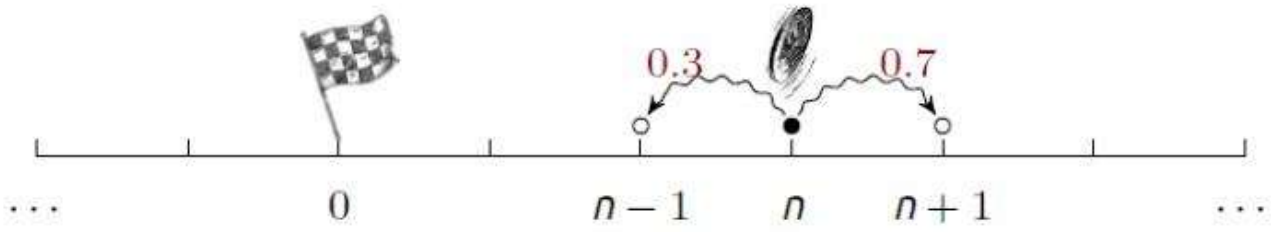
Piecewise Analysis of Probabilistic Programs via k-Induction

Tengshun Yang, Hongfei Fu, Jingyu Ke, Naijun Zhan, Shiyang Wu

Probabilistic Programs

$$C ::= \mathsf{skip} \mid x := e \mid x :\approx \mu \mid C; C \mid \{C\}\ [p]\ \{C\} \mid$$
$$\mathsf{if}\ (\varphi)\ \{C\}\ \mathsf{else}\ \{C\} \mid \mathsf{while}\ (\varphi)\ \{C\}$$
$$\varphi ::= e < e \mid \neg\varphi \mid \varphi \wedge \varphi \qquad e ::= c \mid x \mid c \cdot e \mid e + e \mid e - e$$

C_{brw}: $\mathsf{while}\,(n > 0)\,\{\,n := n - 1\ [0.3]\ n := n + 1\,\}$



Background

- Quantitative analysis of probabilistic programs
  - Concerns quantitative properties in many situations: expectation, assertion probability, expected running time, etc.
  - Derive tight numerical bounds for probabilistic properties

- Synthesis of bounds
  - Most previous works consider monolithic bounds.
  - Monolithic bounds are either conservative, or not expressive and succinct enough

Previous Works

- Synthesis of monolithic polynomials
  - (Sub)-Invariants
    (Feng et al., ATVA 2017, Chen et al., CAV 2015)
  - Probabilistic Program Analysis
    (Sankaranarayanan et al., CAV 2013)
  - Monolithic polynomial for Termination analysis
    (Chatterjee et al., CAV 2016, TOPLAS 2018)

- Synthesis of (piecewise) bounds to **verify** an input bound
  - Counter-example guided inductive synthesis (Batz et al., TACAS 2023)
  - Data-driven Learning (Bao et al., CAV 2022)

This Work

- Target
  - Synthesis of **tight** piecewise polynomial bounds for the expected value of certain quantitative properties on probabilistic while loops without a target bound to be verified (static analysis)

- Main Ideas
  - A novel combination between (latticed) k-induction and Optional Stopping Theorem.
  - Novel algorithms to synthesize bounds.

Problem Statements

Given a probabilistic while loop P and a return function f, synthesize tight piecewise upper and lower bounds on the expected output of f after the execution of P.

Our contributions

- Explore piecewise information
  - Equivalence of k-induction conditions and their applications
- Automated Algorithms
  - Propositions and efficient algorithms in constraints derivation
  - Novel algorithms for constraints solving:
    1. linear cases: reduced to bilinear programming
    2. polynomial cases: relaxed to semi-definite programming
  - Numerical Repair
- Soundness
  - Use Extended OST (Wang et al., PLDI 2024) to ensure the soundness

(Latticed) k-induction

- Given a monotone operator Φ:

1. the upper k-induction operator Ψ_u in (Batz et al., CAV2021) w.r.t. u and Φ is defined by $\Psi_u: v \mapsto \Phi(v) \sqcap u$

2. the upper k-induction operator Ψ in (Lu et al., APSEC 2022) is defined by $\Psi: v \mapsto \Phi(v) \sqcap v$.

- lower k-induction operators: replacing the meet operation $\sqcap$ with join $\sqcup$.
- 1-induction is the special case and has been used extensively in existing work.

Equivalence Theorem:

For any k, $\Psi_u^k(u) = \Psi^k(u)$ for any u.

Inductive Theorem:

$$\Phi(\Psi_u^k(u)) \leq u \Rightarrow \Phi(\Psi_u^k(u)) \leq \Psi_u^k(u)$$

Some Necessary Definitions

- Syntax: while $(\varphi)\ \{C\}$

$$C ::= \text{skip} \mid x := e \mid x :\approx \mu \mid C;C \mid \{C\}\ [p]\ \{C\} \mid \text{if } (\varphi)\ \{C\} \text{ else } \{C\}$$
$$\varphi ::= e < e \mid \neg\varphi \mid \varphi \wedge \varphi \qquad e ::= c \mid x \mid c \cdot e \mid e + e \mid e - e$$

- Characteristic Function——the monotone operator Φ

$$\Phi(h) \coloneqq [\neg\varphi] \cdot f \ + \ [\varphi] \cdot pre_C(h)$$

Informally, the characteristic function Φ outputs f if the loop guard φ is violated and the expected value of h after the execution of the loop body C otherwise.

A Simplified Growing Walk Example

```
while (x >=0) {
    if (flip(0.5)){x = -1}
    else {x += 1;
          y += x;}
};
return y
```

Questions:

Return function: y

Intend to analyze the expected value of y after the program terminates.

Monolithic polynomial via 1-induction:

Monolithic linear upper bound do not exist.

Monolithic polynomial upper bounds up to degree 5 are much more conservative.

Through our methods: (the exact result)

$[x < 0] \cdot y + [x \geq 0] \cdot (x + y + 2)$

A Simplified Growing Walk Example

```
while (x >=0) {
    if (flip(0.5)){x = -1}
    else {x += 1;
          y += x;}
};
return y
```

Our approach: (for upper bounds)

Template-based approach:

- polynomial template h
- pre-processing
- Deriving k-induction-based constraints: $\Phi(\Psi_h^{k-1}(h)) \leq h$
- polynomial solving

The latter two steps are non-trivial and need novel algorithms to address it!

Deriving k-induction Constraints

- Transfer $\Phi(\Psi_h^{k-1}(h)) \leq h$ into a simpler form $\min\{h_1, h_2, \ldots, h_n\} \preccurlyeq h$
- We propose an efficient algorithm to obtain the form:

1. Make a decision whether we continue to unfold the loop at each state we reach.

2. this decision process continues until there are no unfolding to be executed or we have already unfolded for k times.

3. Each strategy, composed of each decision, decides a distinct loop-free program.

PROPOSITION 5.3. *The upper (resp. lower) k-induction condition* $\overline{\Phi}_f(\overline{\Psi}_h^{k-1}(h)) \preceq h$ *(resp.* $\overline{\Phi}_f(\overline{\Psi}'_h)^{k-1}(h)) \succeq h$*) is equivalent with* $\min\{h_1, h_2, \ldots, h_m\} \preceq h$ *(resp.* $\max\{h_1, h_2, \ldots, h_m\} \succeq h$*), where each* h_i *uniquely corresponds to one* $C_d \in \{C_1, \ldots, C_m\}$ *and is equal to* $pre_{C_d}(h)$.

A Simple but Illustrative Example ——for k-induction

while (φ) {

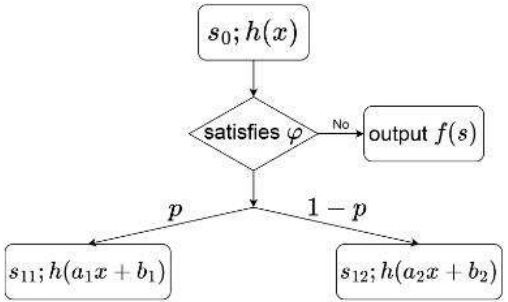
if (flip(p)){$x = a_1 x + b_1$}

else {$x = a_2 x + b_2$};

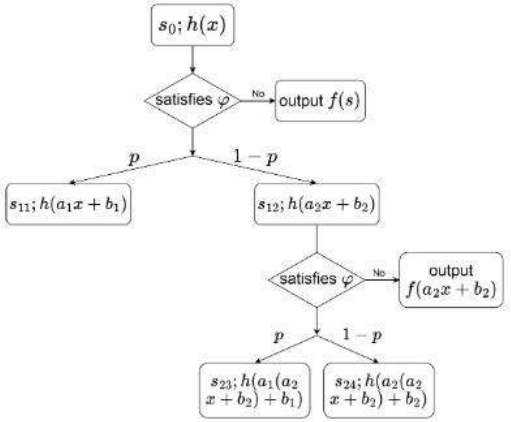
return $f(x)$

2-induction:

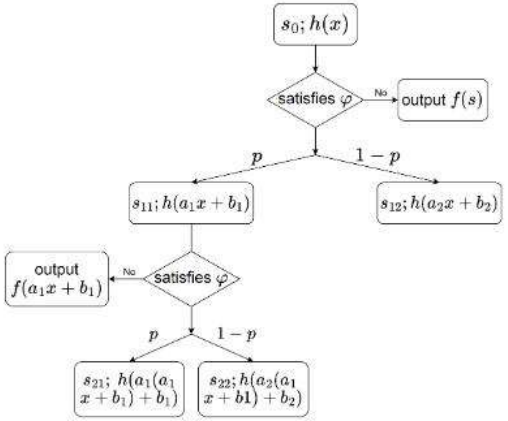
$\Phi(\Psi_h^1(h)) \leq h$



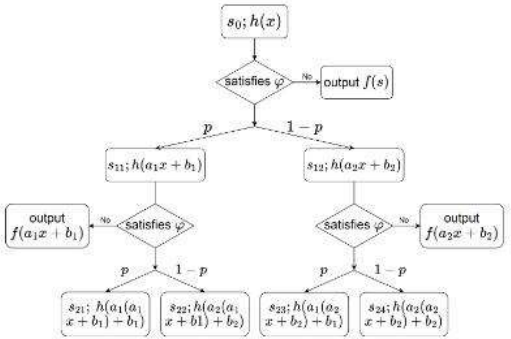
(a) **Case 1:** C_1



(b) **Case 2:** C_2



(c) **Case 3:** C_3



(d) **Case 4:** C_4

A Simplified Growing Walk Example

```
while (x >=0) {
    if (flip(0.5)){x = -1}
    else {x += 1;
          y += x;}
};
return y
```

Deriving k-induction-based constraints:

$$\min\{h_1, h_2, \ldots, h_n\} \preccurlyeq h$$

Each h_i is a piecewise function

Transforms to a Canonical Form:

$$[B_1] \Rightarrow \min\{e_{11}, \ldots, e_{m1}\} \leq h, , \ldots,$$
$$[B_l] \Rightarrow \min\{e_{1l}, \ldots, e_{ml}\} \leq h$$

$$[x < 0] \implies \min\{y\} \leq h$$

$$[x \geq 0] \implies \min \begin{Bmatrix} 0.5 \cdot h(x+1, x+y+1) + 0.5 \cdot h(-1, y) \\ 0.25 \cdot h(-1, y+x+1) + 0.25 \cdot h(x+2, 2x+y+3) + 0.5 \cdot h(-1, y) \\ 0.25 \cdot h(-1, y+x+1) + 0.25 \cdot h(x+2, 2x+y+3) + 0.5 \cdot y \\ 0.5 \cdot h(x+1, x+y+1) + 0.5 \cdot y \end{Bmatrix} \leq h$$

Constraints Solving——Linear Case

$$[x < 0] \implies \min\{y\} \le h$$

$$[x \ge 0] \implies \min \begin{cases} 0.5 \cdot h(x+1, x+y+1) + 0.5 \cdot h(-1, y) \\ 0.25 \cdot h(-1, y+x+1) + 0.25 \cdot h(x+2, 2x+y+3) + 0.5 \cdot h(-1, y) \\ 0.25 \cdot h(-1, y+x+1) + 0.25 \cdot h(x+2, 2x+y+3) + 0.5 \cdot y \\ 0.5 \cdot h(x+1, x+y+1) + 0.5 \cdot y \end{cases} \le h$$

$$\exists \lambda_0 \ge 0, \lambda_1 \ge 0, \cdots, \lambda_5 \ge 0 \quad \textbf{s.t.} \quad (\lambda_2 \ne 0 \vee \lambda_3 \ne 0 \vee \lambda_4 \ne 0 \vee \lambda_5 \ne 0) \wedge$$
$$0 = (-1) \cdot \lambda_1 + 0.5(a-b) \cdot \lambda_2 + 0.75(a-b) \cdot \lambda_3 + 0.75(a-b) \cdot \lambda_4 + 0.5(a-b) \cdot \lambda_5 \wedge$$
$$0 = 0.5(b-1) \cdot \lambda_4 + 0.5(b-1) \cdot \lambda_5 \wedge$$
$$0 = -0.5b \cdot \lambda_2 - (b - 0.25a) \cdot \lambda_3 + (0.5c - 0.25a - b) \cdot \lambda_4 + 0.5(c - a - b) \cdot \lambda_5 - \lambda_0$$

Constraints Solving:

1. Transform $\forall x, \wedge S_i \Rightarrow T_i$ into $\wedge(S_i \wedge \neg T_i)$ is unsatisfiable (note that $\neg T_i$ can eliminate the pointwise minimum operation)

2. Consider a variant of Motzkin's Transposition Theorem and give a proof.

3. Call the bilinear solver

Constraints Solving——Polynomial Case

Transforms to a Canonical Form:

$$[B_1] \Rightarrow \min\{e_{11}, \dots, e_{m1}\} \leq h, , \dots, [B_l] \Rightarrow \min\{e_{1l}, \dots, e_{ml}\} \leq h$$

Relaxing by pulling the minimum inside the implications out as disjunction:

$$([B_1] \Rightarrow (e_1 \leq h) \vee \cdots \vee [B_1] \Rightarrow (e_{m1} \leq h)), \dots$$
$$([B_l] \Rightarrow (e_{1l} \leq h) \vee \cdots \vee [B_l] \Rightarrow (e_{ml} \leq h))$$

By Distributive Law:

$$([B_1] \Rightarrow e_1 \leq h \wedge [B_2] \Rightarrow e_{12} \leq h \wedge \cdots \wedge [B_l] \Rightarrow e_{1l} \leq h) \vee$$
$$\dots \vee$$
$$([B_1] \Rightarrow e_{m1} \leq h \wedge [B_2] \Rightarrow e_{m2} \leq h \wedge \cdots \wedge [B_l] \Rightarrow e_{ml} \leq h)$$

For each case, apply Putinar's Positivstellensatz.

Our Template Algorithm

- Establish polynomial template
- Derive k-induction condition constraints
- Transforms to a Canonical Form
- Polynomial solving (Linear case and Polynomial Case)

Experimental Evaluation (Piecewise Linear)

Benchmark	f	Inv	$k=1$	2-induction		3-induction		Piecewise Upper Bound
				Time(s)	Solution	Time(s)	Solution	
Geo	x	$[0 \le x]$	-	0.59	$x+1$	1.92	$x+1$	$[c>0]\cdot x+[c\le 0]\cdot(x+1)$
k-Geo	y	$[0\le x \wedge 0\le y \; k\le N+1]$	$-k+N$ $+x+y+1$	100.94	$-k+N$ $+x+y+1$	132.76	$-k+N$ $+x+y+1$	$[k>N]\cdot y+$ $[k\le N-1]\cdot(-k+N+x+y+1)+$ $[N-1<k\le N]\cdot(-0.5k+0.5N+x+y+1)$
Bin-ran	y	$[0\le i\le 11 \wedge$ $0\le x \wedge 0\le y]$	-	106.29	$0.9x-21i$ $+y+233$	TO	-	$[i>10]\cdot y+$ $[\frac{90}{11}<i\le 10]\cdot(0.9x-21i+y+233)$ $[i\le \frac{90}{11}]\cdot(0.9x-18.8i+y+215)$
Coin	i	$[0\le x\le 1 \wedge$ $0\le y\le 1 \wedge$ $0\le i$	-	104.13	$i+\frac{8}{3}$ **or** $i+\frac{8}{3}y-\frac{8}{3}x+\frac{8}{3}$	437.06	$i+\frac{8}{3}$	$[x\ne y]\cdot i+$ $[x=y]\cdot(i+\frac{8}{3})$
Mart	i	$[0\le x]$	-	0.7	$i+2$	19.29	$i+2$	$[x\le 0]\cdot i+[x>0]\cdot(i+2)$
Growing Walk	y	$[-1\le x]$	-	1.91	$x+y+2$	4.03	$x+y+2$	$[x<0]\cdot y+[x\ge 0]\cdot(x+y+2)$
Growing Walk -variant	y	$[-1\le x]$	$x+y+1$	2.60	$x+y+1$	125.19	$x+y+1$	$[x<0]\cdot y+$ $[0\le x<1]\cdot(0.5x+y+0.25)$ $+[x\ge 1]\cdot(x+y)$
Expected Time	t	$[-1\le x\le 10]$	-	100.74	$12.1925x+t$ $+13.5502$	109.35	$4.4280x+t+6.2461$	$[x<0]\cdot t+[0\le x<1]\cdot(t+1)$ $[1\le x<3.258]\cdot(3.9852x+t+7.39)$ $[3.258\le x<3.3772]\cdot(4.4280x+t+6.2461)$ $[3.3772\le x]\cdot(3.5867x+t+9.0874)$
Zero-Conf -variant	cur	$[0\le start\le 1 \wedge$ $0\le est\le 1]$	-	105.09	$-140est+$ $cur+140$	180.42	$cur+140$	$[est>0]\cdot cur+$ $[start==0\wedge est\le 0]\cdot(cur+140)$ $+[start\ge 1\wedge est\le 0]\cdot(cur+42)$
Equal-Prob-Grid	$goal$	$[0\le a\le 10 \wedge$ $0\le b\le 10$ $goal\ge 0]$	-	142.68	$goal+1.5$	TO	-	$[a>10\vee b>10\vee goal\ne 0]\cdot goal$ $[a\le 10\wedge b\le 10\wedge goal=0]\cdot 1.5$
RevBin	z	$[x\ge 0]$	$2x+z$	100.73	$2x+z$	70.30	$2x+z$	$[x<1]\cdot z+[1\le x<2]\cdot(z+x+1)$ $+[x\ge 2]\cdot(z+2x)$
Fair Coin	i	$[0\le x\le 1 \wedge$ $0\le y\le 1]$	$i-2y+2$	66.13	$i+\frac{4}{3}$	129.34	$i+\frac{4}{3}$	$[x>0\vee y>0]\cdot i+$ $[x\le 0\wedge y\le 0]\cdot(i+\frac{4}{3})$
St-Petersburg variant	y	$[0\le x\le 1 \wedge$ $y\le 0]$	-	0.56	$\frac{3}{2}y$	1.53	$\frac{3}{2}y$	$[x>0]\cdot y+[x\le 0]\cdot\frac{3}{2}y$

Experimental Evaluation (Piecewise Linear)

Comparison with monolithic polynomial (via 1-induction)

Benchmark	f	Our Approach		Monolithic Polynomial via Simple Induction ($k=1$)	PCT
		k	Piecewise Upper Bound		
Geo	x	3	$[c>0]\cdot x+$ $[c\le 0]\cdot(x+1)$	$1.0000-1.9996*c+1.0000*x+$ $0.9996*c^2-0.0002*x*c+0.0002*x*c^2$	0.0%
k-Geo	y	3	$[k>N]\cdot y+$ $[k\le N-1]\cdot(-k+N*x+y+1)+$ $[N-1<k\le N]\cdot(-0.5k+0.5N+x+y+1)$	$274.1142-53.62281*N-1.0000*k+$ $1.0000*y+1.0000*x+2.7311*N^2$	2.82%
Bin-ban	y	2	$[i>10]\cdot y+$ $[\frac{90}{11}<i\le 10]\cdot(0.9x-21i+y+233)$ $[i\le\frac{90}{11}]\cdot(0.9x-18.8i+y+215)$	$66.8036+21.0161*i-29.5267*y-$ $17.6524*x-1.5735*i^2-0.2059*y*i-$ $0.0157*y^2-0.4056*x*i-0.2380*x*y-$ $1.7910*x^2-0.0102*i^3+0.2917*y*i^2+$ $0.0103*y^2*i-0.0045*y^3+0.4251*x*i^2-$ $0.0036*x*y*i-0.0095*x*y^2+0.6938*x^2*i-$ $0.03827*x^2*y+0.6886*x^3$	49.59%
Coin	i	3	$[x\neq y]\cdot i+$ $[x=y]\cdot(i+\frac{3}{5})$	$2.6667+1.0000*i-0.6381*y+$ $4.2840*x-2.0286*y^2-2.0067*x*y$ $+0.3893*x^2$	0.0%
Mart	i	3	$[x\le 0]\cdot i+$ $[x>0]\cdot(i+2)$	$0.0248+1.0000*i+$ $199999.6588*x+0.1643*x^3$	0.0%
Growing Walk	y	3	$[x<0]\cdot y+$ $[x\ge 0]\cdot(x+y+2)$	$2.5000+1.0000*y+1.900*x$ $-0.5000*x^2+0.1000*x^3$	0.0%
Growing Walk variant	y	3	$[x<0]\cdot y+$ $[0\le x<1]\cdot(0.5x+y+0.25)$ $+[x\ge 1]\cdot(x+y)$	$1.0000*y-0.2380*x+0.1041*y^2-$ $0.0686*x*y+0.0951*x^2+0.03558*x*y^2$ $+0.0686*x^2*y+0.1430*x^3$	5.52%
Expected Time	t	3	$[x<0]\cdot t+[0\le x<1]\cdot(t+1)$ $[1\le x<3.258]\cdot(3.9852x+t+7.39)$ $[3.258\le x<3.3772]\cdot(4.4280x+t+6.2461)$ $[3.3772\le x]\cdot(3.5867x+t+9.0874)$	$3.1203+0.9622*t+2.8278*x+$ $0.0015*t^2-0.01558*x*t-0.1397*x^2-$ $0.0003*x*t^2-0.0002*x^2*t+0.0025*x^3$	50.0%
Zero-Cond -variant	cur	3	$[est>0]\cdot cur+$ $[start==0\wedge est\le 0]\cdot(cur+140)$ $+[start\ge 1\wedge est\le 0]\cdot(cur+42)$	$109.8660-0.1357*cur+293795.0410*start+$ $209178.7117*est+0.0019*cur^2+0.7202*start*cur-$ $293865.0570*start^2+1.0313*est*cur+$ $274251.8886*est*start-209283.0750*est^2$	0.5 %
Equal-Prob-Grid	$goal$	2	$[a>10\vee b>10\vee goal\neq 0]\cdot goal$ $[a\le 10\wedge b\le 10\wedge goal=0]\cdot 1.5$	$1.6661+5.7396*goal-9.4857*10^{-5}*b+$ $1.5707*10^{-5}*a+0.6003*goal^2-0.6740*b*goal$ $+1.5975^{1}0{-}5*b^2+2.2074*10^{-5}*a*goal$	0.0%
RevBin	z	3	$[x<1]\cdot z+$ $[1\le x<2]\cdot(z+x+1)$ $+[x\ge 2]\cdot(z+2x)$	$1.0000*z+2.0000*x$	0.0%
Fair Coin	i	3	$[x>0\vee y>0]\cdot i+$ $[x\le 0\wedge y\le 0]\cdot(i+\frac{4}{5})$	$1.3333+1.0000*i-0.4141*y-$ $0.4141*x+1.1743*i^2-2.3486*y*i+$ $0.2551*y^2-2.3486*x*i+3.6820*x*y$ $+0.2551*x^2$	0.0%
St-Petersburg variant	y	3	$[x>0]\cdot y+[x\le 0]\cdot\frac{1}{2}y$	$0.0197+1.5047*y+371727.7656*x$ $-0.5028*x*y-371727.7734*x^2$	0.0%

Experimental Evaluation (Piecewise Polynomial)

Benchmark	f	Inv	deg	Our Algorithms			Exist	
				Solution h^*	Time(s)	Piecewise Upper Bound	Exact Invariant	Time(s)
GeoAr	x	$[0 \le x \wedge$ $0 \le y \wedge$ $0 \le z$	2	$2.0 + y + x$	1.11	$[z > 0] \cdot (x + y + 2) + [z \le 0] \cdot x$	-	-
Bin0	x	$[0 \le x \wedge$ $0 \le y \wedge$ $0 \le n$	2	$x + 0.5 * y * n$	1.3	$x + [n > 0] \cdot 0.5 * y * n$	$x + [n > 0] \cdot$ $0.5 * y * n$	79.04
Bin2	x	$[0 \le x \wedge$ $0 \le y \wedge$ $0 \le n$	2	$0.25 * n + x + 0.25 * n^2 + 0.5 * y * n$	1.5	$x + [n > 0] \cdot (0.25 * n + x + 0.25 * n^2 + 0.5 * y * n)$	$x + [n > 0] \cdot (0.25 * n$ $+x + 0.25 * n^2$ $+0.5 * y * n)$	250.60
DepRV	$x * y$	$[0 \le x \wedge$ $0 \le y \wedge$ $0 \le n$	2	$-0.25 * n + 0.25 * n^2 + 0.5 * y * n$ $+0.5 * x * n + x * y$	1.51	$[n > 0] \cdot (-0.25 * n + 0.25 * n^2 +$ $0.5 * y * n + 0.5 * x * n + x * y)$ $+[n \le 0] \cdot x * y$	-	-
Prinsys	$[x == 1]$	$[-1 \le x \le 1]$	2	$0.5 + 0.5 * x$	2.57	$[x == 1] * 1 +$ $[x == 0] * 0.5$	$[x == 1] * 1 +$ $[x == 0] * 0.5$	3.02
CAV-7	$[c \le 10]$	$[0 \le i \wedge$ $0 \le c \wedge$ $i \le 5$	3	$0.99062 - 0.01064 * c + 0.0091 * i +$ $0.00109 * c^2 - 0.00461 * i * c + 0.00398 * i^2 -$ $3.0e-5 * c^3 - 9.0e-5 * i * c^2$ $+0.00091 * i^2 * c - 0.00051 * i^3$	5.93	$\min\{[i <= 4] \cdot (0.989822 + 0.01265 * c -$ $0.009756 * i + 0.00336 * c^2 - 0.002394 * i * c +$ $0.00053 * i^2 - 0.00051 * c^3 + 0.000728 * i * c^2$ $-5.4e-5 * i^2 * c - 1.2e-5 * i^3) + [i > 4] \cdot [c <= 10], h^*\}$	-	-
CAV-5	$[i \ge 10]$	$[0 \le i \wedge$ $-1 \le money]$	3	$0.04917 - 0.11675 * money + 0.09427 * i -$ $0.15015 * money^2 + 0.01759 * i * money + 0.00833 * i^2 -$ $0.02099 * money^3 - 0.01539 * i * money^2$ $-0.00286 * i^2 * money - 0.0002 * i^3$	4.09	$\min\{[money \ge 0] \cdot (-0.252504 + 0.093682 * money$ $-0.480242 * i + 0.00559 * money^2 -$ $0.012046 * i * money - 0.210042 * i^2 - 0.0002 * money^3$ $-0.00286 * i * money^2 - 0.01539 * i^2 * money -$ $0.02099 * i^3) + [money < 0] \cdot [i \ge 10], h^*\}$	-	-
Add	$[x > 5]$	$[0 \le x \wedge$ $0 \le y \wedge$ $y \le 2$	4	$0.57842 + 11917.63974 * y - 0.20264 * x -$ $13444.64271 * y^2 + 605.58152 * x * y + 0.00691 * x^2 -$ $8877.43347 * y^3 - 1253.692 * x * y^2 - 4.14294 * x^2 * y -$ $0.0045 * x^3 + 10405.09909 * y^4 + 648.20042 * x * y^3 +$ $4.16388 * x^2 * y^2 + 0.00093 * x^3 * y + 9.0e-5 * x^4$	4.46	$\min\{[y \le 1] \cdot (0.55084 + 10018.55345 * y -$ $0.18411 * x - 7284.60638 * y^2 + 486.3985 * x * y +$ $0.00073 * x^2 - 34.79386 * y^3 - 858.10954 * x * y^2 -$ $2.475156 * x^2 * y - 0.004026 * x^3 + 10405.09909 * y^4 +$ $648.20042 * x * y^3 + 4.16388 * x^2 * y^2 + 0.00093 * x^3 * y$ $+9.0e-5 * x^4) + [y > 1] \cdot [x > 5], h^*$	-	-

Experimental Evaluation (Piecewise Polynomial)

Comparison with monolithic polynomial (via 1-induction)

Benchmark	f	Our Approach			Monolithic Polynomial via 1-induction			PCT
		d	T(s)	Piecewise Polynomial Upper Bound	d	T(s)	Monolithic Polynomial Upper Bound	
GeoAr	x	2	1.77	$\min(\{[z > 0] \cdot (0.00015 * x^2 + 0.00028 * x * y - 0.00012 * x * z + 0.00048 * y^2 - 0.00017 * y * z + 0.00018 * z^2 + 0.99834 * x + 1.50004 * y + 0.00048 * z + 3.50925) + [z \leq 0] \cdot x, h^*\})$	3	0.46	$-0.0001 * x^3 - 0.0063 * x^2 * z - 0.0016 * x * y^2 - 0.0566 * x * y * z + 0.572 * x * z^2 + 0.0051 * y^3 + 0.0058 * y^2 * z - 2.3985 * y * z^2 + 57624.6898 * z^3 + 0.0001 * x^2 + 0.0106 * x * y - 0.3802 * x * z - 0.0608 * y^2 + 2.9711 * y * z - 115246.0643 * z^2 + 0.9988 * x + 0.1765 * y + 57625.0405 * z + 0.472$	2.7%
Bin0	x	2	6.28	$x + [n > 0] \cdot 0.5 * y * n$	3	0.30	$0.5 * y + n + x$	0.0%
Bin2	x	2	6.03	$x + [n > 0] \cdot (0.25 * n + x + 0.25 * n^2 + 0.5 * y * n)$	3	0.34	$0.25 * n + x + 0.25 * n^2 + 0.5 * y * n$	0.0%
DetRV	$x * y$	2	5.96	$[n > 0] \cdot (-0.25 * n + 0.25 * n^2 + 0.5 * y * n + 0.5 * x * n + x * y) + [n \leq 0] \cdot x * y$	3	0.31	$-0.24988 * n + 0.24999 * n^2 + 0.00001 * y + 0.5 * y * n + 0.49998 * x * n + x * y + 0.0001$	5.4%
Prevsys	$[x == 1]$	2	2.04	$[x == 1] * 1 + [x == 0] * 0.5$	3	0.22	$0.2973 * x^2 + 0.2027 * x + 0.5$	0.0%
CAV-7	$[x \leq 30]$	3	13.42	$\min(\{[i <= 4] \cdot (-0.00010i^3 + 0.00040 * i^2 * x - 0.00023i * x^2 + 0.00003 * x^3 + 0.00080 * i^2 - 0.00155 * i * x + 0.00084 * x^2 + 0.00339 * i - 0.00392 * x + 0.99782) + [i > 4] \cdot [x <= 10], h^*\})$	5	0.36	$0.00012 * i^3 + 0.00008 * i^2 * x - 0.00031 * i^2 * x^2 + 0.00016 * i^2 * x^3 - 0.00004 * i * x^4 + 0.00001 * x^5 + 0.00031 * i^2 - 0.0026 * i^2 * x + 0.00252 * i * x^2 - 0.00055 * i * x^3 + 0.00013 * x^4 - 0.00084 * i^2 + 0.00585 * i^2 * x - 0.00193 * i * x^2 - 0.00165 * x^3 + 0.01018 * i^2 - 0.02507 * i * x + 0.01292 * x^2 + 0.01597 * i - 0.0167 * x + 0.99954$	3.54%
CAV-5	$[i \geq 10]$	3	10.46	$\min(\{[money \geq 0] \cdot (0.00001 * i^3 + 0.00011 * i^2 * money + 0.00051 * i * money^2 + 0.00107 * money^3 - 0.00199 * i^2 - 0.01755 * i * money - 0.03655 * money^2 + 0.10297 * i + 0.34438 * money + 0.38274) + [money < 0] \cdot [i \geq 10], h^*\})$	5	0.35	$-0.00001 * i^3 - 0.00014 * i^2 * money - 0.00279 * i^2 * money^2 + 0.00683 * i^2 * money^3 - 0.1148 * i * money^4 + 0.27662 * money^5 - 0.0006 * i^4 - 0.03539 * i^3 * money - 0.21378 * i^2 * money^2 - 1.39256 * i * money^3 - 2.06087 * money^4 + 0.08149 * i^3 + 1.16915 * i^2 * money - 9.90873 * i * money^2 - 6.12975 * money^3 - 4.75556 * i^2 - 102.83331 * i * money - 347.64019 * money^2 + 182.58802 * i - 560.73996 * money - 0.10574$	86.37%
ADD	$[x > 5]$	3	11.89	$\min(\{[y \leq 1] \cdot (0.00054 * x^3 - 0.00611 * x^2 * y + 0.49496 * x * y^2 + 657.28319 * y^3 - 0.01131 * x^2 - 0.22373 * x * y - 921.46523 * y^2 + 0.08352 * x + 526.19688 * y + 0.92294) + [y > 1] \cdot [x > 5], h^*\})$	5	0.33	$0.0001 * x^4 * y - 0.00274 * x^3 * y^2 + 0.05232 * x^2 * y^3 - 0.0702 * x * y^4 + 106.30308 * y^5 - 0.0007 * x^4 + 0.02127 * x^3 * y - 0.30975 * x^2 * y^2 - 0.74934 * x * y^3 - 648.75956 * y^4 - 0.01305 * x^3 + 0.30582 * x^2 * y + 3.08095 * x * y^2 + 1445.38101 * y^3 - 0.01798 * x^2 - 1.39327 * x * y - 1391.22966 * y^2 - 1.8068 * x + 496.2657 * y + 0.45687$	45.08%

Conclusion and Future Works

- Piecewise Expectation Analysis
  - probabilistic while loops
  - Combination of k-induction and OST
  - polynomial solving

- Future works

  more efficient polynomial solving (e.g., bilinear programming)

Thanks for your attention!

Questions!