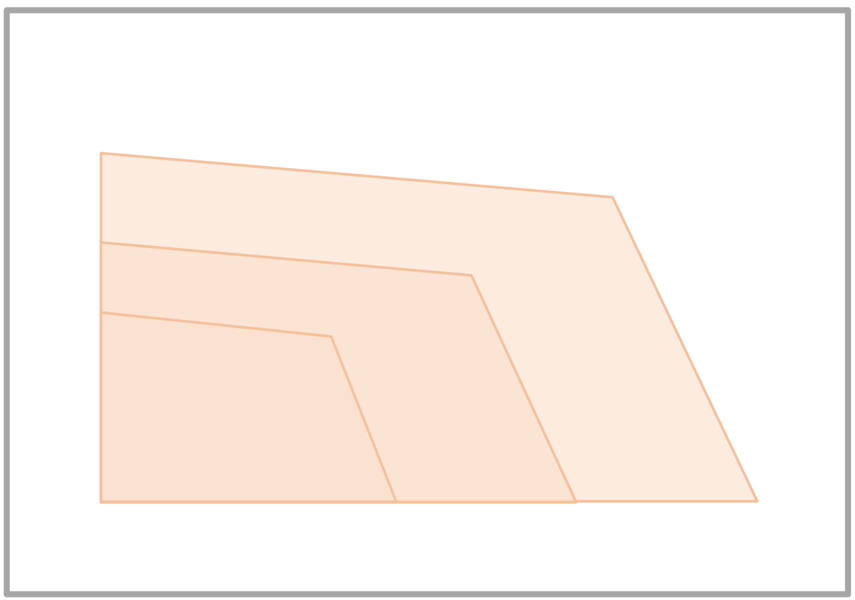
$\tau(S) = S \cup \delta(S)$

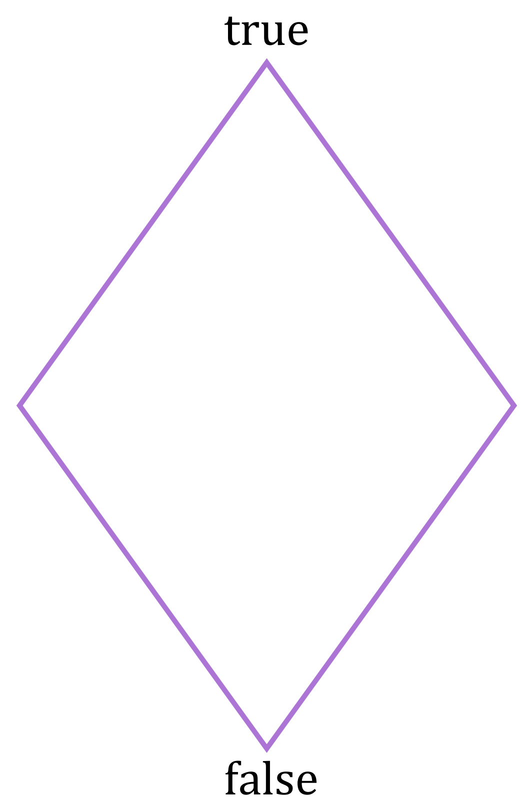


Abstract Interpretation in MSpan

$$\langle C, \subseteq \rangle \underset{\alpha\, =\, \mathrm{MHull}_{\mathcal{B}_k}}{\overset{\gamma}{\leftrightarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$

$\tau(S) = S \cup \delta(S)$



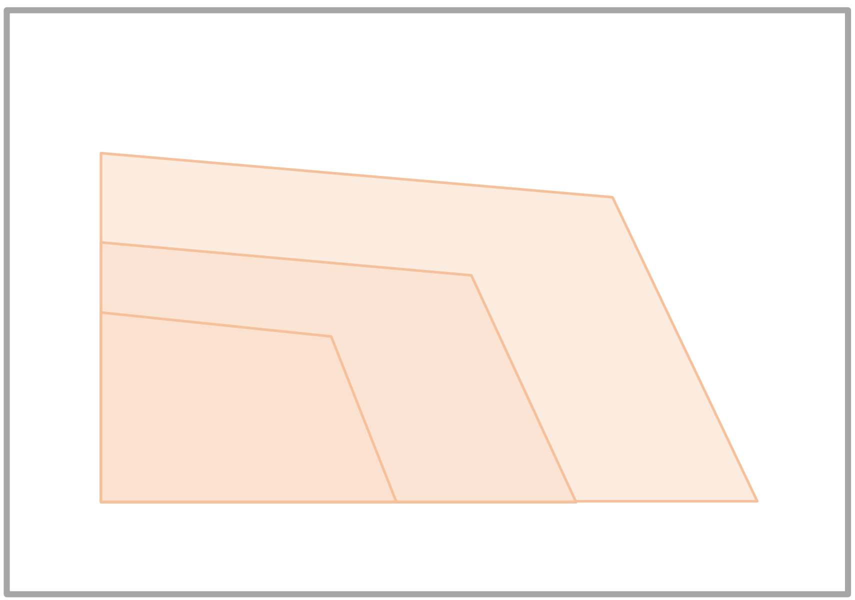


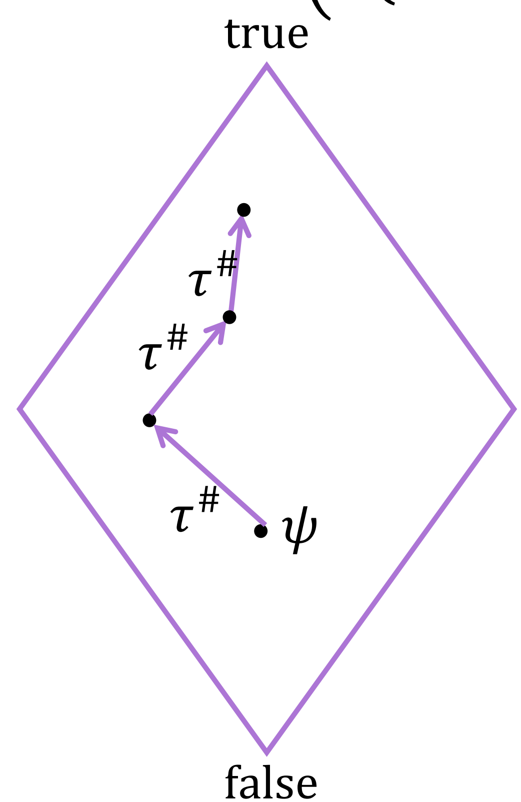
Abstract Interpretation in MSpan

$$\langle C, \subseteq \rangle \underset{\alpha\, =\, \mathrm{MHull}_{\mathcal{B}_k}}{\overset{\gamma}{\leftrightarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$

$\tau(S) = S \cup \delta(S)$

$\tau^{\#}(\psi) = \alpha\Big(\tau\big(\gamma(\psi)\big)\Big)$



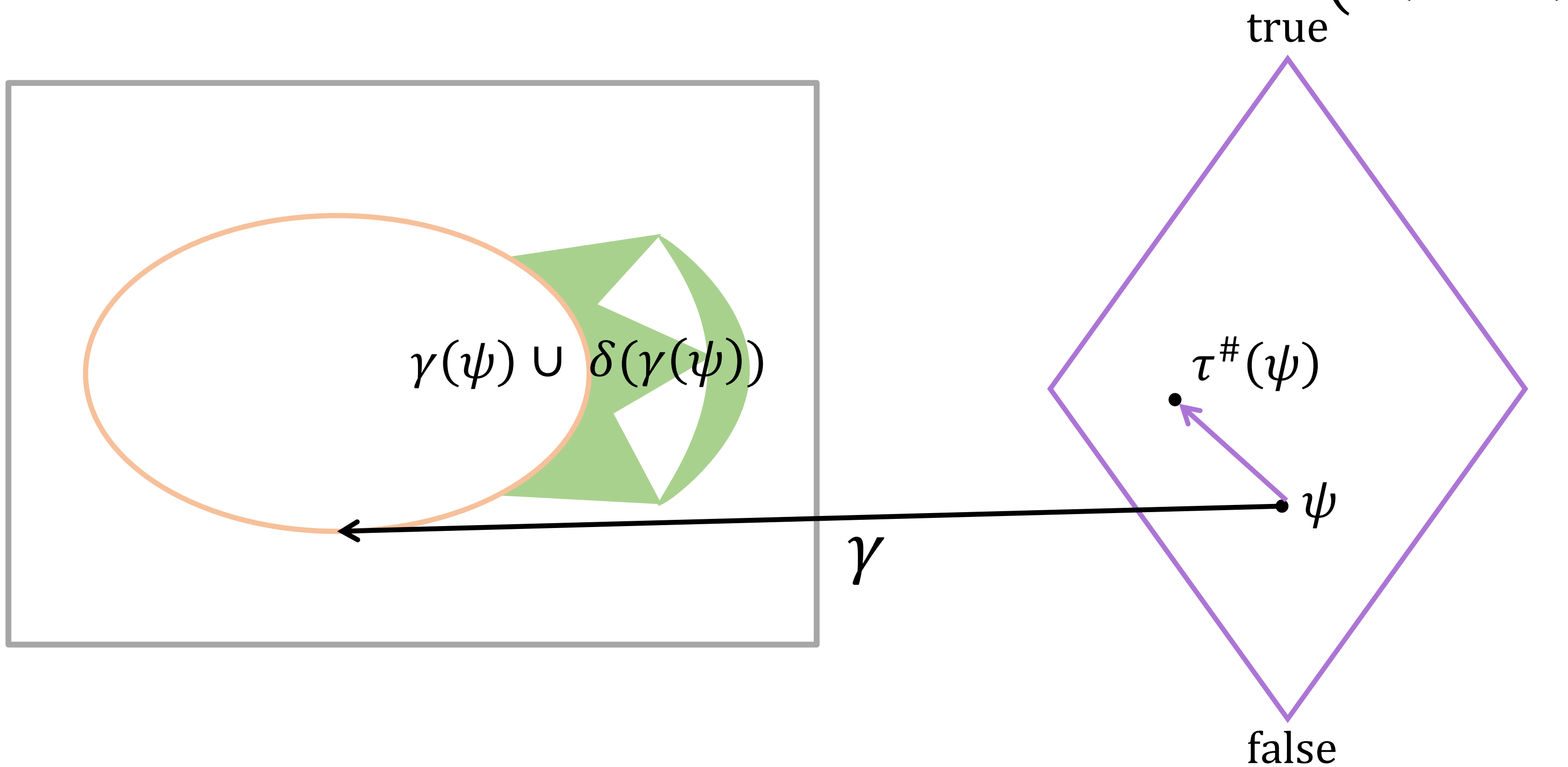


Best Abstract Transformer

$$\langle C, \subseteq \rangle \underset{\alpha \,=\, \mathrm{MHull}_{\mathcal{B}_k}}{\overset{\gamma}{\rightleftarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$

$\tau(S) = S \cup \delta(S)$

$\tau^{\#}(\psi) = \alpha\Big(\tau\big(\gamma(\psi)\big)\Big)$



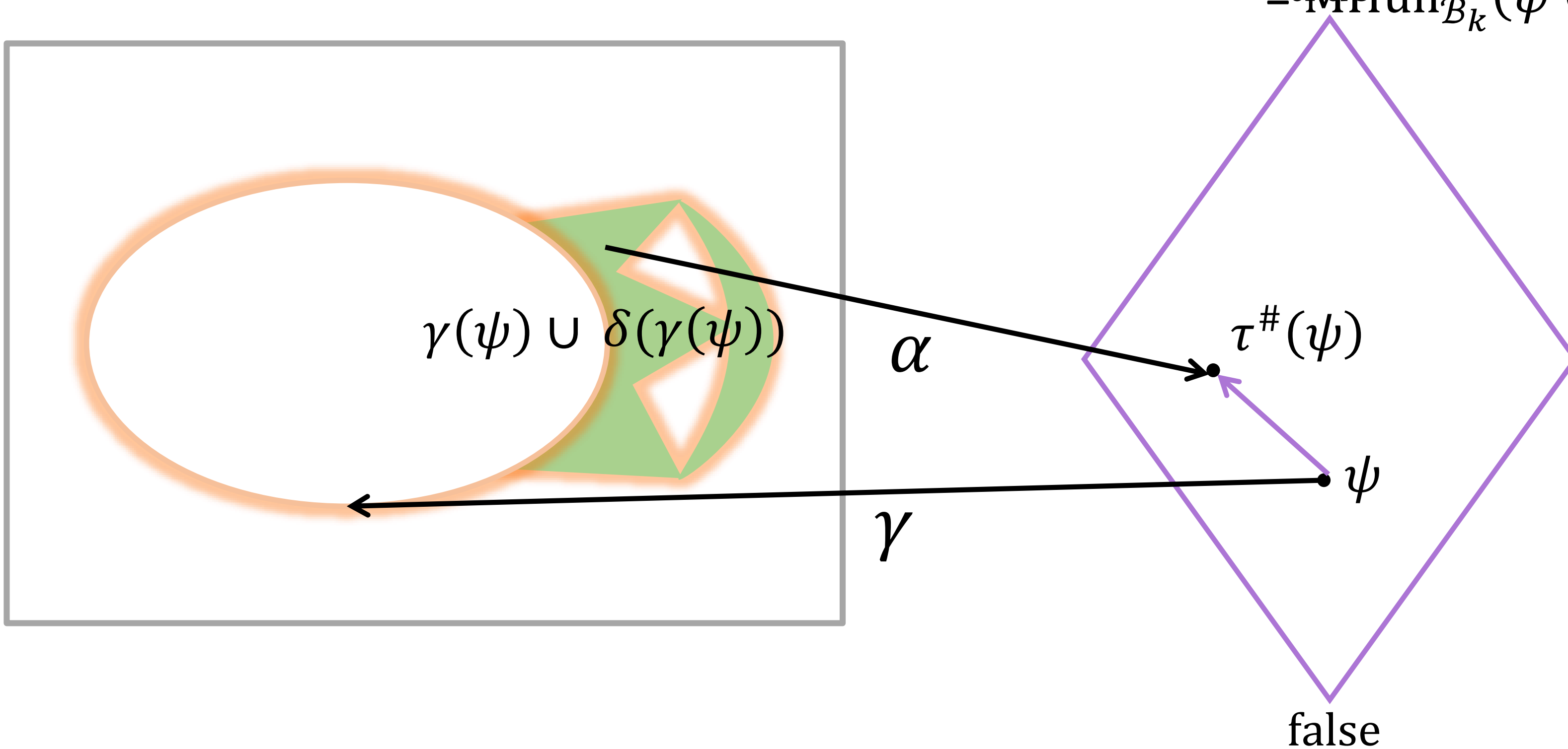
Best Abstract Transformer

$$\langle C, \subseteq \rangle \underset{\alpha\, =\, \mathrm{MHull}_{\mathcal{B}_k}}{\overset{\gamma}{\leftrightarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$

$\tau(S) = S \cup \delta(S)$

$\tau^{\#}(\psi) = \alpha\Big(\tau\big(\gamma(\psi)\big)\Big)$

$= \mathrm{MHull}_{\mathcal{B}_k}(\psi \cup \delta(\psi))$



Best Abstract Transformer

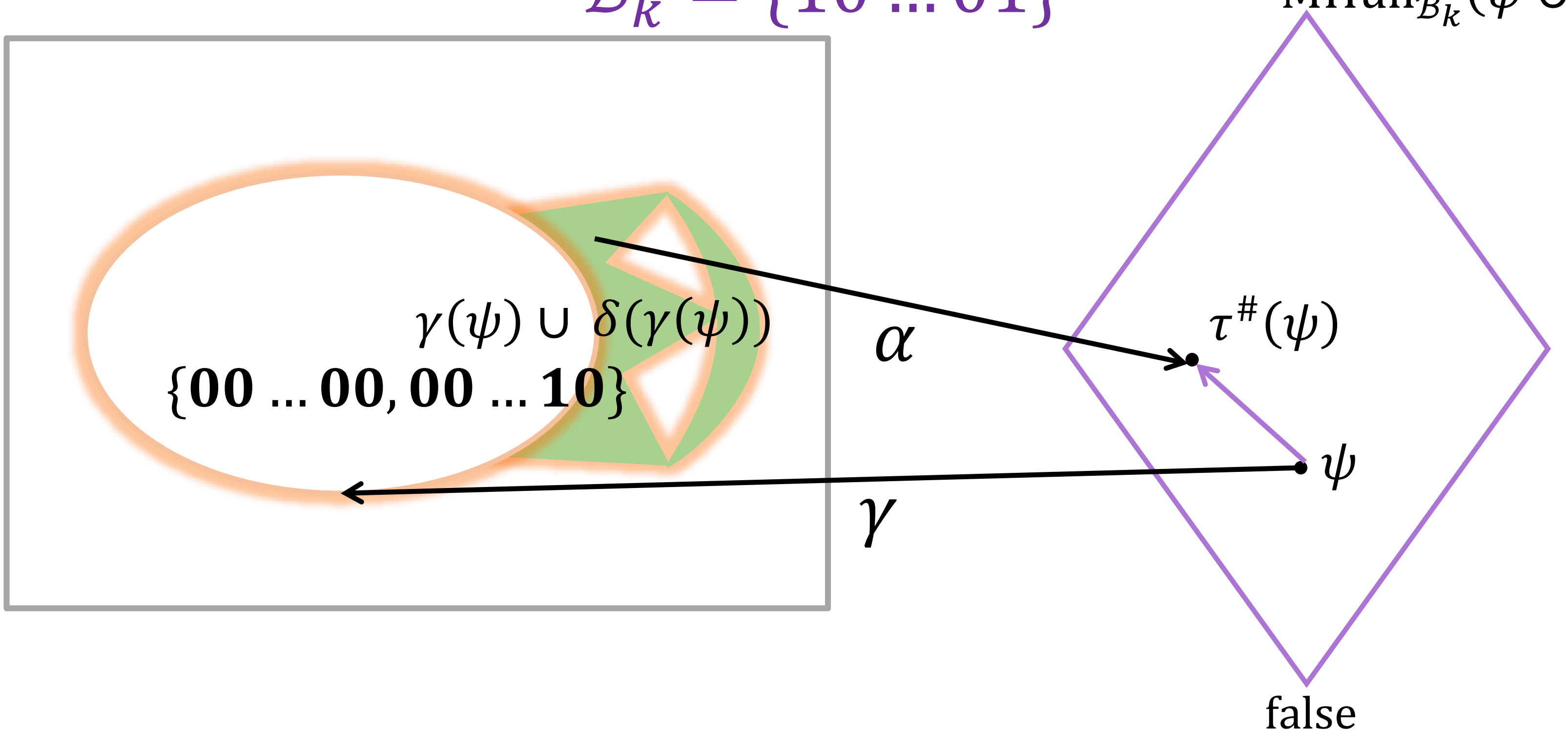
$$\langle C, \subseteq \rangle \underset{\alpha = \mathrm{MHull}_{\mathcal{B}_k}}{\overset{\gamma}{\rightleftarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$

$\tau(S) = S \cup \delta(S)$

$\tau^{\#}(\psi) = \alpha\left(\tau\left(\gamma(\psi)\right)\right)$

$= \mathrm{MHull}_{\mathcal{B}_k}(\psi \cup \delta(\psi))$

$\mathcal{B}_k = \{10 \ldots 01\}$



Best Abstract Transformer

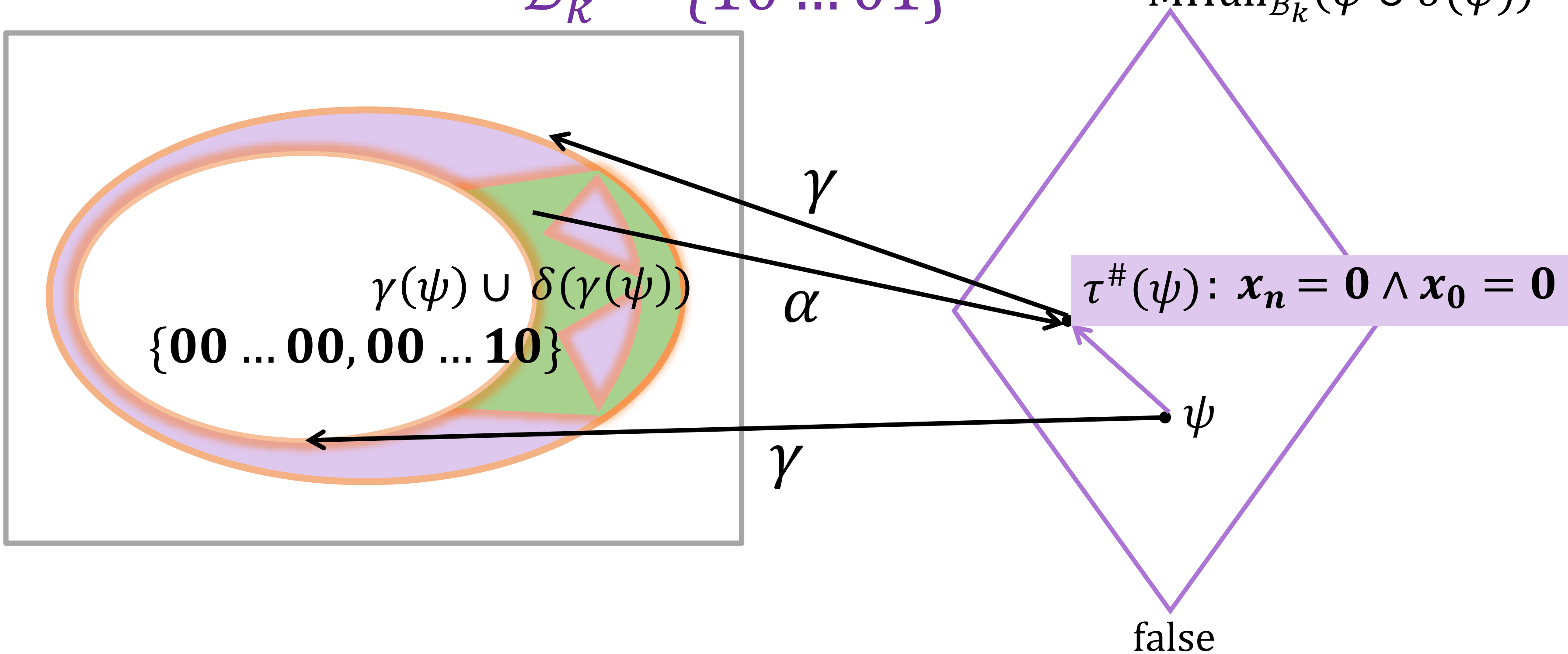
$$\langle C, \subseteq \rangle \underset{\alpha \,=\, \mathrm{MHull}_{\mathcal{B}_k}}{\overset{\gamma}{\leftrightarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$

$\tau(S) = S \cup \delta(S)$

$\tau^{\#}(\psi) = \alpha\Big(\tau\big(\gamma(\psi)\big)\Big)$

$= \mathrm{MHull}_{\mathcal{B}_k}(\psi \cup \delta(\psi))$

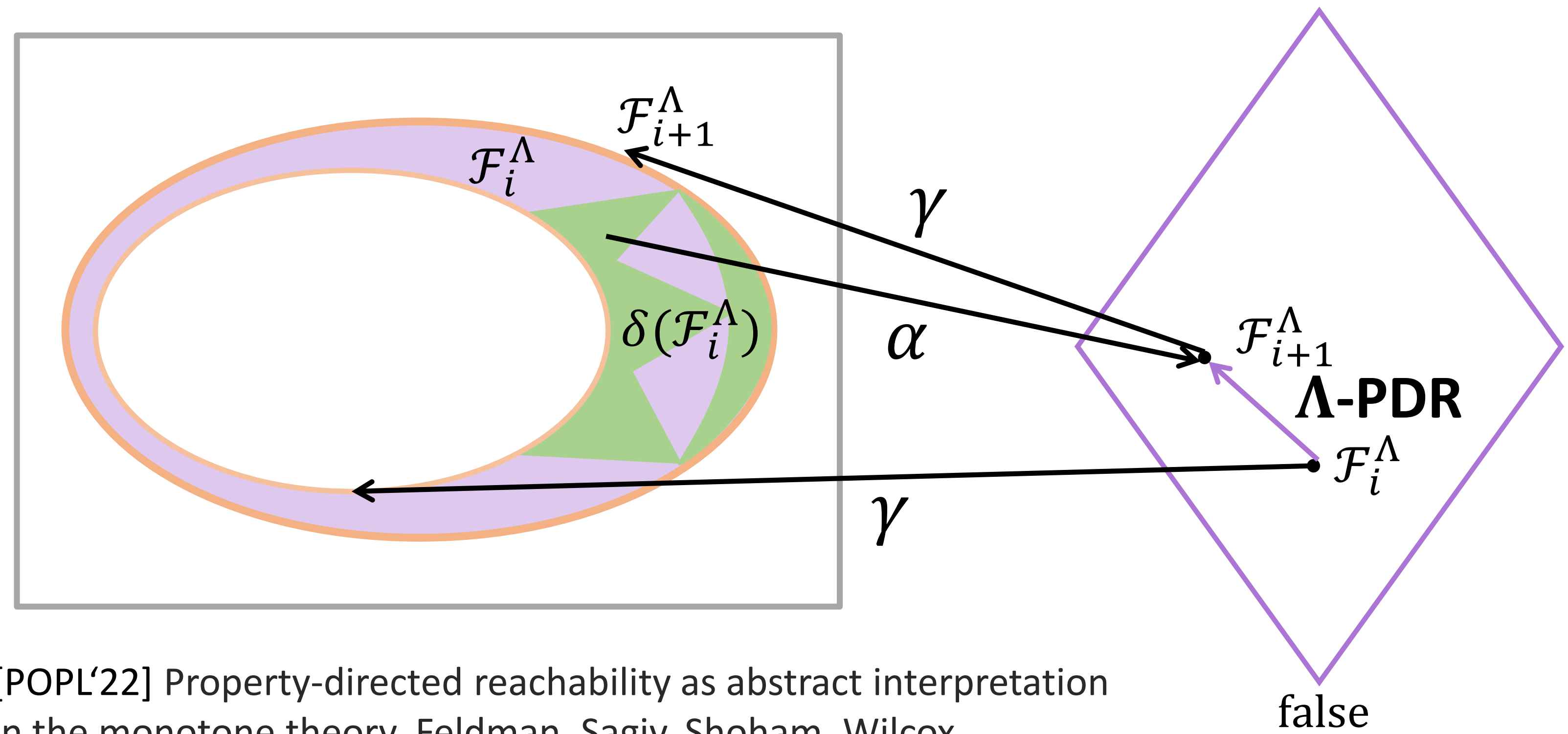
$\mathcal{B}_k = \{10 \ldots 01\}$



Best Abstract Transformer

$$\langle C, \subseteq \rangle \underset{\alpha\, =\, \mathrm{MHull}_{\mathcal{B}_k}}{\overset{\gamma}{\rightleftarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$

<u>Thm</u>: $\mathcal{F}_{i+1}^{\Lambda} = \tau^{\#}(\mathcal{F}_i^{\Lambda}) \quad \left(= \mathrm{MHull}_{\mathcal{B}_k}\left(\mathcal{F}_i^{\Lambda} \cup \delta\left(\mathcal{F}_i^{\Lambda}\right)\right)\right)$



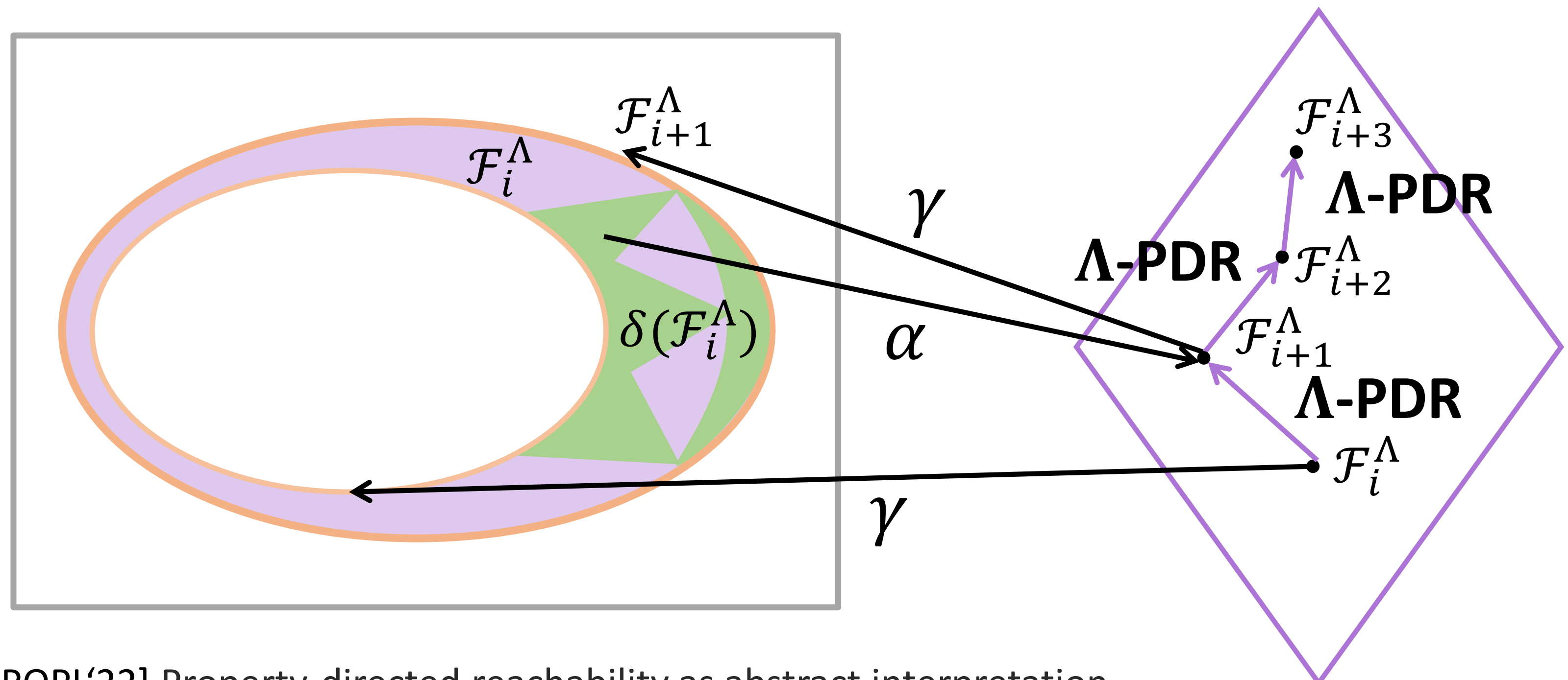
[POPL'22] Property-directed reachability as abstract interpretation in the monotone theory. Feldman, Sagiv, Shoham, Wilcox

Λ-PDR as Abstract Interpretation

$$\langle C, \subseteq \rangle \underset{\alpha \,=\, \mathrm{MHull}_{\mathcal{B}_k}}{\overset{\gamma}{\leftrightarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$

Thm: $\mathcal{F}_{i+1}^{\Lambda} = \tau^{\#}(\mathcal{F}_i^{\Lambda}) \quad \left(= \mathrm{MHull}_{\mathcal{B}_k}\left(\mathcal{F}_i^{\Lambda} \cup \delta\left(\mathcal{F}_i^{\Lambda}\right)\right)\right)$

Corollary: Frames in Λ-PDR are **Kleene iterations**



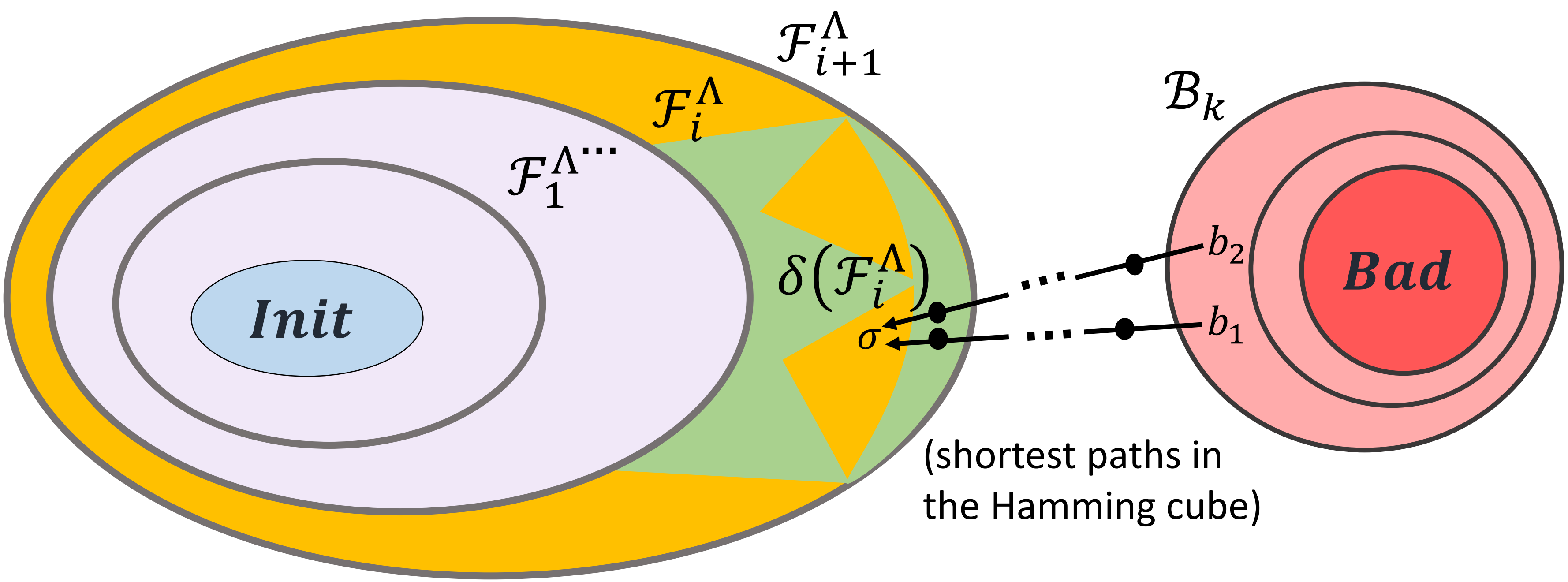
[POPL'22] Property-directed reachability as abstract interpretation in the monotone theory. Feldman, Sagiv, Shoham, Wilcox

Abstraction & Monotone Theory

$$\langle C, \subseteq \rangle \underset{\alpha \,=\, \mathrm{MHull}_{\mathcal{B}_k}}{\overset{\gamma}{\rightleftarrows}} \langle \mathrm{MSpan}(\mathcal{B}_k), \Longrightarrow \rangle$$

<u>Thm</u>: $\mathcal{F}_{i+1}^{\wedge} = \tau^{\#}(\mathcal{F}_i^{\wedge})$ $\quad \left(= \boxed{\mathrm{MHull}_{\mathcal{B}_k}}\left(\mathcal{F}_i^{\wedge} \cup \delta\left(\mathcal{F}_i^{\wedge}\right)\right)\right)$

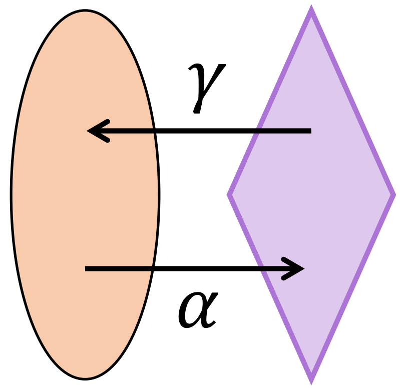
<u>Def:</u> $\mathrm{MHull}_{\mathcal{B}_k}(S) = \bigwedge_{b \in \mathcal{B}_k} \mathcal{M}_b(S)$



Key Ideas and Results

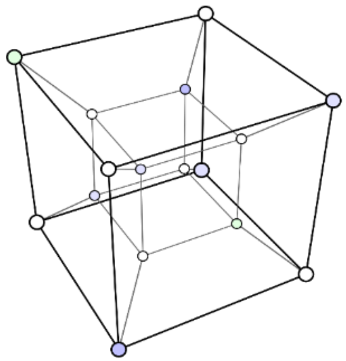
Λ-PDR: an algorithm that lower bounds the overapproximation of PDR

Λ-PDR =



Abstract Interpretation

+

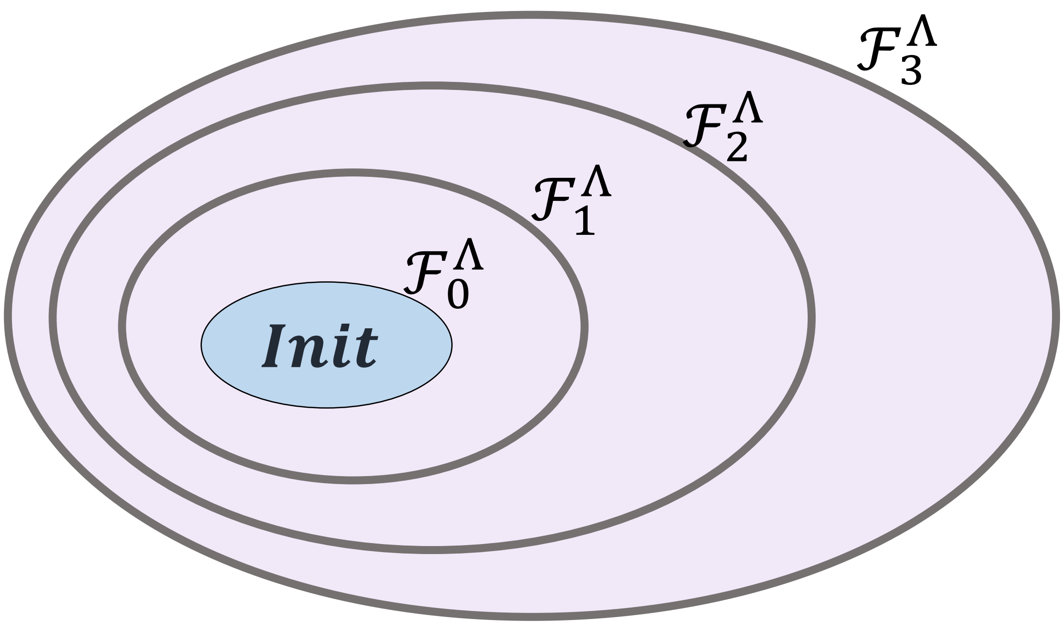


Monotone Theory

- Exponential gap between #frames in Λ-PDR and exact forward reachability, interpolation

Λ-PDR's Frames Overapproximate

$\mathcal{F}_0^{\Lambda} : (x_n, \ldots, x_0) = 00 \ldots 00$ $(k = 0, \mathcal{B}_0 = \text{Bad})$



Init:
$(x_n, \ldots, x_0) \coloneqq 00 \ldots 00$

Bad:
$(x_n, \ldots, x_0) = 10 \ldots 01$

δ:
$(x_n, \ldots, x_0) \coloneqq (x_n, \ldots, x_0) + 00 \ldots 10$

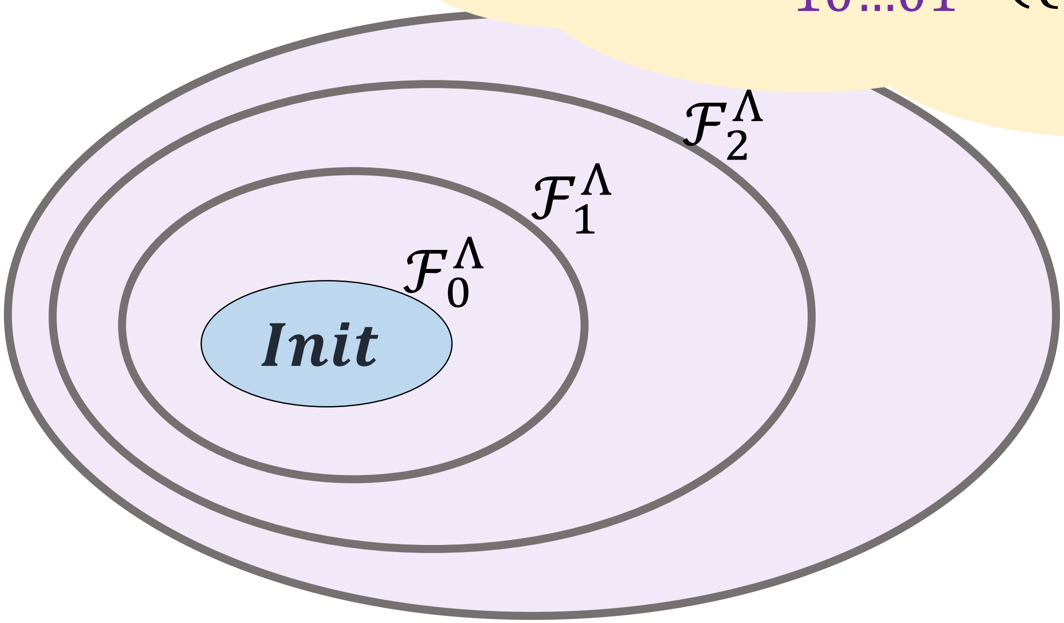
Λ-PDR's Frames Overapproximate

$\mathcal{F}_0^\Lambda : (x_n, \dots, x_0) = 00 \dots 00$ $\quad(k = 0, \mathcal{B}_0 = \mathrm{Bad})$

$\mathcal{F}_1^\Lambda : \; x_n = 0 \wedge \; x_0 = 0$

$= \; \mathrm{MHull}_{10\dots01} \; (\mathcal{F}_0^\Lambda \cup \delta(\mathcal{F}_0^\Lambda))$

$= \; \mathrm{MHull}_{10\dots01} \; (\{00 \dots 00, 00 \dots 10\})$



$(x_n, \dots, x_0) := 00 \dots 00$

Bad:

$(x_n, \dots, x_0) \; = \; 10 \dots 01$

δ:

$(x_n, \dots, x_0) \; := (x_n, \dots, x_0) + 00 \dots 10$

Λ-PDR's Frames Overapproximate

$\mathcal{F}_0^{\Lambda} : (x_n, \ldots, x_0) = 00 \ldots 00$ $\qquad (k = 0, \mathcal{B}_0 = \text{Bad})$

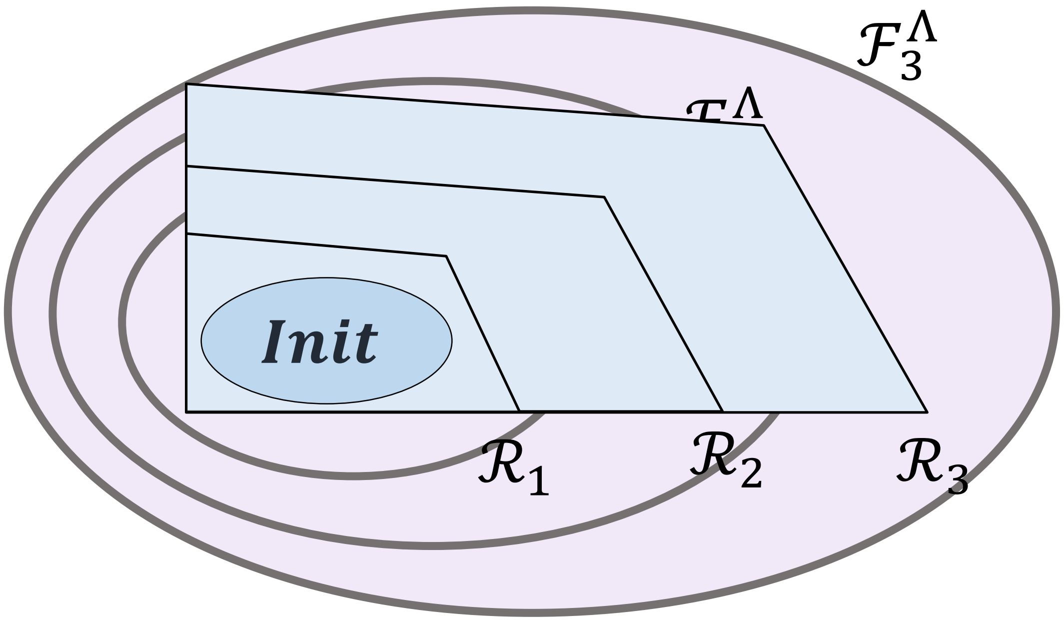
$\mathcal{F}_1^{\Lambda} : \ x_n = 0 \wedge \ x_0 = 0$

$\mathcal{F}_3^{\Lambda} \equiv \mathcal{F}_2^{\Lambda} : \ \ x_0 = 0$

Also from (dual) model-based interpolation

Thm: **exponential gap ↑↓**

$\mathcal{R}_i \ : \ (x_n, \ldots, x_0) \leq 2i \ \wedge \ x_0 = 0$



Init:
$(x_n, \ldots, x_0) \ \coloneqq \ 00 \ldots 00$

Bad:
$(x_n, \ldots, x_0) \ = \ 10 \ldots 01$

δ:
$(x_n, \ldots, x_0) \coloneqq (x_n, \ldots, x_0) + 00 \ldots 10$

PDR's Frames Overapproximate

$\mathcal{F}_0^{\wedge} : (x_n, \dots, x_0) = 00 \dots 00$

$(k = 0, \mathcal{B}_0 = \text{Bad})$

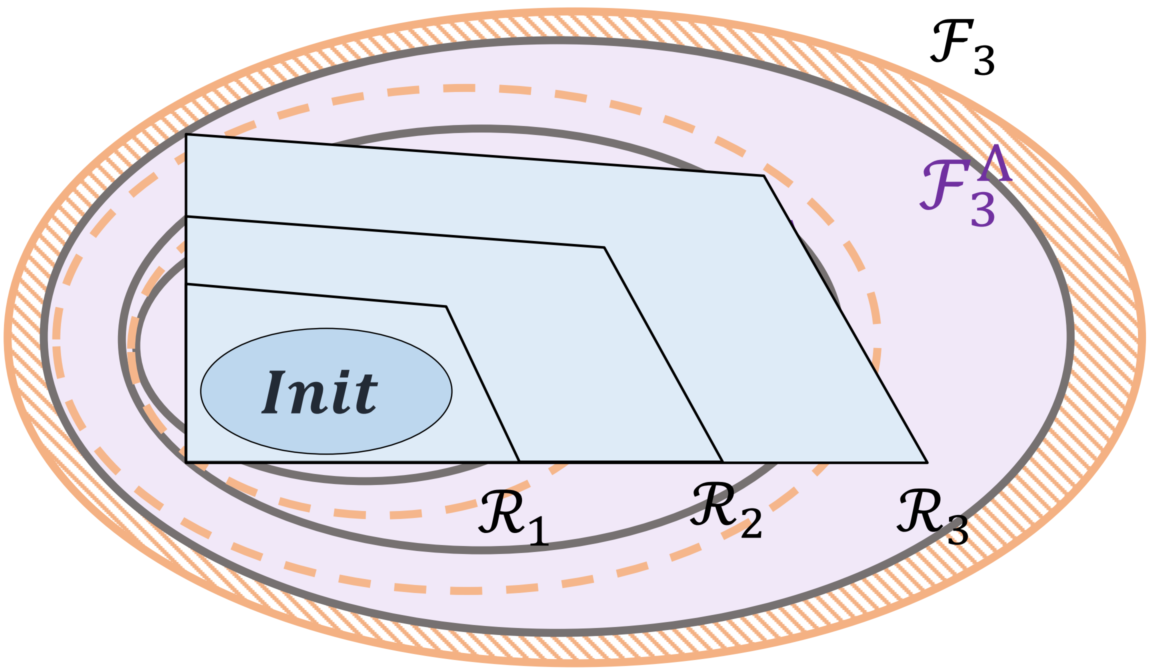
$\mathcal{F}_1^{\wedge} : \; x_n = 0 \wedge x_0 = 0$

$\mathcal{F}_3^{\wedge} \equiv \mathcal{F}_2^{\wedge} : \; x_0 = 0$

$$\boxed{\mathcal{F}_i^{\wedge} \subseteq \mathcal{F}_i^{\text{pdr}}}$$

Thm: **exponential gap ↑↓**

$\mathcal{R}_i : \; (x_n, \dots, x_0) \leq 2i \wedge x_0 = 0$



Init:
$(x_n, \dots, x_0) \coloneqq 00 \dots 00$

Bad:
$(x_n, \dots, x_0) = 10 \dots 01$

δ:
$(x_n, \dots, x_0) \coloneqq (x_n, \dots, x_0) + 00 \dots 10$

Convergence Bounds for Λ-PDR

$\mathcal{F}_0^\Lambda : (x_n, \dots, x_0) = 00 \dots 00$ $\quad (k = 0, \mathcal{B}_0 = \text{Bad})$

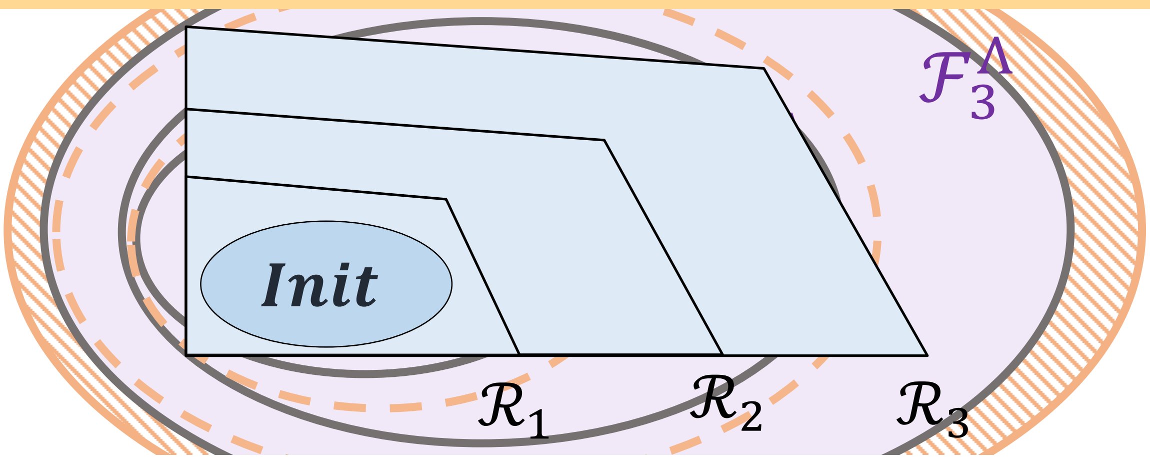
$\mathcal{F}_1^\Lambda : \; x_n = 0 \wedge x_0 = 0$

$\mathcal{F}_3^\Lambda \equiv \mathcal{F}_2^\Lambda : \; x_0 = 0$

Bound on **#frames** of Λ-PDR: linear in $\left|\mathcal{M}_{(!b,b)}(\delta)\right|_{\text{dnf}}$

Bound on **time** of Λ-PDR: quadratic in $\left|\mathcal{M}_{(!b,b)}(\delta)\right|_{\text{dnf}}$

* new algorithm for computing $\mathcal{M}_b(\psi)$



Bad:

$(x_n, \dots, x_0) = 10 \dots 01$

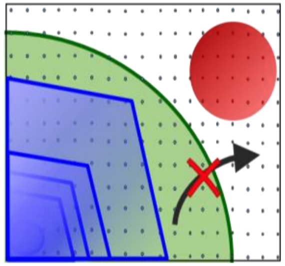
δ:

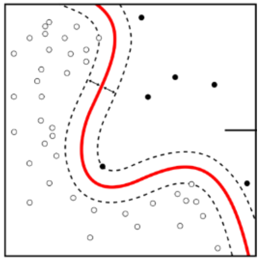
$(x_n, \dots, x_0) := (x_n, \dots, x_0) + 00 \dots 10$

[POPL'22] PDR as abstract interpretation in the monotone theory. Feldman, Sagiv, Shoham, Wilcox
[SAS'22] Invariant Inference With Provable Complexity From the Monotone Theory. Feldman, Shoham

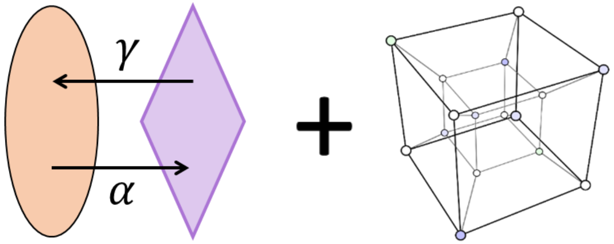
Summary

- PDR uses rich SAT queries
  - Exponential complexity gap between Hoare queries (PDR) and Inductiveness queries (ICE)





- Overapproximation in PDR
  - Λ-PDR:



  - Lower bound on the overapproximation of PDR
  - Exponential gap between #frames in Λ-PDR and exact forward reachability, interpolation