# The Lattice-Theoretic Essence of Property Directed Reachability Analysis

Mayuko Kori[1,2], Natsuki Urabe[2], Shin-ya Katsumata[2], Kohei Suenaga[3], and Ichiro Hasuo[1,2]

1 The Graduate University for Advanced Studies (SOKENDAI)
2 National Institute of Informatics
3 Kyoto University

7th Aug in CAV'22

# PDR is ...

Knaster
– Tarski

Kleene

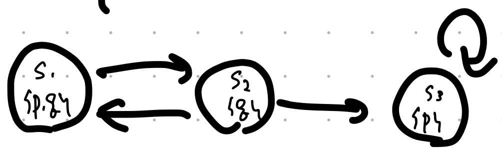
# Property Directed Reachability (PDR)

Model checking Algorithm
to verify / refute safety problems. of state transition systems.

- known for its performance
  - hardware verification [Bradley, VMCAI'11]
  - used in Spacer [Hoder+, CAV'11] as part of Z3 [MouraB. TACAS'08]

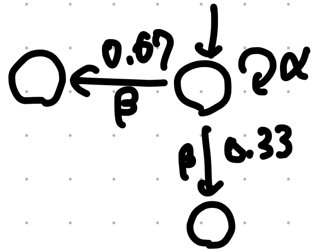
- many variants:
  1. IC3/PDR [Bradley, VMCAI'11], [Een+, FMCAD'11]:
     original one.
  2. Reverse PDR, fbPDR [Seufert & Scholl, DATE'18,'19]:
     search direction is reversed

  Model checking
  Kripke structures




  3. PrIC3 [Batz+, CAV'20]:
     probabilistic extension.

  Model checking
  MDPs



# Our Contribution: Mathematical Abstraction of PDR

① Lattice-Theoretical abstraction

Our algorithm

**LT-PDR**

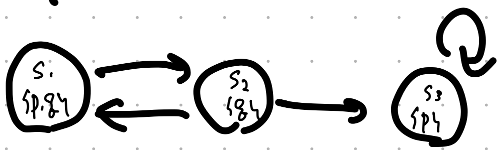
for $\mu F \overset{?}{\leq} \alpha$ in L

Essence: Combination of Knaster-Tarski & Kleene

② Instantiate the theory

**IC3/PDR**

for Kripke structures



**Instance 1**

**Instance 2**

for Markov Decision Processes

**Instance 3**

for Markov Reward Models

# Outline

1. LT-PDR — lattice-theoretic essences
    - Positive & Negative
    - LT-PDR — let these help each other
2. Instances of LT-PDR for quantitative model checking
3. Categorical classification of PDR variants
4. From mathematical abstraction to programming abstraction.
5. Experiments

# Overview of LT-PDR

## Positive Side

Thm. (Knaster-Tarski)
Prefixed points form a complete lattice.

$\mu F \le \alpha$
$\iff$ There exists $X$ s.t. $FX \le X \le \alpha$

$X_0 \le X_1 \le \cdots \le X_{n-2} \le X_{n-1}$
s.t. $FX_i \le X_{i+1}$ and $X_{n-2} \le \alpha$

verifies

## Negative Side

Thm. (Kleene)
$\mu F = \bigvee_{n \in \omega} F^n \bot$

$\mu F \not\le \alpha$
$\iff$ There exists $n$. s.t. $F^n \bot \not\le \alpha$

$(C_1, \cdots, C_{n-1})$
s.t. $C_{i+1} \le FC_i$. $C_{n-1} \not\le \alpha$

refutes

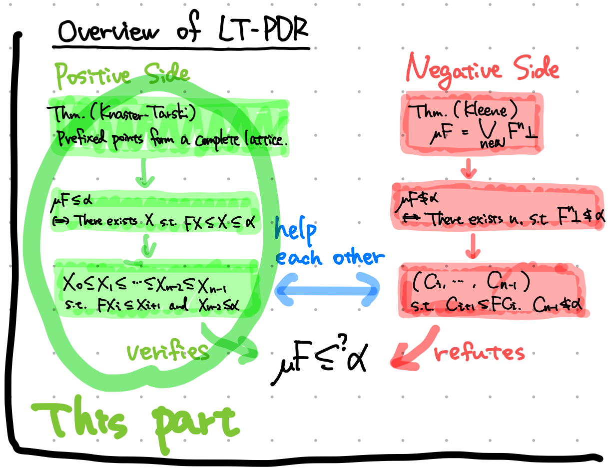
help each other

$$\mu F \overset{?}{\le} \alpha$$

# Outline

1. LT-PDR — lattice-theoretic essences
   - Positive & Negative
   - LT-PDR — let these help each other
2. Instances of LT-PDR for quantitative model checking
3. Categorical classification of PDR variants
4. From mathematical abstraction to programming abstraction.
5. Experiments

# Positive LT-PDR




... aims to verify $\mu F \leq \alpha$.
Comes from KT thm.

Thm. (Knaster-Tarski)

$\mu F \leq \alpha \Leftrightarrow$ There exists $\underbrace{X}_{\text{invariant}}$ s.t. $FX \leq X \leq \alpha$

How to search for $X$ ?

- (Naively) Because $FX \leq X \Rightarrow \forall i.\ F^i \bot \leq X$,
  construct $\underbrace{\bot \leq F\bot \leq F^2\bot \leq \cdots}_{\text{small candidates of } X.}$ and check $\underbrace{F^{i+1}\bot \leq F^i\bot}_{\text{stabilized}} \leq \alpha$.

  It's difficult to find this ...

- (Positive LT-PDR)

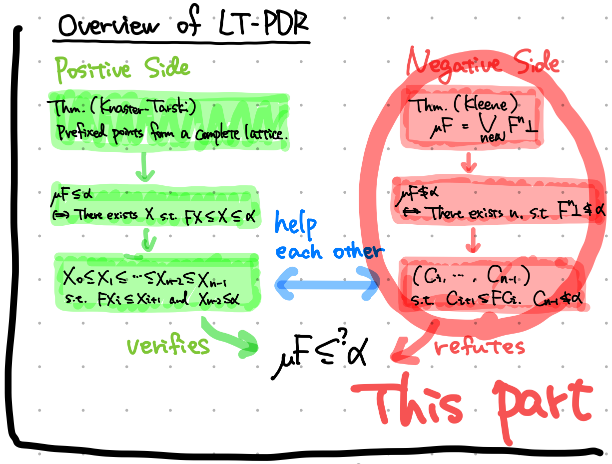
  easier to find !

  - Construct $\underbrace{X_0 \leq X_1 \leq X_2 \leq \cdots}$ $(\leq \alpha)$ and check $FX_i \leq X_{i+1}$.

    preemptively inflate $\bot \leq F\bot \leq \cdots$. It accelerates the search

  - Preemptively inflate and shrink if necessary.

# Negative LT-PDR




... aims to refute $\mu F \le \alpha$.
Comes from Kleene thm.

**Thm. (Kleene)**

$\mu F \not\le \alpha \iff$ There exists $n$ s.t. $F^n \bot \not\le \alpha$.

$\iff$ There exists $n$, $C$ s.t. $\underbrace{C \le F^n \bot}_{\text{reachable}}$ and $\underbrace{C \not\le \alpha}_{\text{unsafe}}$.

Counterexample

How to search for $C$? (in Negative LT-PDR)

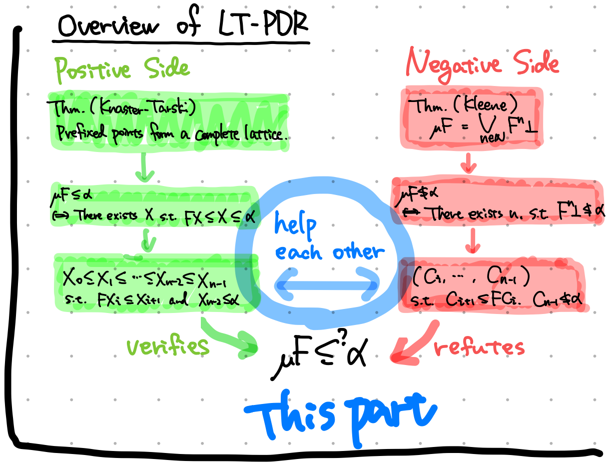
1. Pick $\boxed{C_n \not\le \alpha}$. unsafe
2. Construct $C_i, C_{i+1}, \cdots, C_n$ (Construct ←)
   s.t. $C_i \le F^i \bot \Rightarrow C_{i+1} \le F^{i+1} \bot \Rightarrow \cdots \Rightarrow C_n \le F^n \bot$

   try to prove $\boxed{C_n \le F^n \bot}$ (reachable) by backtracking
3. If $C_0 = \bot$ then $C_n$ is the one!
   (Let's hope this is the case...)

# Outline

1. LT-PDR — lattice-theoretic essences
    - Positive & Negative
    - LT-PDR — let these help each other
2. Instances of LT-PDR for quantitative model checking
3. Categorical classification of PDR variants
4. From mathematical abstraction to programming abstraction.
5. Experiments

# LT-PDR




Combination of Positive & Negative ones, helping each other.

hints for shrinking overly-inflated $X$

**Positive Side**

$X_0 \leq \cdots \leq X_{n-1}$

**Negative Side**

$(C_i, \cdots, C_{n-1})$

tells $C_0 = \bot$ will never happen

by a lattice-theoretic proposition:

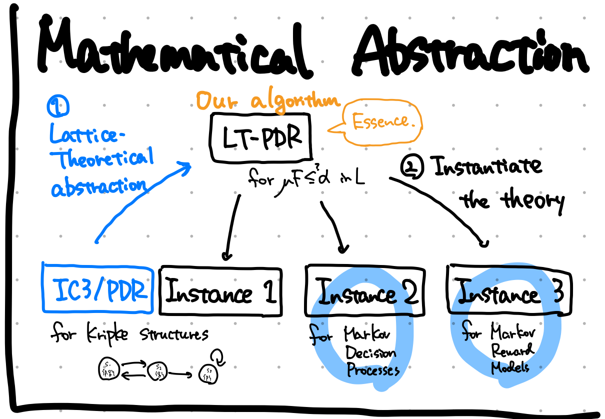
**Proposition 3.17.** *Let $C = (C_i, \ldots, C_{n-1})$ be a Kleene sequence $(2 \leq n, 0 < i \leq n-1)$ and $X = (X_0 \leq \cdots \leq X_{n-1})$ be a KT sequence. Then*

1. *$C_i \not\leq X_i$ implies that $C$ cannot be extended to a conclusive one, that is, there does not exist $C_0, \ldots, C_{i-1}$ such that $(C_0, \ldots, C_{n-1})$ is conclusive.*
2. *$C_i \not\leq FX_{i-1}$ implies that $C$ cannot be extended to a conclusive one.*
3. *There is no conclusive Kleene sequence with length $n-1$.* □

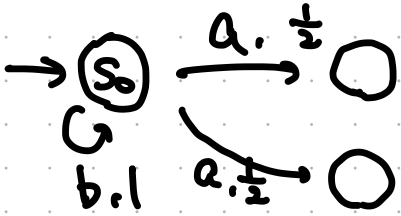
# Outline

1. LT-PDR — lattice-theoretic essences
    - Positive & Negative
    - LT-PDR — let these help each other
2. Instances of LT-PDR for quantitative model checking
3. Categorical classification of PDR variants
4. From mathematical abstraction to programming abstraction.
5. Experiments

# Instances of LT-PDR




- Instance for Markov Decision Processes

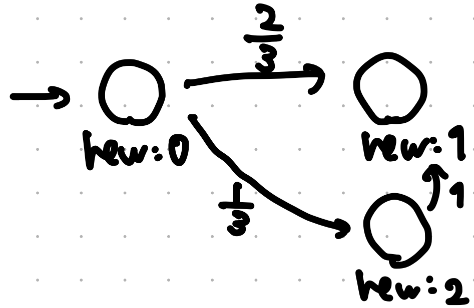



$$L := [0,1]^S$$

$$F(d : S \to [0,1]) = \left( s \mapsto \begin{cases} \max_a \sum_{s'} d(s') \cdot \delta(s)(a)(s') & \text{if } s \in \alpha \\ 1 & \text{otherwise} \end{cases} \right)$$

Similar to PrIC3 [Batz+, CAV'20]
PrIC3 needs additional check before returning False,
but our instance does not.

- Instance for Markov Reward Models.




$$L := [0,1]^S$$

$$F(d : S \to [0,1]) = \left( s \mapsto \begin{cases} rew(s) + \sum_{s' \in S} d(s') \cdot \delta(s, s') & \text{if } s \in \alpha \\ 0 & \text{otherwise} \end{cases} \right)$$

New PDR variant for MRM.

# Outline

1. LT-PDR — lattice-theoretic essences
   - Positive & Negative
   - LT-PDR — let these help each other
2. Instances of LT-PDR for quantitative model checking
3. Categorical classification of PDR variants
4. From mathematical abstraction to programming abstraction.
5. Experiments

# LT-PDR for Dynamical Systems

fibration of predicates

$$\begin{tikzcd} \dot{G} \circlearrowright \mathbb{E} \arrow[d, "p"] \\ G \circlearrowright \mathbb{B} \end{tikzcd}$$

cat. of sets & functions

$$\begin{tikzcd} \mathbb{E}_S \arrow[r, bend right, "\dot{G}_S"'] & \mathbb{E}_{GS} \arrow[l, bend right, "\delta^*"'] \end{tikzcd}$$

L F

"pullback" along dynamics

"predicate lifting"

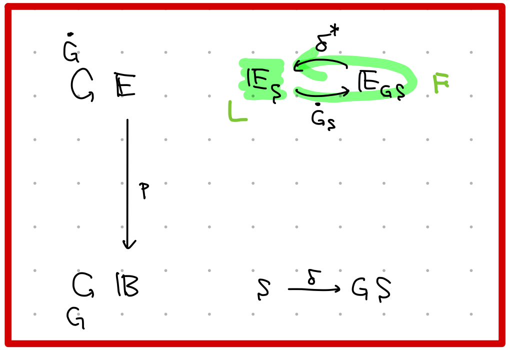
$$S \xrightarrow{\delta} GS$$

dyn. sys. as a coalgebra

A categorical frmwk

- Coalgebras as dyn. sys. [Rutten, Jacobs, ...]
- Fibrations for predicates [Benabou, Jacobs, ...]

# Unifying View of PDR Variants



## Bwd PDR

$$L \overset{?}{\dot\leq} \nu x.\ \alpha \wedge \delta^* \dot G x$$

if $\mathbb{E}_S^p \xrightarrow{\exists\,\neg} \mathbb{E}_S$ involution

if $\mathbb{E}_S \underset{T}{\xrightarrow{\dot G_S}} \mathbb{E}_{GS}$, $\exists\, \dot H_S$

$$\mu x.\ (\neg\alpha) \vee (\neg \circ \delta^* \dot G_S \circ \neg x) \overset{?}{\dot\leq} \neg L$$

## Inv.-Bwd PDR

- Reverse PDR [Seufert, Scholl, ...]
- PrIC3 [Batz, Junges, ...]

$$\mu x.\ L \vee \dot H_S \delta_* x \overset{?}{\dot\leq} \alpha$$

## Fwd PDR

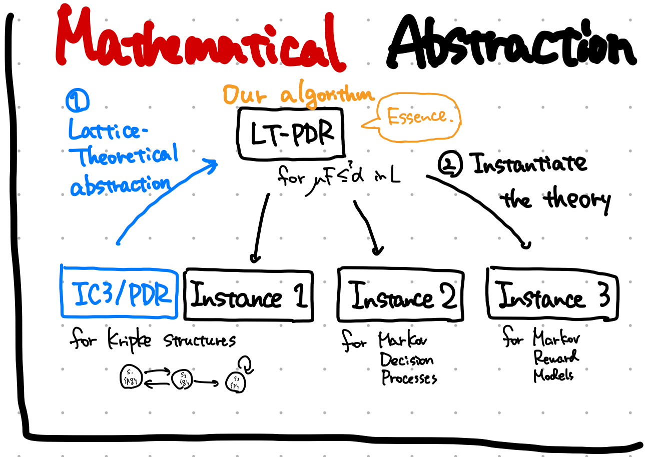
- IC3/PDR [Bradley, Een, ...]
- Unlikely for MDPs

- fbPDR [Seufert, Scholl, ...]

# Outline

1. LT-PDR — lattice-theoretic essences
   - Positive & Negative
   - LT-PDR — let these help each other
2. Instances of LT-PDR for quantitative model checking
3. Categorical classification of PDR variants
4. From mathematical abstraction to programming abstraction.
5. Experiments

# Programming Abstraction




Mathematical abstraction yields it naturally.

We can get instances of LT-PDR by small efforts.

type classes allows generic code

**Generic Haskell code of LT-PDR** ≒ pseudo code of LT-PDR

~50 lines

Lattice-Theoretical abstraction

Instantiate the theory

~900 lines (C++)

~130 lines

~80 lines

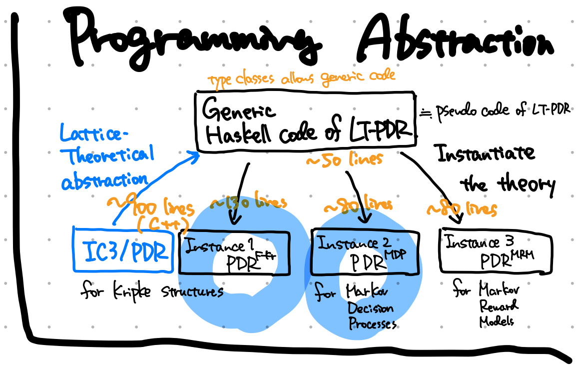
~80 lines

| IC3/PDR | Instance 1 $PDR^{F \leftarrow r}$ | Instance 2 $PDR^{MDP}$ | Instance 3 $PDR^{MRM}$ |
|---|---|---|---|
| for Kripke structures | | for Markov Decision Processes | for Markov Reward Models |

# Outline

1. LT-PDR — lattice-theoretic essences
   - Positive & Negative
   - LT-PDR — let these help each other
2. Instances of LT-PDR for quantitative model checking
3. Categorical classification of PDR variants
4. From mathematical abstraction to programming abstraction.
5. Experiments

# Experiments



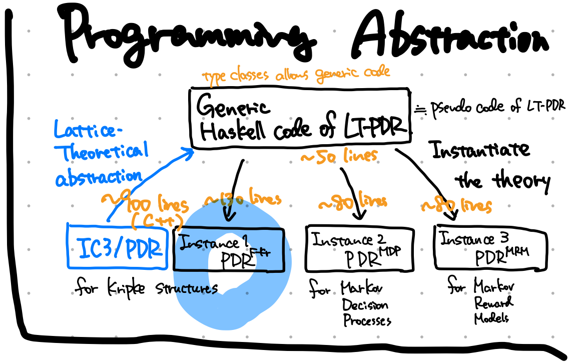

| target system | ours | to compare w/ | machine |
|---|---|---|---|
| Kripke structure | $PDR^{F\text{-}Kr}$ | IC3ref [Bradley, VMCAI'11] | Apple M1 chip with 16GB memory |
| MDPs | $PDR^{MDP}$ | PrIC3 [Batz+, CAV'20] | 1.2GHz Quad-Core Intel Core i7 with 10GB memory using Docker. |

Observations:

Performance is not too bad,
especially given that the coding effort is minimal

# Results    Comparison to IC3ref

(https://github.com/arbrad/IC3ref)




| Benchmark | $\lvert S \rvert$ | Result | $\mathbf{PDR}^{\mathbf{F\text{-}Kr}}$ | IC3ref |
|---|---|---|---|---|
| latch0.smv | $2^3$ | True | 317 µs | 270 µs |
| counter.smv | $2^5$ | False | 1.620 s | 3.27 ms |
| power2bit8.smv | $2^{15}$ | True | 1.516 s | 4.13 ms |
| ndista128.smv | $2^{17}$ | True | TO | 73.1 ms |
| shift1add256.smv | $2^{21}$ | True | TO | 174 ms |

Benchmarks:

HWMCC'15 competition

+ our own (latch0, counter)

TO: 600 sec.

IC3ref uses Z3.

$\mathrm{PDR}^{\mathrm{F\text{-}Kr}}$ uses toysolver

(https://github.com/msakai/toysolver)

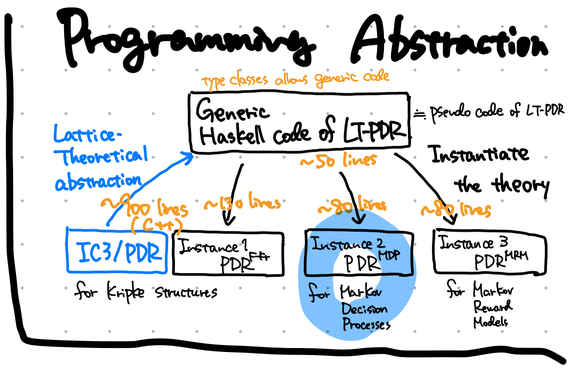
for Haskell compatibility

IC3ref outperforms our $\mathrm{PDR}^{\mathrm{F\text{-}Kr}}$.

Potential improvement: Use generalization techniques used in IC3ref.

Use another SAT-solver.

# Results   Comparison to PrIC3 [Batz+, CAV'20]




| Benchmark | $\lvert S \rvert$ | GT pr. | $\lambda$ | **PDR^IB-MDP** | PrIC3 | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | none | lin. | pol. | hyb. |
| Grid | $10^2$ | $1.2E^{-3}$ | 0.3 | 0.31 | 1.31 | 19.34 | – | – |
| | | | 0.2 | 0.48 | 1.75 | 24.62 | – | – |
| Grid | $10^3$ | $4.4E^{-10}$ | 0.3 | 122.29 | – | – | – | – |
| | | | 0.2 | 136.46 | – | – | – | – |
| BRP | $10^3$ | 0.035 | 0.1 | – | – | – | – | – |
| | | | 0.01 | 18.52 | 56.55 | 594.89 | – | 722.38 |
| | | | 0.005 | 1.36 | 11.68 | 238.09 | – | – |
| ZeroConf | $10^4$ | 0.5 | 0.9 | – | – | – | 0.58 | 0.51 |
| | | | 0.75 | – | – | – | 0.55 | 0.46 |
| | | | 0.52 | – | – | – | 0.48 | 0.46 |
| | | | 0.45 | <0.1 | <0.1 | <0.1 | <0.1 | <0.1 |
| Chain | $10^3$ | 0.394 | 0.9 | – | 72.37 | – | 0.91 | 0.70 |
| | | | 0.4 | – | 80.83 | – | 0.93 | – |
| | | | 0.35 | 177.12 | 115.98 | – | – | – |
| | | | 0.3 | 88.27 | 66.89 | 557.68 | – | – |
| DoubleChain | $10^3$ | 0.215 | 0.9 | – | – | – | 1.83 | 1.99 |
| | | | 0.3 | – | – | – | 1.88 | 1.96 |
| | | | 0.216 | – | – | – | 139.76 | – |
| | | | 0.15 | 7.46 | – | – | – | – |

Benchmarks are from [Batz+, CAV'20].

TO: 600 sec.

PrIC3 uses Z3

$\mathrm{PDR}^{\mathrm{MDP}}$ uses GLPK

No clear comparison. Ours are often better with smaller benchmarks.

Potential improvement:

- express MDPs symbolically.
- use other (so-called) generalization (linear ⇒ polynomial / hybrid)

# Conclusions

- LT-PDR : lattice-theoretic abstraction of PDR.
  - Essence of PDR : ingenious combination of KT & Kleene theorem.
  - Gives many instances for qualitative & quantitative systems.

- Generic implementation of LT-PDR in Haskell.
  - Easily-obtained instances have at least reasonable performance.

# Future work

- Relation to Abstract Interpretation.

- Get instances for other systems (e.g. hybrid systems [Suenaga I. VMCAI'20])