Exploiting Adjoints in Property Directed Reachability Analysis.

Mayuko Kori[1,2], Flavio Ascari[3], Filippo Bonchi[3], Roberto Bruni[3], Roberta Gori[3], Ichiro Hasuo[1,2]

1. National Institute of Informatics
2. The Graduate University for Advanced Studies (SOKENDAI)
3. Università di Pisa

in CAV'23

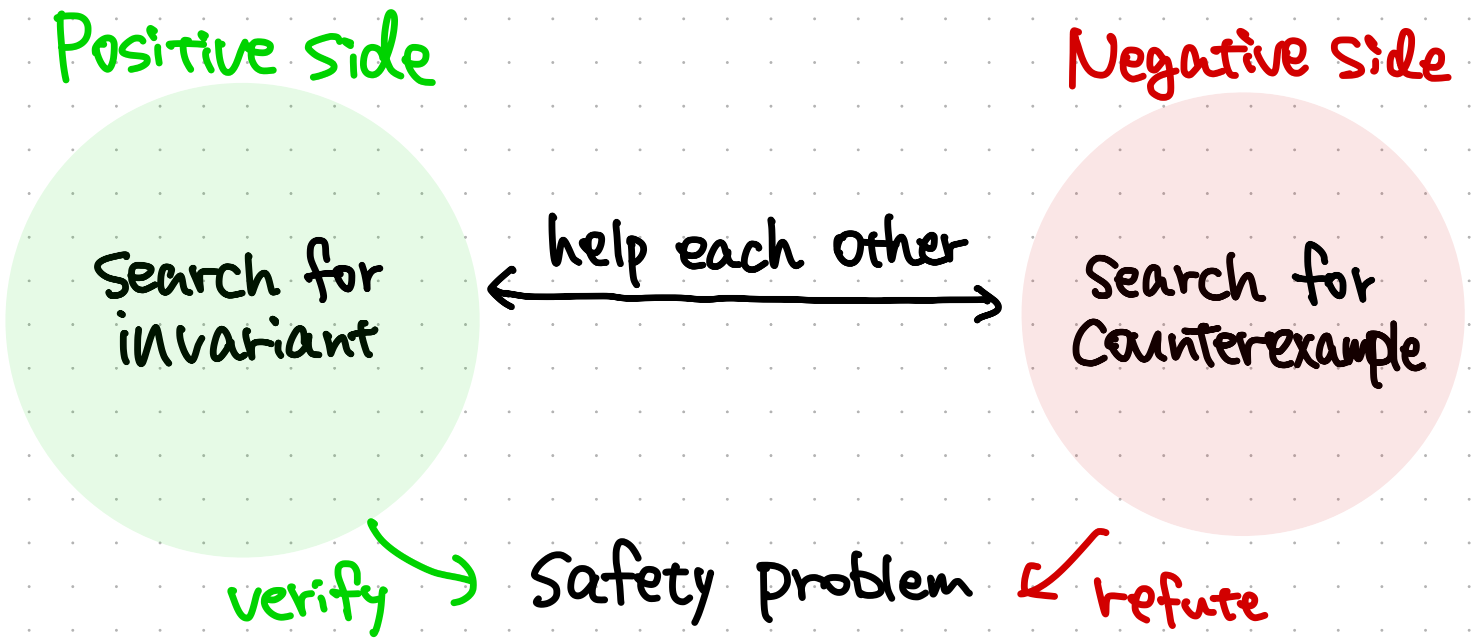
Property Directed Reachability Analysis (PDR)

Model checking Algorithm for Safety problems of state transition systems.

- original: IC3/PDR [Bradley, VMCAI'11], [Een+, FMCAD'11]
- active researches:

  GPDR [Hoder&Bjørner, SAT'12], fbPDR [Seufert & Scholl, DATE'18,'19], Λ-PDR [Feldman+, POPL'22]
  HGPDR [Suenaga & Ishizawa, VMCAI'20], PrIC3 [Batz+, CAV'20],

- lattice-theoretic generalization: LT-PDR [Kori+, CAV'22]

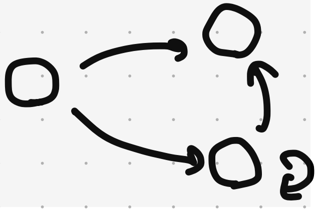


Problem Setting: General Safety Problem

$\mu g \overset{?}{\le} p$ in L. where L: Complete lattice, $\overset{\text{monotone}}{g : L \to L}$, $p \in L$

e.g.1 safety problem for Kripke frame $(i \subseteq S,\ \delta : S \to \mathcal{P}S)$

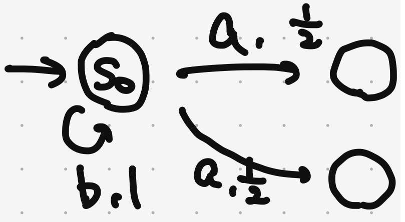
$$\underbrace{\mu\left(\bigcup \delta(-) \cup \overset{\text{initial}}{i}\right)}_{\text{reachable states}} \overset{?}{\subseteq} \overset{\text{safe}}{P} \quad \text{in } \mathcal{P}S$$



e.g.2 max reachability problem for MDP $(\overset{\text{init}}{s_0} \in S,\ \delta : S \times A \to \mathcal{D}S + 1)$

… $\Pr(\text{reaching } \overset{\text{bad}}{\beta} \subseteq S) \overset{?}{\le} \lambda$ for given $\lambda \in [0,1]$

$$\mu \underbrace{g}_{\text{defined by Bellman operator}} \overset{?}{\le} p_\lambda \quad \text{in } [0,1]^S$$



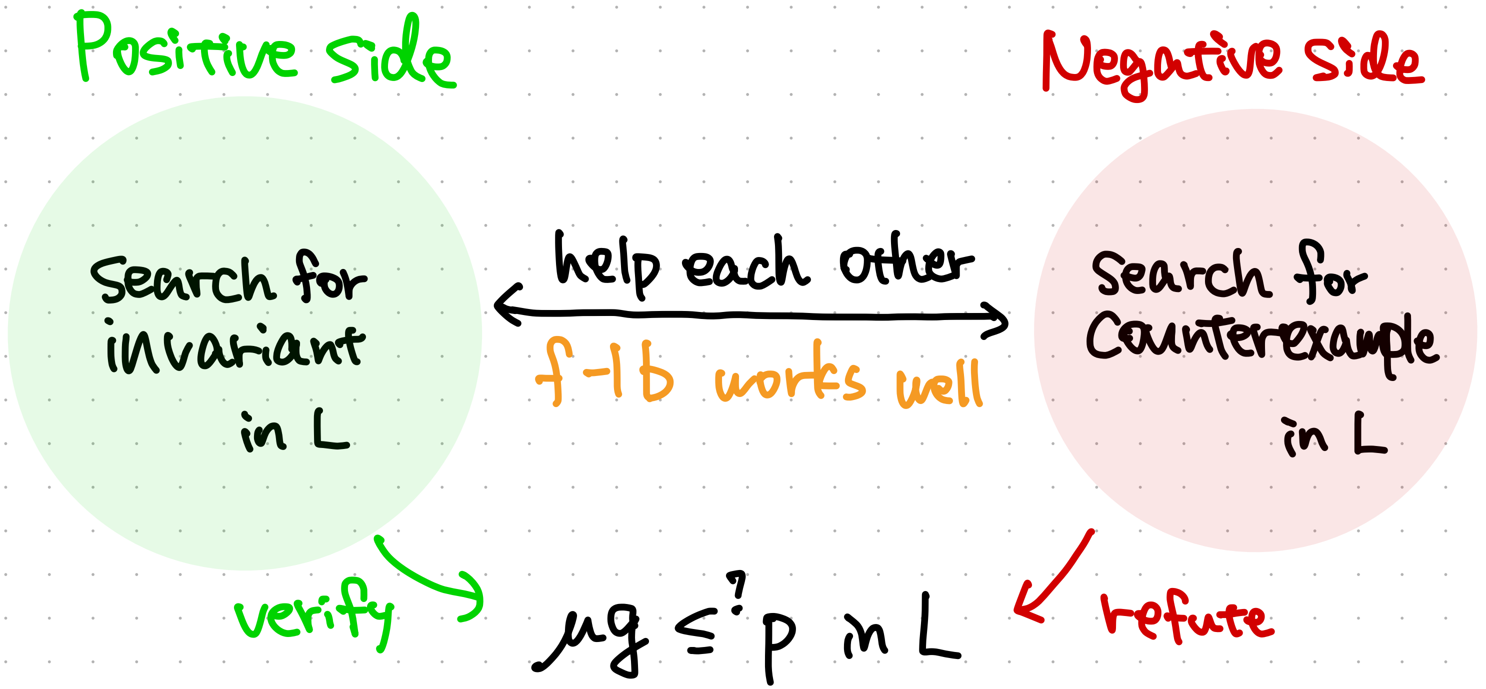
Contribution 1. AdjointPDR

generalization of PDR

We assume $\mu g \leq^? p$ satisfies $g = f \vee i$,

$$\begin{tikzcd} L \arrow[r, bend right, "b\ \text{backward}"'] \arrow[r, phantom, "\perp"] & L \arrow[l, bend right, "f\ \text{forward}"'] \end{tikzcd}$$

... galois connection



Positive side

Search for invariant in L

Negative side

Search for Counterexample in L

help each other

f-$|b$ works well

verify

refute

$\mu g \leq^? p$ in L

✓ Safety problem for Kripke frame

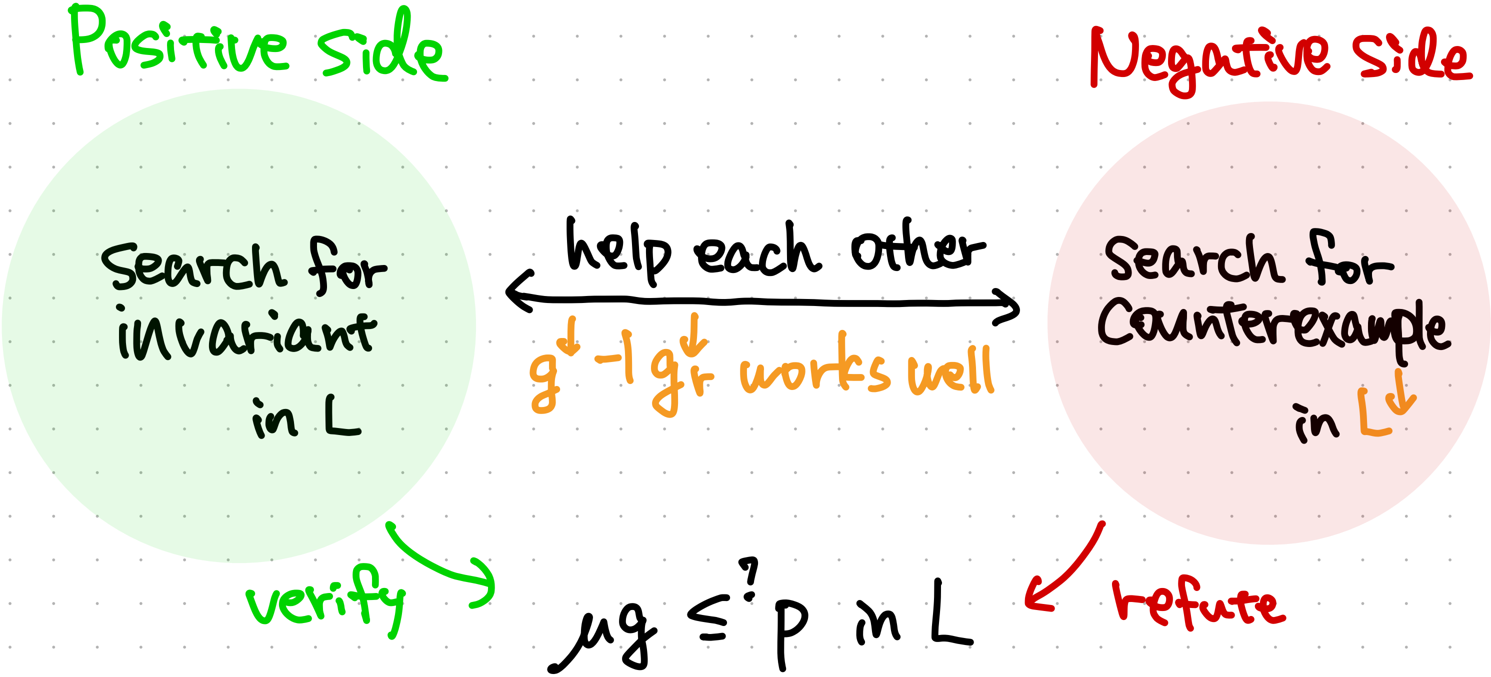
✗ max reachability problem for MDP

Contribution 2. AdjointPDR$^{\downarrow}$

for $\mu g \leq^{?} p$ without $f \dashv b$.

We recover adjoints with lower sets $L^{\downarrow}$.

$$\begin{tikzcd} g \arrow[r, phantom, "\circlearrowright"] \arrow[rrr, bend left, "\text{lift}"] & L \arrow[r, bend right, "(-)^{\downarrow}"'] \arrow[r, phantom, "\perp"] & L^{\downarrow} \arrow[l, bend right, "\vee"'] \arrow[r, phantom, "\circlearrowleft"] & g^{\downarrow} \dashv g^{\downarrow}_{r} \end{tikzcd}$$



Positive side

Search for invariant in L

help each other

$g^{\downarrow} \dashv g^{\downarrow}_{r}$ works well

Negative side

Search for Counterexample in $L^{\downarrow}$

verify

$\mu g \leq^{?} p$ in L

refute

✓ max reachability problem for MDP

$\Longrightarrow$ Mathematically simple PDR by adjoints.
Abstract theory helps devising heuristics

Outline

1. AdjointPDR – generalization of PDR

2. AdjointPDR$^\downarrow$ – extension of AdjointPDR

3. Experiments

Target Problem of Adjoint PDR

$\mu g \overset{?}{\leq} p$ in L with $g = \underbrace{f}_{\text{left adjoint.}} \vee i$ $\qquad$ ($f, g : L \to L$, $i, p \in L$)

and
$$\begin{tikzcd} L \arrow[r, bend right, "b\ (\text{backward})"'] \arrow[r, phantom, "\perp"] & L \arrow[l, bend right, "f\ (\text{forward})"'] \end{tikzcd}$$
… forward/backward adjoint

e.g. safety problem for a Kripke frame $(S, i \subseteq S, \delta : S \to \mathcal{P}S)$.

$\mu(\underbrace{\cup\delta(-)}_{\text{left adjoint}} \cup i) \overset{?}{\leq} p$ in $\mathcal{P}S$ with
$$\begin{tikzcd} \mathcal{P}S \arrow[r, bend right, "\{s \mid \delta s \subseteq (-)\}\ \text{backward}"'] \arrow[r, phantom, "\perp"] & \mathcal{P}S \arrow[l, bend right, "\cup\delta(-)\ \text{forward}"'] \end{tikzcd}$$

Forward / Backward form of $\mu g \overset{?}{\leq} p$ with

$g = f \vee i,$

$$\begin{tikzcd} L \arrow[r, phantom, "\perp"] \arrow[r, bend right, "b\ (\text{backward})"'] & L \arrow[l, bend right, "f\ (\text{forward})"'] \end{tikzcd}$$

Target prob.

$\mu g \leq p$ in L $\iff \mu(f \vee i) \leq p$ forward

$\overset{KT}{\iff} \exists x. \begin{cases} f x \vee i \leq x \\ x \leq p \end{cases}$

(KT: Knaster-Tarski thm)

$\iff \exists x. \begin{cases} f x \leq x \\ i \leq x \leq p \end{cases}$

$\overset{f \dashv b}{\iff} \exists x. \begin{cases} x \leq b x \\ i \leq x \leq p \end{cases}$

$\iff \exists x. \begin{cases} x \leq b x \wedge p \\ i \leq x \end{cases}$

$\overset{KT}{\iff} i \leq \nu(b \wedge p)$ backward

How to solve?

$$\begin{tikzcd} L \arrow[r, bend right, "b\ (\text{backward})"'] \arrow[r, phantom, "\perp"] & L \arrow[l, bend right, "f\ (\text{forward})"'] \end{tikzcd}$$

$\mu g \leq^? p$

1. forward form:

$$\mu(f \vee i) \leq^? p$$

By Kleene thm,

initial chain

$\bot \leq i \leq i \vee f i \leq \cdots$ $\quad \mu(f \vee i) \leq^? p$

$\xrightarrow{\hspace{4em}}$

Converge to $\mu(f \vee i)$

2. backward form:

$$i \leq^? \nu(b \wedge p)$$

By Kleene thm,

final chain

$i \leq^? \nu(b \wedge p)$ $\quad \cdots \leq bp \wedge p \leq p \leq \top$

$\xleftarrow{\hspace{4em}}$

converge to $\nu(b \wedge p)$

in AdjointPDR.

negative seq.

positive chain

$$\begin{array}{ccccccccc}
 & & & & y_1, \ \cdots & & ,\ y_{n-1} & & \\
 & & & & \rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} & & \\
 & & \bigwedge b^i p & \leq & \cdots & \leq & p & \leq & \top \\
 & & \rotatebox{90}{$\leq$} & & & & \rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} \\
\bot & \leq & x_1 & \leq & \cdots & \leq & x_{n-2} & \leq & x_{n-1}. \\
\rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} & & & & \rotatebox{90}{$\leq$} & & \\
\bot & \leq & i & \leq & \cdots & \leq & \bigvee f^j i & &
\end{array}$$

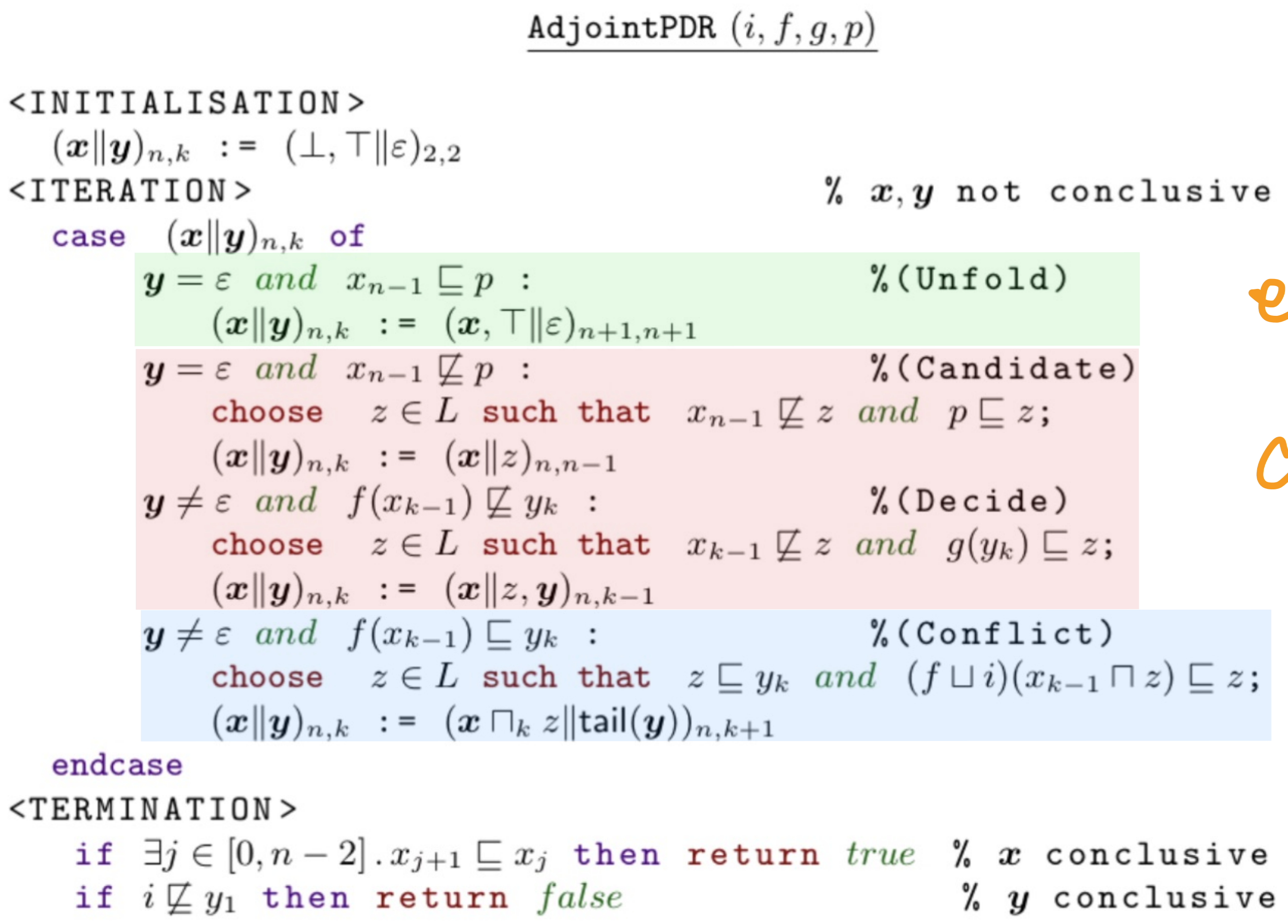
over-approx.

under-approx.

over-approx.

approximation accelerates the algorithm.

AdjointPDR solves $\mu g \leq^{?} p$ in L with $g = f \vee i$, $f \dashv b$.



```
AdjointPDR $(i, f, g, p)$

<INITIALISATION>
  $(\boldsymbol{x} \| \boldsymbol{y})_{n,k}$ := $(\bot, \top \| \varepsilon)_{2,2}$
<ITERATION>                                         % $\boldsymbol{x}, \boldsymbol{y}$ not conclusive
  case $(\boldsymbol{x} \| \boldsymbol{y})_{n,k}$ of
      $\boldsymbol{y} = \varepsilon$ and $x_{n-1} \sqsubseteq p$ :                %(Unfold)
          $(\boldsymbol{x} \| \boldsymbol{y})_{n,k}$ := $(\boldsymbol{x}, \top \| \varepsilon)_{n+1,n+1}$
      $\boldsymbol{y} = \varepsilon$ and $x_{n-1} \not\sqsubseteq p$ :                %(Candidate)
          choose $z \in L$ such that $x_{n-1} \not\sqsubseteq z$ and $p \sqsubseteq z$;
          $(\boldsymbol{x} \| \boldsymbol{y})_{n,k}$ := $(\boldsymbol{x} \| z)_{n,n-1}$
      $\boldsymbol{y} \neq \varepsilon$ and $f(x_{k-1}) \not\sqsubseteq y_k$ :            %(Decide)
          choose $z \in L$ such that $x_{k-1} \not\sqsubseteq z$ and $g(y_k) \sqsubseteq z$;
          $(\boldsymbol{x} \| \boldsymbol{y})_{n,k}$ := $(\boldsymbol{x} \| z, \boldsymbol{y})_{n,k-1}$
      $\boldsymbol{y} \neq \varepsilon$ and $f(x_{k-1}) \sqsubseteq y_k$ :            %(Conflict)
          choose $z \in L$ such that $z \sqsubseteq y_k$ and $(f \sqcup i)(x_{k-1} \sqcap z) \sqsubseteq z$;
          $(\boldsymbol{x} \| \boldsymbol{y})_{n,k}$ := $(\boldsymbol{x} \sqcap_k z \| \mathsf{tail}(\boldsymbol{y}))_{n,k+1}$
  endcase
<TERMINATION>
    if $\exists j \in [0, n-2] \,.\, x_{j+1} \sqsubseteq x_j$ then return true  % $\boldsymbol{x}$ conclusive
    if $i \not\sqsubseteq y_1$ then return false                    % $\boldsymbol{y}$ conclusive
```

positive chain
$x_0 \leq x_1 \leq \cdots \leq x_{n-1}$
negative seq
$y_k, \cdots, y_{n-1}$

extend positive chain

Construct negative seq.

refine approximation
(shrink overly-inflated positive chain)

users need to specify heuristics.

1. how to construct negative seq. y.
2. how to shrink overly-inflated positive chain. x.

Property of AdjointPDR

Thm. **Soundness**

If AdjointPDR returns true/false then $\mu(f \vee i) \leq p$ / $\mu(f \vee i) \not\leq p$.

Thm. **Progression**

In any run, there's no loop.

Thm. **Negative Termination**

If $\mu(f \vee i) \not\leq p$ and choices of $y = (y_k, y_{k+1}, \cdots, y_{n-1})$ is finite,

AdjointPDR terminates.

This holds
when L is finite
or whenever we use canonical choice

$$y = (b^{n-k}p, \cdots, bp, p)$$

$\longleftarrow$ final chain

Outline

1. AdjointPDR – generalization of PDR

2. AdjointPDR$^{\downarrow}$ – extension of AdjointPDR

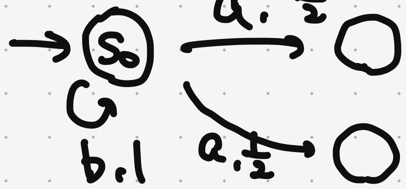
3. Experiments

$\mu g \overset{?}{\leq} p$ without $f \dashv b$

e.g. max reachability problem for MDP $(A, S, \overset{\text{init}}{s_0 \in S}, \delta : S \times A \to \mathcal{D}S+1)$

Bellman operator $f(d : S \to [0,1]) = s \mapsto \max_{a} \sum_{s'} d s' \cdot \delta(s,a,s')$

$i = s \mapsto \begin{cases} 1 & \text{if } s \in \overset{\text{bad}}{\beta} \\ 0 & \text{otherwise} \end{cases}$



then $\mu(f \vee i) = s \mapsto \Pr(\text{reaching } \overset{\text{bad}}{\beta} \subseteq S \text{ from } s)$ in $[0,1]^S$

$\Pr(\text{reaching } \beta \text{ from } s_0) \overset{?}{\leq} \lambda$ in $[0,1]$

$\Leftrightarrow \mu(f \vee i) \overset{?}{\leq} \begin{pmatrix} s_0 \mapsto \lambda \\ \_ \mapsto 1 \end{pmatrix}$ in $[0,1]^S$

f doesn't have a right adjoint ... any left adjoint preserve joins.
but $f(d_1 \vee d_2) \neq f(d_1) \vee f(d_2)$.

So this problem is out of scope of AdjointPDR.

Recovering adjoints with lower sets

$$\begin{tikzcd} (L,\le) \arrow[loop left, "g"', "\omega\text{-conti.}"] \arrow[r, bend right, "(-)^{\downarrow}"'] \arrow[r, phantom, "\perp"] & (L^{\downarrow},\subseteq) \arrow[l, bend right, "\bigvee"'] \arrow[loop right, "g^{\downarrow} \dashv g_r^{\downarrow}"] \end{tikzcd}$$

$g^{\downarrow} := \{gy \mid y \in (-)\}$

$g_r^{\downarrow} := \{y \mid gy \in (-)\}$

$\underbrace{g^{\downarrow} \dashv g_r^{\downarrow}}_{g \text{ always have}}$

$(-)^{\downarrow} := \{x \mid x \le (-)\}$

Target prob.

$$\underbrace{\mu g \le^{?} p \text{ in } L}_{\uparrow \text{ AdjointPDR may not solve}} \iff \underbrace{\mu(g^{\downarrow} \vee \bot^{\downarrow}) \le^{?} p^{\downarrow}}_{\uparrow \text{ AdjointPDR can solve.}} \text{ in } L^{\downarrow}.$$

But $L^{\downarrow}$ is too large to get convergence of positive chain.

acceleration.

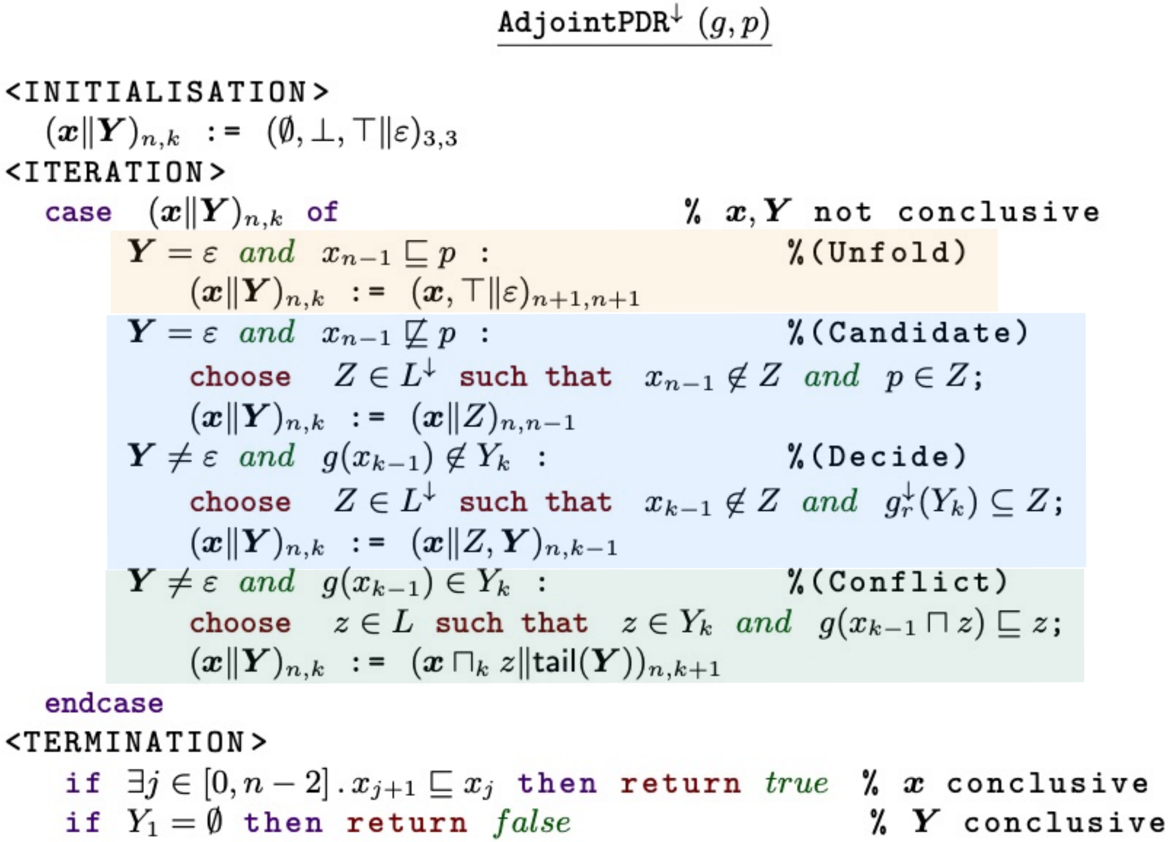
So AdjointPDR$^{\downarrow}$ uses positive chain x in L

negative seq. Y in $L^{\downarrow}$.

a set of negative seq. y in L of AdjointPDR.

AdjointPDR$^\downarrow$

solves $\mu g \overset{?}{\leq} p$ in L. almost the same as AdjointPDR except for negative seq.



$\texttt{AdjointPDR}^\downarrow\,(g,p)$

```
<INITIALISATION>
  $(\boldsymbol{x}\|\boldsymbol{Y})_{n,k}$ := $(\emptyset,\bot,\top\|\varepsilon)_{3,3}$
<ITERATION>
  case $(\boldsymbol{x}\|\boldsymbol{Y})_{n,k}$ of                % $\boldsymbol{x},\boldsymbol{Y}$ not conclusive
      $\boldsymbol{Y}=\varepsilon$ and $x_{n-1}\sqsubseteq p$ :                %(Unfold)
          $(\boldsymbol{x}\|\boldsymbol{Y})_{n,k}$ := $(\boldsymbol{x},\top\|\varepsilon)_{n+1,n+1}$
      $\boldsymbol{Y}=\varepsilon$ and $x_{n-1}\not\sqsubseteq p$ :                %(Candidate)
          choose $Z\in L^\downarrow$ such that $x_{n-1}\notin Z$ and $p\in Z$;
          $(\boldsymbol{x}\|\boldsymbol{Y})_{n,k}$ := $(\boldsymbol{x}\|Z)_{n,n-1}$
      $\boldsymbol{Y}\neq\varepsilon$ and $g(x_{k-1})\notin Y_k$ :                %(Decide)
          choose $Z\in L^\downarrow$ such that $x_{k-1}\notin Z$ and $g_r^\downarrow(Y_k)\subseteq Z$;
          $(\boldsymbol{x}\|\boldsymbol{Y})_{n,k}$ := $(\boldsymbol{x}\|Z,\boldsymbol{Y})_{n,k-1}$
      $\boldsymbol{Y}\neq\varepsilon$ and $g(x_{k-1})\in Y_k$ :                %(Conflict)
          choose $z\in L$ such that $z\in Y_k$ and $g(x_{k-1}\sqcap z)\sqsubseteq z$;
          $(\boldsymbol{x}\|\boldsymbol{Y})_{n,k}$ := $(\boldsymbol{x}\sqcap_k z\|\mathsf{tail}(\boldsymbol{Y}))_{n,k+1}$
  endcase
<TERMINATION>
   if $\exists j\in[0,n-2]\,.\,x_{j+1}\sqsubseteq x_j$ then return $true$    % $\boldsymbol{x}$ conclusive
   if $Y_1=\emptyset$ then return $false$                      % $\boldsymbol{Y}$ conclusive
```

positive chain
$x_0 \leq x_1 \leq \cdots \leq x_{n-1}$ in L

negative seq.
$Y_k, \cdots, Y_{n-1}$ in $L^\downarrow$

users need to specify heuristics.

1. how to construct negative seq. Y.
2. how to shrink overly-inflated positive chain. x.

Property of Adjoint PDR$^\downarrow$

Thm. **Soundness**

If AdjointPDR$^\downarrow$ returns true/false then $\mu g \leq p$ / $\mu g \nleq p$.

Thm. **Progression**

In any run, there's no loop.

Thm. **Negative Termination**

If $\mu g \nleq p$ and choices of $Y = (Y_k, Y_{k+1}, \cdots, Y_{n-1})$ is finite,

AdjointPDR$^\downarrow$ terminates.

This holds

whenever we use canonical choice

$$Y = (g_r^{\downarrow n-k} p^\downarrow, \cdots, g_r^{\downarrow 2} p^\downarrow, g_r^\downarrow p^\downarrow, p^\downarrow)$$

$\longleftarrow$

final chain

Adjoint PDR$^{\downarrow}$ for MDPs

max reachability problem for MDP ($s_0 \in S$, $\delta : S \times A \to \mathcal{D}S+1$)

... Pr(reaching some bad states $\beta \subseteq S$) $\overset{?}{\leq} \lambda$ for given $\lambda \in [0,1]$

⇔ LFP problem w.r.t. Bellman operator $s \mapsto \max_{a \in A} \sum_{s'} ds' \cdot \delta(s,a,s')$

Canonical heuristics based on final chain.

$Y_{n-1} = \{ d \in [0,1]^S \mid d(s_0) \leq \lambda \}$, $Y_{n-2} = \{ d \mid \max_{a \in A} \sum_{s'} ds' \cdot \delta(s_0, a, s') \leq \lambda \}$, $Y_{n-3}, Y_{n-4}, \cdots$

natullarly get

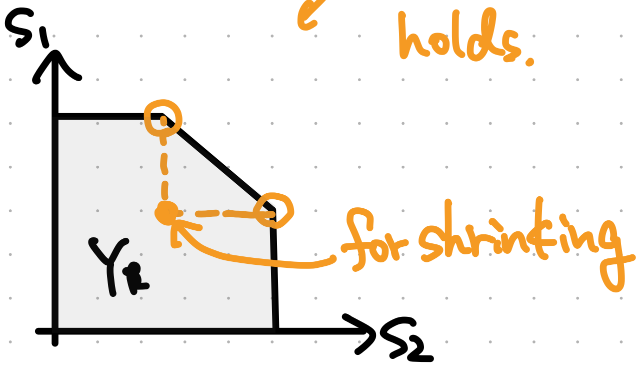
negative termination holds.

Our heuristics:

By choosing a scheduler,

Y_k can be expressed by a linear inequality.

shrink x by taking meet of generators of Y_k.



I'll show it gives practical performance in experiments.

Outline

1. AdjointPDR – generalization of PDR

2. AdjointPDR$^{\downarrow}$ – extension of AdjointPDR

3. Experiments

Experiment

We implemented a generic template for AdjointPDR$^{\downarrow}$ in Haskell.

→ By specifying heuristics, users get an instance.
(e.g. instance for Kripke frame, MDP, …)

We compared an instance of AdjointPDR$^{\downarrow}$ for MDPs

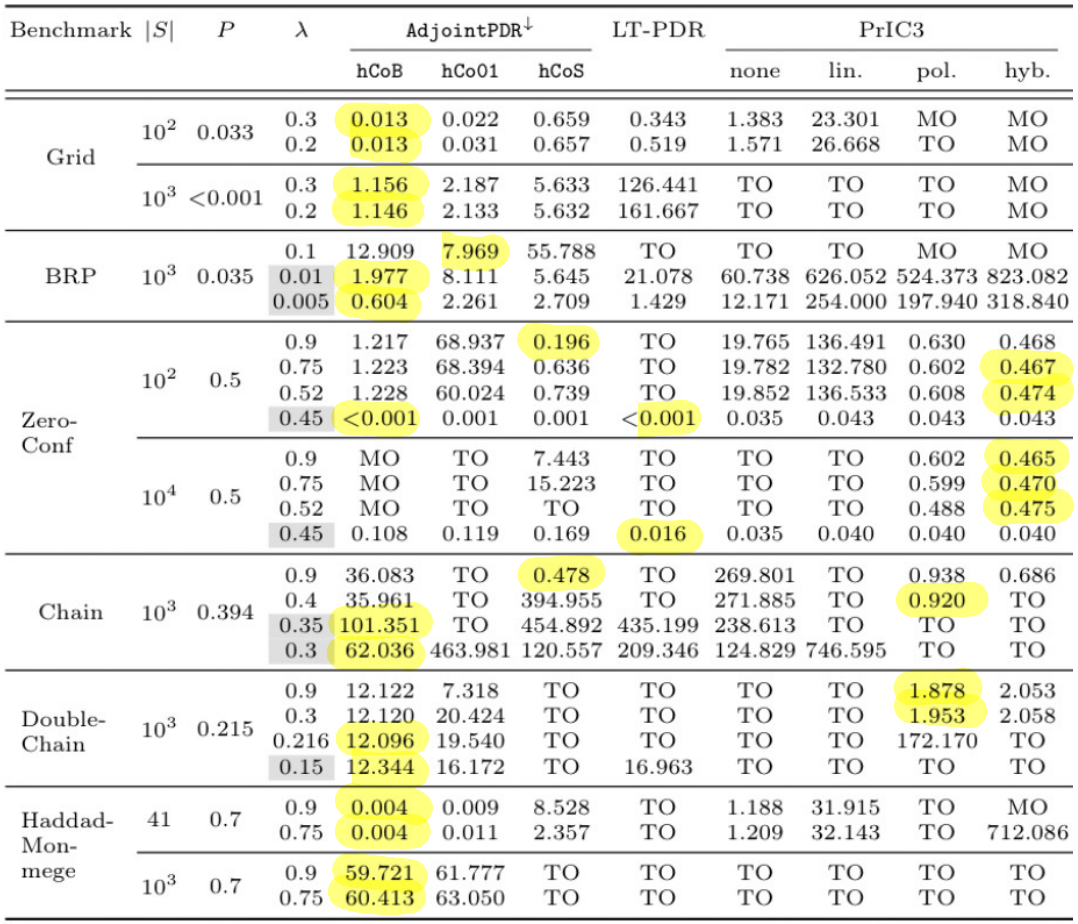
to LT-PDR [Kori+, CAV'22]
PrIC3 [Batz+, CAV'20]
) PDR algorithms for MDPs.

Storm [Dehnert+, CAV'17]) non-PDR algorithm for MDPs.

machine: Ubuntu 18.04, 4 CPUs, 16 GB memory, up to 3.0 GHz
Intel Scalable Processor.

Results Comparison to LT-PDR, PrIC3 (PDR algorithms)

[Kori+.CAV22] [Batz+.CAV'20]



Benchmark	$\lvert S\rvert$	P	λ	$\texttt{AdjointPDR}^{\downarrow}$			LT-PDR	PrIC3			
				hCoB	hCo01	hCoS		none	lin.	pol.	hyb.
Grid	10^2	0.033	0.3	0.013	0.022	0.659	0.343	1.383	23.301	MO	MO
			0.2	0.013	0.031	0.657	0.519	1.571	26.668	TO	MO
	10^3	<0.001	0.3	1.156	2.187	5.633	126.441	TO	TO	TO	MO
			0.2	1.146	2.133	5.632	161.667	TO	TO	TO	MO
BRP	10^3	0.035	0.1	12.909	7.969	55.788	TO	TO	TO	MO	MO
			0.01	1.977	8.111	5.645	21.078	60.738	626.052	524.373	823.082
			0.005	0.604	2.261	2.709	1.429	12.171	254.000	197.940	318.840
Zero-Conf	10^2	0.5	0.9	1.217	68.937	0.196	TO	19.765	136.491	0.630	0.468
			0.75	1.223	68.394	0.636	TO	19.782	132.780	0.602	0.467
			0.52	1.228	60.024	0.739	TO	19.852	136.533	0.608	0.474
			0.45	<0.001	0.001	0.001	<0.001	0.035	0.043	0.043	0.043
	10^4	0.5	0.9	MO	TO	7.443	TO	TO	TO	0.602	0.465
			0.75	MO	TO	15.223	TO	TO	TO	0.599	0.470
			0.52	MO	TO	TO	TO	TO	TO	0.488	0.475
			0.45	0.108	0.119	0.169	0.016	0.035	0.040	0.040	0.040
Chain	10^3	0.394	0.9	36.083	TO	0.478	TO	269.801	TO	0.938	0.686
			0.4	35.961	TO	394.955	TO	271.885	TO	0.920	TO
			0.35	101.351	TO	454.892	435.199	238.613	TO	TO	TO
			0.3	62.036	463.981	120.557	209.346	124.829	746.595	TO	TO
Double-Chain	10^3	0.215	0.9	12.122	7.318	TO	TO	TO	TO	1.878	2.053
			0.3	12.120	20.424	TO	TO	TO	TO	1.953	2.058
			0.216	12.096	19.540	TO	TO	TO	TO	172.170	TO
			0.15	12.344	16.172	TO	16.963	TO	TO	TO	TO
Haddad-Monmege	41	0.7	0.9	0.004	0.009	8.528	TO	1.188	31.915	TO	MO
			0.75	0.004	0.011	2.357	TO	1.209	32.143	TO	712.086
	10^3	0.7	0.9	59.721	61.777	TO	TO	TO	TO	TO	TO
			0.75	60.413	63.050	TO	TO	TO	TO	TO	TO

$P = Pr(\text{reaching bad states}) \overset{?}{\leq} \lambda$

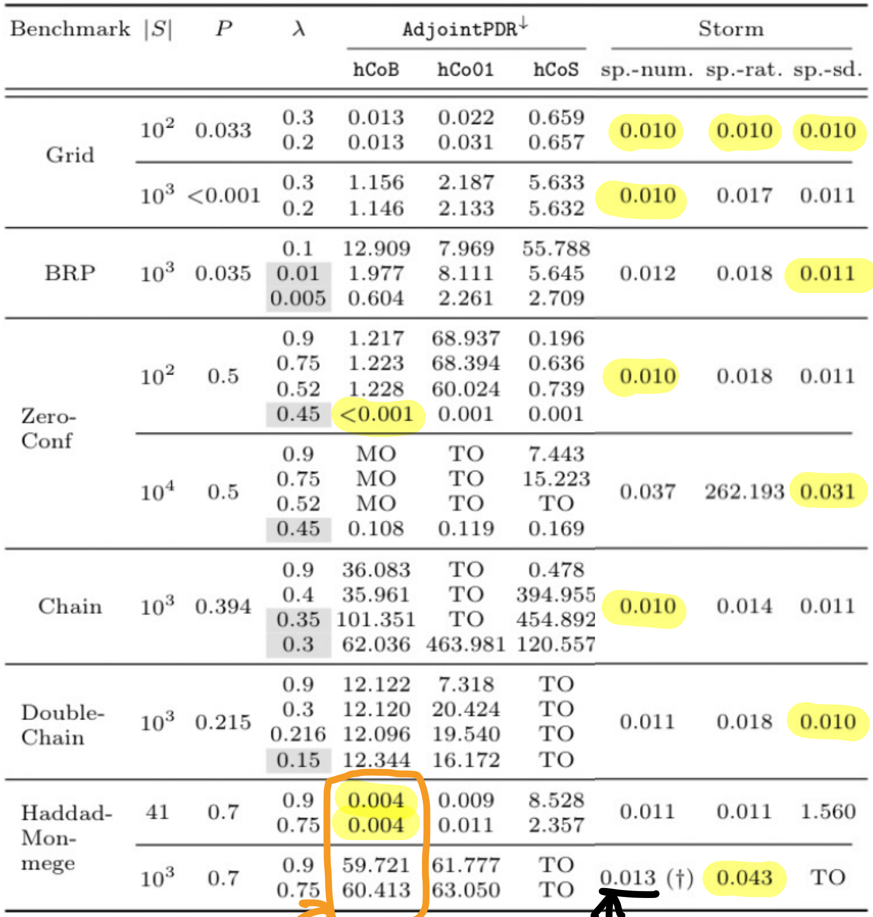
$\text{AdjointPDR}^{\downarrow}$ outperformed LT-PDR.

$\text{AdjointPDR}^{\downarrow}$ outperformed PrIC3 except when polynomial and hybrid in PrIC3 perform well.

potential improvement: use polynomial or hybrid template.

Results Comparison to Storm (non-PDR algorithm)

[Dehnert+, CAV'17]



Benchmark	$\|S\|$	P	λ	AdjointPDR$^{\downarrow}$			Storm		
				hCoB	hCo01	hCoS	sp.-num.	sp.-rat.	sp.-sd.
Grid	10^2	0.033	0.3	0.013	0.022	0.659	0.010	0.010	0.010
			0.2	0.013	0.031	0.657			
	10^3	<0.001	0.3	1.156	2.187	5.633	0.010	0.017	0.011
			0.2	1.146	2.133	5.632			
BRP	10^3	0.035	0.1	12.909	7.969	55.788	0.012	0.018	0.011
			0.01	1.977	8.111	5.645			
			0.005	0.604	2.261	2.709			
Zero-Conf	10^2	0.5	0.9	1.217	68.937	0.196	0.010	0.018	0.011
			0.75	1.223	68.394	0.636			
			0.52	1.228	60.024	0.739			
			0.45	<0.001	0.001	0.001			
	10^4	0.5	0.9	MO	TO	7.443	0.037	262.193	0.031
			0.75	MO	TO	15.223			
			0.52	MO	TO	TO			
			0.45	0.108	0.119	0.169			
Chain	10^3	0.394	0.9	36.083	TO	0.478	0.010	0.014	0.011
			0.4	35.961	TO	394.955			
			0.35	101.351	TO	454.892			
			0.3	62.036	463.981	120.557			
Double-Chain	10^3	0.215	0.9	12.122	7.318	TO	0.011	0.018	0.010
			0.3	12.120	20.424	TO			
			0.216	12.096	19.540	TO			
			0.15	12.344	16.172	TO			
Haddad-Monmege	41	0.7	0.9	0.004	0.009	8.528	0.011	0.011	1.560
			0.75	0.004	0.011	2.357			
	10^3	0.7	0.9	59.721	61.777	TO	0.013 (†)	0.043	TO
			0.75	60.413	63.050	TO			

Sparcity may improve

return wrong P (=0.5)

$P = Pr(\text{reaching bad states}) \overset{?}{\leq} \lambda$

sp.-num. : Value iteration alg.
it may return a wrong answer
while AdjointPDR$^{\downarrow}$ is precise.

sp.-rat. : exact model checking

sp.-sd. : sound model checking

Storm was faster than AdjointPDR$^{\downarrow}$ in many benchmarks,
although AdjointPDR$^{\downarrow}$ compared well with Storm in a couple of benchmarks.

potential improvement: use sparce representation.

Conclusions

Two PDR algorithms got by exploiting adjoints.

1. AdjointPDR for $\mu g \leq^? p$ with $g = f \vee i$.

$$\begin{tikzcd} L \arrow[r, phantom, "\perp"] \arrow[r, bend right, "b\ \text{(backward)}"'] & L \arrow[l, bend right, "f\ \text{(forward)}"'] \end{tikzcd}$$

2. AdjointPDR$^\downarrow$ for $\mu g \leq^? p$

- Recover $f \dashv b$ with lower sets.

$$\begin{tikzcd} g \in L \arrow[r, phantom, "\perp"] \arrow[r, bend right, "(-)^\downarrow"'] & L^\downarrow \ni g^\downarrow \dashv g_r^\downarrow \arrow[l, bend right, "\vee"'] \end{tikzcd}$$

- We successfully derived practical heuristics from canonical one. The performance for MDPs is encouraging.

$\Longrightarrow$ Mathematically simple PDR by adjoints.
Abstract theory helps devising heuristics