Probabilistic Program Analysis and Concentration of Measure

Part I: Concentration of Measure

Sriram Sankaranarayanan
University of Colorado, Boulder
Concentration of Measure: Experiment #1

Heads → Gain one dollar

Tails → Lose one dollar

Repeat 1000 times.

Best Case: +1000 Dollars
Worst Case: -1000 Dollars.
Average Case: 0 Dollars.
Concentration of Measure: Experiment #2

Vehicle on a road.

\[
\begin{align*}
    x(t + 1) &= x(t) + 0.1 \cos(\theta) \\
    y(t + 1) &= y(t) + 0.1 \sin(\theta) \\
    \theta(t + 1) &= 0.8\theta(t) + w \\
    w &\sim \mathcal{N}(0, 0.1)
\end{align*}
\]
**Systems Acting Under Disturbances**

- Stochastic Verification
- Reliability
- Stochastic Controls
- Uncertainty Quantification

- Artificial Intelligence
  - Uncertainty Representations

External Disturbances → System → Output

- “Classic” Formal Verification.
- “Set-Valued” Robust Control.
Topics Covered

• Quantitative Reasoning:
  • Prove bounds on probabilities of assertions.
  • Bounds on expectations/moments.

• Qualitative Reasoning:
  • Almost sure termination, recurrence, persistence, ..
  • Limit behavior.

Please ask questions during the talk!
Papers


Co-Authors

Aleksandar Chakarov
Univ. Colorado, Boulder
now at Phase Change

Olivier Bouissou
CEA, now at Mathworks

Eric Goubault
Ecole Polytechnique

Sylvie Putot
Ecole Polytechnique

Yuen-Lam Voronin
Univ. Colorado, Boulder
Motivating Examples
Example #1: Repetitive Robot

Sawyer Robotic Arm (rethink robotics)

Small errors at each step.

Probability of going out of bounds?

Repeat this 100 times.

angles = [10, 60, 110, 160, 140, ...
          100, 60, 20, 10, 0]

x := TruncGaussian(0, 0.05, -0.5, 0.5)
y := TruncGaussian(0, 0.1, -0.5, 0.5)

for reps in range(0, 100):
    for theta in angles:
        # Distance travelled variation
        d = Uniform(0.98, 1.02)
        # Steering angle variation
        t = deg2rad(theta) * (1 + ...
                               TruncGaussian(0, 0.01, -0.05, 0.05))
        # Move distance d with angle t
        x = x + d * cos(t)
        y = y + d * sin(t)

    # Probability that we went too far?
    assert (x >= 272)
angles = [10, 60, 110, 160, 140, ...
100, 60, 20, 10, 0]
x := TruncGaussian(0, 0.05, -0.5, 0.5)
y := TruncGaussian(0, 0.1, -0.5, 0.5)
for reps in range(0, 100):
    for theta in angles:
        # Distance travelled variation
d = Uniform(0.98, 1.02)
        # Steering angle variation
t = deg2rad(theta) * (1 + ...
            TruncGaussian(0, 0.01, -0.05, 0.05))
        # Move distance d with angle t
        x = x + d * cos(t)
y = y + d * sin(t)
# Probability that we went too far?
assert(x >= 272)
Example #2: UAV Keep Out Zone

\[ y(t + 1) = y(t) + 0.1v \sin(\theta(t)) + 0.1v_w \sin(\theta_w(t)) \]
\[ \theta(t + 1) = 0.95\theta(t) - 0.03y(t) \]
\[ \theta_w(t + 1) \sim 0.6 + w_1 \]
\[ v_w(t + 1) \sim 1 + w_2 \]
\[ v = 4 \]
\[ w_1 \in [-0.1, 0.1], \ E(w_1) = 0, \ E(w_1^2) = 0.01 \]
\[ w_2 \in [-0.1, 0.1], \ E(w_2) = 0, \ E(w_2^2) = 0.01 \]
theta := Uniform(-0.1, 0.1)
y := Uniform(-0.1, 0.1)
for j in range(0, n):
    v := 4
    vw := 1 + random([-0.1, 0.1], 0, 0.01)
    thetaw := 0.6 + random([-0.1, 0.1], 0, 0.01)
    y := y + 0.1 * v * sin(theta) +
        0.1 * vw * sin(thetaw)
    theta := 0.95 * theta - 0.03 * y

Probability(y >= 1.0)
Probability(y <= -1.0)
Anesthesia (Fentanyl) Infusion


\[ u = u(t) + w \]

\[ x_1(t + 1) = 0.9012x_1(t) + 0.0304x_2(t) + 0.0031x_3(t) + 0.2676u \]

\[ x_2(t + 1) = 0.0139x_1(t) + 0.9857x_2(t) + 0.002u \]

\[ x_3(t + 1) = 0.0015x_1(t) + 0.9857x_3(t) + 0.0002u \]

\[ x_4(t) = 0.0838x_1(t) + 0.0014x_2(t) + 0.0001x_3(t) + 0.9117x_4(t) + 0.012u \]
Anesthesia Infusion (Continued)

```python
infusionTimings[7] = {20, 15, 15, 15, 15, 15, 45};
double infusionRates[7] = { 3, 3.2, 3.3, 3.4, 3.2, 3.1, 3.0};
Interval e0(-0.4, 0.4), e1(0.0), e2(0.006,0.0064);
for i in range(0, 7):
    currentInfusion = 20.0*infusionRates[i];
    curTime = infusionTimings[i];
    for j in range(0, 40 * infusionTimings[i]):
        e = 1+ randomVariable(e0, e1, e2)
        u = e * currentInfusion
        x1n = 0.9012 * x1 + 0.0304 * x2 + 0.0031 * x3
             + 2.676e-1 * u
        x2n = 0.0139 * x1 + 0.9857 * x2 + 2e-3*u
        x3n = 0.0015 * x1 + 0.9985 * x3+ 2e-4*u
        x4n = 0.0838 * x1 + 0.0014 * x2 + 0.0001 *x3 +
             0.9117 * x4 + 12e-3 * u
        x1 = x1n;  x2 = x2n;
        x3 = x3;  x4 = x4n
```

\[ P(x_4 \leq 150\text{ng/ml}) \]

\[ P(x_4 \geq 300\text{ng/ml}) \]
Reasoning about Uncertainty

Estimating the probabilities vs. Proving bounds on probabilities.

Random Inputs → Probabilistic Program → Output Property

Demonic Inputs

Probability of Success?

Probability of Failure?

Rare Event \( \leq 10^{-6} \)?
Agenda

• Probabilities and Programs.
  • Probabilistic Properties.

• Concentration of Measure Inequalities.
  • Finite executions, “straight line” programs.

• Martingales and more general programs.
  • Pre-expectation calculus.
  • Reasoning about termination.
  • Reasoning about temporal properties.
Programming with Probabilities

• Imperative programs with random number generation.

```plaintext
real x := Uniform(-1, 1)
real y := Gaussian(2.5, 1.3)
bool b := true
int i := 0
for i in range(0, 100):
    b := Bernoulli(0.5)
    if b:
        x := x + 2 * y + Gaussian(0.5, 1.5)
    else:
        x := 1 - 2.5 * x
        y := 2
assert( x >= y)
```

Random Number Generation Function
Demonic Nondeterminism

- Ignore demonic nondeterminism.
- Focus purely on random variables.

```python
real x := Uniform(-1, 1)
real y := Gaussian(2.5, 1.3)
bool b
int i
for i in range(0, 100):
    b := Bernoulli(0.5)
    if b:
        x := x + 2 * y + Gaussian(0.5, 1.5)
    else:
        x := 1 - 2.5 * x - Choice(-1, 1)
y := 2
assert( x >= y)
```
Parametric Nondeterminism

real $x := \text{Uniform}(-1, 1)$
real $y := \text{Gaussian}(2.5, 1.3)$
bool $b$
int $i$
for $i$ in range(0, 100):
    $b := \text{Bernoulli}(0.5)$
    if $b$:
        $x := x + 2 \times y + \text{Gaussian}(0.5, 1.5)$
    else:
        $x := 1 - 2.5 \times x - \text{RandomVariable}([-1, 1], [-0.1, 0.1], [0.001, 0.0015])$

$y := 2$
assert($x \geq y$)
real $x := \text{Uniform}(-1, 1)$
real $y := \text{Gaussian}(2.5, 1.3)$
bool $b := \text{true}$
int $i := 0$
for $i$ in range(0, 100):
    $b := \text{Bernoulli}(0.5)$
    if $b$:
        $x := x + 2 \times y + \text{Gaussian}(0.5, 1.5)$
    else:
        $x := 1 - 2.5 \times x$
        $y := 2$
assert($x >= y$)

- Probabilistic Program = Markov Process.
- State Variables $(x, y, i, b)$
- Initial Distribution $X_0$
- State Update Rule:

$$(x', y', i', b') = f(x, y, i, b, w_1, w_2)$$

$$(x', y', i', b') = \begin{cases} 
(x + 2y + w_1, y, i + 1, w_2) & i \leq 100 \land w_2 \\
(1 - 2.5x, 2, i + 1, w_2) & i \leq 100 \land \neg w_2 \\
(x, y, i, b) & i > 100 
\end{cases}$$
Concentration of Measure
Concentration of Measure: Experiment #1

Heads → Gain one dollar

Tails → Lose one dollar

Repeat 1000 times.

Best Case: + 1000 Dollars
Worst Case: - 1000 Dollars.
Average Case: 0 Dollars.
What is the distribution of $X_n$?

$$\mathbb{P}(X_n = j) = \binom{n}{(n+j)/2} \frac{1}{2^n}$$
Coin Toss

\[ X_n = C_1 + C_2 + \cdots + C_n \]

What is the probability \( X_n \geq 100 \)?

\[ \sum_{j=100}^{n} \Pr(X_n = j) = \sum_{j=100}^{n} \binom{n}{(n+j)/2} \frac{1}{2^n} \]

**Problem:** Not easy to calculate.

**Solution:** A bound on the probability is good enough.
“Large Deviation” Inequalities

One-Sided Inequality

$$\mathbb{P}(X \leq \mathbb{E}(X) - t) \leq ??$$

Two-Sided Inequality

$$\mathbb{P}(X \geq \mathbb{E}(X) + t) \leq ??$$

$$\mathbb{P}(X \leq \mathbb{E}(X) - t) \leq ??$$

$$\mathbb{P}(|X - \mathbb{E}(X)| \geq t) \leq ??$$
Markov Inequality

• Let $X$ be a **non-negative** random variable.

$$
\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}(X)}{t}, \text{ for all } t > 0
$$

• Corollary (Chebyshev-Cantelli):

$$
\mathbb{P}(|X - \mathbb{E}(X)| \geq t) \leq \frac{\text{Var}(X)}{t^2}, \text{ for all } t > 0
$$
Chernoff-Hoeffding Bounds

1. $C_1, \ldots, C_n \in \{0, 1\}$ independent r.v.

2. $X_n = C_1 + \ldots + C_n$

3. $\mathbb{P}(X_n \geq k)$?

$$\mathbb{E}(X_n) = \sum_{j=1}^{n} \mathbb{E}(C_j) =: \mu_n$$

Theorem (Chernoff’ 52, Hoeffding’ 63):

$$\mathbb{P}(X_n \geq k) \leq \exp \left( \frac{-2(k-\mu_n)^2}{n(b-a)^2} \right)$$
Coin Toss

\[ X_n = C_1 + C_2 + \cdots + C_n \]

What is the probability \( X_n \geq 100 \)?

\[ \mu_n = 0, \ k = 100, \ b = 1, \ a = -1 \]

\[ \mathbb{P}(X_n \geq k) \leq \exp \left( \frac{-2(k-\mu_n)^2}{n(b-a)^2} \right) \quad \mathbb{P}(X_n \geq 100) \leq \exp \left( \frac{-100^2}{2n} \right) \]

\[ \mathbb{P}(X_{1000} \geq 100) \leq \exp(-\frac{100000}{2 \times 10000}) \leq 0.006 \]

\[ C_i = \begin{cases} 
-1 & \text{w.p. } \frac{1}{2} \\
1 & \text{w.p. } \frac{1}{2}
\end{cases} \]
Coin Toss

• Probability bound (0.006) is conservative.
  • Actual value is 10x smaller \( \sim 5 \times 10^{-4} \)

• What information about the coin tosses did we use?

\[
C_i = \begin{cases} 
-1 & \text{w.p. } \frac{1}{2} \\
1 & \text{w.p. } \frac{1}{2}
\end{cases} \quad \Rightarrow \quad C_i \in [-1, 1], \quad \mathbb{E}(C_i) = 0
\]

Could we use higher moments to obtain better bounds?
Bernstein Inequality

Extend Chernoff Inequalities with information about moments

\[ \mathbb{P}\left( \sum_{i=1}^{n} X_i \geq t \right) \leq \exp \left( \frac{-t^2}{2 \sum \mathbb{E}(X_i^2) + \frac{2}{3} Mt} \right) \]

\[ M = \max_{i=1}^{n} |X_i| \]
Quantitative Analysis

Random Inputs

Probabilistic Program

Output Quantity

Lane Keeping Example

\[ (x, y, \theta) \]

\[
y(t + 1) = y(t) + 0.1\theta
\]

\[
\theta(t + 1) = 0.8\theta(t) + w
\]

\[ w \in [-0.01, 0.01] \]

\[ \mathbb{E}(w) = 0 \]

\[ \mathbb{E}(w^2) = 0.001 \]

\[
y := \text{Uniform}(-0.01, 0.01)
\]

\[
\text{th} := \text{Uniform}(-0.01, 0.01)
\]

for i in range(0, n):

\[
y := y + 0.1 * \text{th}
\]

\[
\text{th} := 0.8 * \text{th} + \text{randomw}()
\]

Probability( y >= 1) <= ??
Lane Keeping Example

\[ y := \text{Uniform}(-0.01, 0.01) \]
\[ \text{th} := \text{Uniform}(-0.01, 0.01) \]

for \( i \) in range(0, 10):
\[ y := y + 0.1 \times \text{th} \]
\[ \text{th} := 0.8 \times \text{th} + \text{randomw()} \]

Probability(\( y \geq 0.1 \)) \( \leq \) ??
“Heterogeneous” Chernoff-Hoeffding Bounds

1. $X = Y_1 + \ldots + Y_n$
2. $Y_i$ are independent r.v.
3. $Y_i \in [a_i, b_i]$
4. $\mu : \mathbb{E}(X)$

\[
\mathbb{P}(X \geq \mu + t) \leq \exp \left( \frac{-2t^2}{\sum_{i=1}^{n} (b_i - a_i)^2} \right)
\]

\[
\mathbb{P}(X \leq \mu - t) \leq \exp \left( \frac{-2t^2}{\sum_{i=1}^{n} (b_i - a_i)^2} \right)
\]
Lane Keeping Example

\[ y := \text{Uniform}(-0.01, 0.01) \]
\[ \text{th} := \text{Uniform}(-0.01, 0.01) \]

for \( i \) in range(0, 10):
    \[ y := y + 0.1 \times \text{th} \]
    \[ \text{th} := 0.8 \times \text{th} + \text{randomw()} \]

Probability( \( y \geq 0.1 \) ) \( \leq \) ??

\[ y[10] := \begin{pmatrix} y_0 + 0.357050327w_0 + \\ 0.4463129088w_1 + 0.432891136w_2 + 0.41611392w_3 + \\ 0.3951424w_4 + 0.368928w_5 + 0.33616w_6 + \\ 0.2952w_7 + 0.244w_8 + 0.18w_9 + 0.1w_{10} \end{pmatrix} \]

\[ \mathbb{P}(y[10] \geq 0.1) \leq 0.16 \]
Problem Setup

Random Inputs \((w_0, w_1, \ldots, w_m)\)

\[
y_n = f(w_0, w_1, \ldots, w_m)
\]

\[
\mathbb{P}(y_n \geq t) \leq ?
\]
Setup

Deterministic Control Flow

\[ x := \text{InitialDistribution}() \]

for \( i \) in range(0, n):
    \[ w := \text{RandomInputs}() \]
    \[ x := f(i, x, w) \]

\[ \text{assert}(g(x) \geq t) \]

\[ g(x_n) = F(x_0, w_1, \ldots, w_n) \]
\[ \mathbb{E}(g(x_n)) = ? \]
\[ \mathbb{P}(g(x) \geq t) \leq ? \]

2. Need to estimate expectations and possibly higher moments to apply.
Setup

• Chernoff-Hoeffding bounds:
  • Sums of random variables.
  • Extensions to general functions.

• Solution 1: Affine arithmetic and concentration of measure (Bouissou et al. TACAS‘16).

• Solution 2: Method of Bounded Differences.

```plaintext
x := InitialDistribution()
for i in range(0, n):
  w := RandomInputs()
  x := f(i, x, w)
assert(g(x) >= t)
```
Affine Form Overview

• Affine Form: how program variables depend on the uncertainties.

\[
\begin{align*}
  y &:= \text{Uniform}(-0.01, 0.01) \\
  \theta &:= \text{Uniform}(-0.01, 0.01) \\
  \text{for } i \in \text{range}(0, 10): \\
  &\quad y := y + 0.1 \times \theta \\
  &\quad \theta := 0.8 \times \theta + \text{randomw}() \\
  \text{Probability}( y \geq 0.1) &\leq ??
\end{align*}
\]

\[
\begin{align*}
  y[0] &= y_0 \\
  \theta[0] &= \theta_0 \\
  y[1] &= y_0 + 0.1\theta_0 \\
  \theta[1] &= 0.8\theta_0 + w_0 \\
  y[2] &= y_0 + 0.1\theta_0 + 0.1(0.8\theta_0 + w_0) \\
        &= y_0 + 0.18\theta_0 + 0.1w_0
\end{align*}
\]
Affine Form Definition [Figueirido+Stolfi’04, Bouissou et al.]

\[ x : a_0 + \sum_{i=1}^{n} a_i w_i \]

\[ w_i \in [a_i, b_i] \]
\[ \mathbb{E}(w_i) \in [c_i, d_i] \]
\[ \mathbb{E}(w_i^2) \in [\ell_i, \mu_i] \]
\[ \mathbb{E}(w_i w_j) \in [f_{ij}, g_{ij}] \]

Noise Symbols

Functional dependency graph
Computing with Affine Forms

• Linear operations
  • Addition.
  • Multiplication with scalar.
  • Introduction of fresh random variables.

• Nonlinear Operations
  • Multiplication.
  • Division.
  • Sine, cosine, tan, log, exp,…

• Reasoning with affine forms.
Multiplication of Affine Forms

\[ x_1 : 2 + 3w_1 + 4w_2 \]
\[ x_2 : 1 + w_2 \]
\[ x_1 \times x_2 : 2 + 3w_1 + 6w_2 + 3w_1w_2 + 4w_2^2 \]

\[ \mathbb{E}(w_3) : \mathbb{E}(w_1w_2) (= \mathbb{E}(w_1)\mathbb{E}(w_2) \text{ if independent}) \]
\[ \mathbb{E}(w_4) : \mathbb{E}(w_1^2) \text{ (second moment of } w_1) \]
Nonlinear Operations

• We will restrict ourselves to smooth operations (continuous + differentiable)

• Let $f$ be a $C^k$ function

\[ f(x) : x_0 + f'(x_0)(x - x_0) + f''(x_0) \frac{(x - x_0)^2}{2!} + \cdots + f^{(k)}(\theta) \frac{(x - x_0)^k}{k!} \]

- $x: \text{Affine Form}$
- $x_0: E(x)$
- Fresh noise symbol.
Nonlinear Operation Example

\[ y = \sin(0.2 + w) \]

\[ w \in [0.05, 0.15], \ E(w) : 0.1, \ E(w^2) = 0.01 \]

\[ y : [0.199986, 0.199987] + [0.9553, 0.9554]w - w_1 \]

\[ w_1 \in [0, 0.0043], \ E(w_1) \in [0, 0.00043], \ E(w_1^2) \in [0, 1.84 \times 10^{-7}] \]
Lane Keeping Example

\[ y := \text{Uniform}(-0.01, 0.01) \]

\[ \text{th} := \text{Uniform}(-0.01, 0.01) \]

for \( i \) in range(0, 10):
    \[ y := y + 0.1 \times \text{th} \]
    \[ \text{th} := 0.8 \times \text{th} + \text{randomw()} \]

Probability( \( y \geq 0.1 \)) \leq ???

\[ y[10] : \left( \begin{array}{c}
    y_0 + 0.357050327w_0 + \\
    0.4463129088w_1 + 0.432891136w_2 + 0.41611392w_3 + \\
    0.3951424w_4 + 0.368928w_5 + 0.33616w_6 + \\
    0.2952w_7 + 0.244w_8 + 0.18w_9 + 0.1w_{10}
\end{array} \right) \]

\[ \mathbb{P}(y[10] \geq 0.1) \leq 0.16 \]

\( w_1, \ldots, w_{10} \) are all independent.
Modified Lane Keeping

\[
y = 0 + y_0 + 0.1y_2 - 0.1y_3 + 0.1y_4 - 0.1y_5 + 0.1y_6 - 0.1y_7 + 0.1y_8 - 0.1y_9 + \ldots + 0.1y_{20} - 0.1y_{21}
\]

\[
y := \text{Uniform}(-0.01, 0.01)
\]

\[
\theta := \text{Uniform}(-0.01, 0.01)
\]

for \(i \in \text{range}(0, 10)\):
\[
y := y + 0.1 \times \sin(\theta)
\]

\[
\theta := \text{randomw}()
\]

\[
\text{Probability}(y \geq 0.1) \leq ??
\]
Modified Lane Keeping

\[ y = 0 + y_0 + 0.1y_2 - 0.1y_3 + 0.1y_4 - 0.1y_5 + 0.1y_6 - 0.1y_7 + 0.1y_8 - 0.1y_9 + \cdots + 0.1y_{20} - 0.1y_{21} \]

Idea:
1. “Compress” connected component to a single noise symbol.
2. Use Chernoff Hoeffding Bounds.
Modified Lane Keeping

\[
P(y \leq -0.06) \leq 0.006 \\
P(y \leq -0.03) \leq 0.16 \\
P(y \leq -0.02) \leq 0.45 \\
P(y \leq -0.01) \leq 0.82 \\
P(y \leq -0.006) \leq 0.96 \\
P(y \geq 0.006) \leq 0.96 \\
P(y \geq 0.01) \leq 0.82 \\
P(y \geq 0.02) \leq 0.45 \\
P(y \geq 0.04) \leq 0.082 \\
P(y \geq 0.06) \leq 0.006
\]

```
y := Uniform(-0.01, 0.01)
\mathrm{th} := \text{Uniform}(-0.01, 0.01)
for i in range(0, 10):
    y := y + 0.1 * \sin(\mathrm{th})
    \mathrm{th} := \text{randomw}()
Probability(y \geq 0.1) \leq ??
```

\[
P(y \geq 0.1) = 0
\]
Example #1: Repetitive Robot

Sawyer Robotic Arm (rethink robotics)

Small errors at each step.

Probability of going out of bounds?

Repeat this 100 times.

angles = [10, 60, 110, 160, 140, ...
100, 60, 20, 10, 0]
x := \text{TruncGaussian}(0, 0.05, -0.5, 0.5)
y := \text{TruncGaussian}(0, 0.1, -0.5, 0.5)
for reps in range(0, 100):
  for theta in angles:
    # Distance travelled variation
d = \text{Uniform}(0.98, 1.02)
    # Steering angle variation
t = deg2rad(theta) * (1 + ...
      \text{TruncGaussian}(0, 0.01, -0.05, 0.05))
    # Move distance d with angle t
    x = x + d * \cos(t)
y = y + d * \sin(t)

# Probability that we went too far?
assert(x >= 272)
Example #1: Continued

angles = [10, 60, 110, 160, 140, ...  
        100, 60, 20, 10, 0]
x := TruncGaussian(0,0.05,-0.5,0.5)
y := TruncGaussian(0,0.1,-0.5,0.5)
for reps in range(0,100):
    for theta in angles:
        # Distance travelled variation
dx = Uniform(0.98,1.02)
        # Steering angle variation
t = deg2rad(theta) * (1 + ... TruncGaussian(0,0.01,-0.05,0.05))
        # Move distance d with angle t
        x = x + d * cos(t)
y = y + d * sin(t)
        # Probability that we went too far?
assert(x >= 272)

\[ \mathbb{P}(x \geq 272) \leq 6.2 \times 10^{-7} \]
Example #2: UAV Keep Out Zone

\[
\theta := \text{Uniform}(-0.1, 0.1) \\
y := \text{Uniform}(-0.1, 0.1) \\
\text{for } j \text{ in range}(0, n): \\
\quad v := 4 \\
\quad vw := 1 + \text{random}([-0.1, 0.1], 0, 0.01) \\
\quad \theta_{\text{old}} := 0.6 + \text{random}([-0.1, 0.1], 0, 0.01) \\
\quad y := y + 0.1 \times v \times \sin(\theta) + \\
\quad \quad 0.1 \times vw \times \sin(\theta_{\text{old}}) \\
\quad \theta := 0.95 \times \theta - 0.03 \times y
\]

Probability( \( y \geq 1.0 \) )
Probability( \( y \leq -1.0 \) )

\[
\begin{align*}
\mathbb{P}(y \leq -0.6138812074) &\leq 0.2358960445 \\
\mathbb{P}(y \leq -0.4102788125) &\leq 0.5036707137 \\
\mathbb{P}(y \leq -0.2881173756) &\leq 0.7921352709 \\
\mathbb{P}(y \geq -0.1659559387) &\leq 0.9176758046 \\
\mathbb{P}(y \geq -0.0030740228) &\leq 0.5036707137 \\
\mathbb{P}(y \geq 0.1190874141) &\leq 0.3195611154 \\
\mathbb{P}(y \geq 0.2005283721) &\leq 0.2358960445
\end{align*}
\]
Anesthesia Infusion

\begin{align*}
\text{infusionTimings}[7] &= \{20, 15, 15, 15, 15, 15, 45\}; \\
\text{double infusionRates}[7] &= \{3, 3.2, 3.3, 3.4, 3.2, 3.1, 3.0\}; \\
\text{Interval e0(-0.4, 0.4), e1(0.0), e2(0.006,0.0064);} \\
\text{for i in range(0, 7):} \\
&\quad \text{currentInfusion} = 20.0 \times \text{infusionRates}[i]; \\
&\quad \text{curTime} = \text{infusionTimings}[i]; \\
&\quad \text{for j in range(0, 40 * \text{infusionTimings}[j]):} \\
&\qquad e = 1 + \text{randomVariable}(e0, e1, e2) \\
&\qquad u = e \times \text{currentInfusion} \\
&\qquad x1n := 0.9012 \times x1 + 0.0304 \times x2 + 0.0031 \times x3 + 2.676e-1 \times u \\
&\qquad x2n := 0.0139 \times x1 + 0.9857 \times x2 + 2e-3 \times u \\
&\qquad x3n := 0.0015 \times x1 + 0.9985 \times x3 + 2e-4 \times u \\
&\qquad x4n := 0.0838 \times x1 + 0.0014 \times x2 + 0.0001 \times x3 + 0.9117 \times x4 + 12e-3 \times u \\
&\qquad x1 := x1n; \quad x2 := x2n; \\
&\qquad x3 := x3; \quad x4 := x4n
\end{align*}

\[ \mathbb{P}(x_4 \geq 300 \text{ng/ml}) \leq 7 \times 10^{-13} \]
\[ \mathbb{P}(x_4 \geq 150 \text{ng/ml}) \leq 10^{-23} \]
Concluding Thoughts
Related Approaches

- Monte Carlo Methods
  - Statistical model checking. [Younes+Simmons, Jha et al, Clarke et al.]
  - Importance Sampling [Legay et al.]
  - Semantic Importance Sampling [Hansen et al. TACAS 2015, RV2016]

- Volume Computation
  - Solve the integration exactly (expensive) [Geldenhuys et al, S et al.]
  - Abstract the program by discretizing state space [Abate et al., PRISM]
  - Abstract the distribution by discretizing [Monniaux, Bouissou et al.]
  - Polynomial-Time Approximation [Chistikov et al. TACAS 2015]
Challenge #1: Representing Nonlinear Computations

How do you represent nonlinear computations?

\[
\text{theta} := \text{Uniform}(-0.1, 0.1) \\
y := \text{Uniform}(-0.1, 0.1) \\
\text{for } j \in \text{range}(0, n): \\
\quad v := 4 \\
\quad \text{vw} := 1 + \text{random}([-0.1, 0.1], 0, 0.01) \\
\quad \text{thetaw} := 0.6 + \text{random}([-0.1, 0.1], 0, 0.01) \\
\quad y := y + 0.1 \times v \times \sin(\text{theta}) + \\
\quad \quad 0.1 \times \text{vw} \times \sin(\text{thetaw}) \\
\quad \text{theta} := 0.95 \times \text{theta} - 0.03 \times y
\]

Option 1: Affine Forms.
- Approximations create dependencies.

Option 2: Nonlinear Forms.
- Keeps random variables independent.
- Hard to reason with.
Challenge #2: Conditional Branches

theta := Uniform(-0.1, 0.1)
y := Uniform(-0.1, 0.1)
for j in range(0, n):
    v := 4
    vw := 1 + random([-0.1, 0.1], 0, 0.01)
    thetaw := 0.6 + random([-0.1, 0.1], 0, 0.01)
    y := y + 0.1 * v * sin(theta) +
        0.1 * vw * sin(thetaw)
    if y >= 0.1
        theta := theta - 0.1
    if y <= -0.1
        theta := theta + 0.1
Probability(y >= 1.0)
Probability(y <= -1.0)

Approach #1: Smoothing the Indicator Function.

Approach #2: Moment method.

- Bounds using the problem of moments.
- “Design your own” inequalities.
Probabilistic Program Analysis and Concentration of Measure

Part II: Martingale

Sriram Sankaranarayanan
University of Colorado, Boulder
Concentration of Measure: Experiment #1

Heads $\rightarrow$ Gain one dollar

Tails $\rightarrow$ Lose one dollar

Repeat N times.

At some point in the experiment:

- I have won $X_i$ dollars thus far.
- If I toss once more, how much do I expect to have?

$$
\mathbb{E}(X_{i+1} \mid X_i) = \frac{1}{2}(X_i + 1) + \frac{1}{2}(X_i - 1) = X_i
$$

Expected fortune in next step = fortune in current step.
Concentration of Measure: Experiment #2

Vehicle on a road. \((x, y, \theta)\)

\[ y(t + 1) = y(t) + 0.1\theta \]
\[ \theta(t + 1) = 0.99\theta(t) + w \]
\[ w \in [-0.01, 0.01] \]
\[ \mathbb{E}(w) = 0 \]

\[ M(t) : y(t) + 10\theta(t) \]

\[
\mathbb{E}(M(t + 1) \mid y(t), \theta(t)) = \mathbb{E}(y(t) + 0.1\theta(t) + 10(0.99\theta(t) + w)) \\
= y(t) + 0.1\theta(t) + 9.9\theta(t) + \mathbb{E}(w) \\
= y(t) + 10\theta(t) = M(t)
\]

Expected value in next step = value in current step.
Conditional Expectation

$$E(X | Y) := f(y)$$

$$E(X | Y) = \lambda y. E(X|Y = y)$$

$$\sum_{x \in \mathcal{X}} x \cdot p(X = x | Y = y) = \sum_{x \in \mathcal{X}} x \frac{P(X = x, Y = y)}{p(Y = y)}$$

$$\frac{1}{f_Y(y)} \int x f_{X,Y}(x, y) dx$$

$$P(Y=y) > 0$$
Martingale

Martingale is a special kind of stochastic process.

\[ X_0, X_1, X_2, \ldots \]
\[ \mathbb{E}(X_{i+1} \mid X_i, \ldots, X_0) = X_i \]

Revisit Experiment #1 and #2 slides now!
Super/SubMartingales

Supermartingale: \[ \mathbb{E}(X_{i+1} \mid X_i, \ldots, X_0) \leq X_i \]

Submartingale: \[ \mathbb{E}(X_{i+1} \mid X_i, \ldots, X_0) \geq X_i \]
First Properties of (Super) Martingales

$X_0, \ldots, X_n$ Martingale

$\mathbb{E}(X_n) = \mathbb{E}(X_0)$

$X_0, \ldots, X_n$ Supermartingale

$\mathbb{E}(X_n) \leq \mathbb{E}(X_0)$
“Adapted” Martingales

\[ X_1, \ldots, X_n \]

\[ \mathbb{E}(f(X_{i+1}) \mid X_i, \ldots, X_1) = f(X_i) \]
Why Martingales?

- **Quantitative:** Concentration of measure involving martingales.

- **Qualitative:** Convergence theorems and proofs of temporal properties.
Martingales and Concentration of Measure (Azuma’s Inequality).
Lipschitz Condition

\[ X_0, \, X_1, \, X_2, \, \ldots \]

Lipschitz (Bounded Difference) Condition:

\[ |X_i - X_{i-1}| \leq c_i, \quad i > 0 \]
Azuma’s Inequality for Martingales

$X_0, \ldots, X_n$ stochastic process.

Lipschitz Condition: $|X_i - X_i| \leq c_i$

Supermartingale:
$$\mathbb{P}(X_n \geq \mathbb{E}(X_n) + t) \leq \exp\left(\frac{-t^2}{2 \sum_{i=1}^{n} c_i^2}\right)$$

Submartingale:
$$\mathbb{P}(X_n \leq \mathbb{E}(X_n) - t) \leq \exp\left(\frac{-t^2}{2 \sum_{i=1}^{n} c_i^2}\right)$$
Coin Toss Experiment

\[ X_n = \sum_{j=0}^{n} C_j, \quad C_j : \begin{cases} -1 & \text{with prob. } \frac{1}{2} \\ 1 & \text{with prob. } \frac{1}{2} \end{cases} \]

Lipschitz Condition:

\[ |X_n - X_{n-1}| \leq 2 \]

Azuma theorem:

\[ \mathbb{P}(X_n \geq t) \leq \exp \left( \frac{-t^2}{8n} \right) \]
\[ \mathbb{P}(X_n \leq t) \leq \exp \left( \frac{-t^2}{8n} \right) \]

Chernoff-Hoeffding:

\[ \mathbb{P}(X_n \geq t) \leq \exp \left( \frac{-t^2}{2n} \right) \]

Azuma theorem:

No independence assumption.
Doob Martingale or the Method of Bounded Differences
Problem Statement

Random Inputs \( (w_0, w_1, \ldots, w_m) \)

\[
y_n = f(w_0, w_1, \ldots, w_m)
\]

\[
P(y_n \geq t) \leq ?
\]
Doob Sequence

\[ f(W_1, W_2, \ldots, W_n) \]

\[ W_1, \ldots, W_n \text{ are independent} \]

\[ X_0 : \mathbb{E}(f(W_1, \ldots, W_n)) \]

\[ X_1 : \mathbb{E}(f(W_1, W_2, \ldots, W_n) \mid W_1) \]

\[ X_2 : \mathbb{E}(f(W_1, W_2, \ldots, W_n) \mid W_1, W_2) \]

\[ \vdots \]

\[ X_n : \mathbb{E}(f(W_1, W_2, \ldots, W_n) \mid W_1, \ldots, W_n) \]

Constant

\[ \mathbb{E}(f(w_1, W_2, \ldots, W_n)) \]

\[ \mathbb{E}(f(w_1, w_2, \ldots, W_n)) \]

\[ f(w_1, \ldots, w_n) \]
Doob Sequences are Martingales

\[ \mathbb{E}(X_{j+1} \mid W_j, \ldots, W_1) = \mathbb{E}(\mathbb{E}(f(W_1, \ldots, W_n) \mid W_1, \ldots, W_{j+1}) \mid W_1, \ldots, W_j) \]

\[ = \mathbb{E}(f(W_1, \ldots, W_n) \mid W_1, \ldots, W_j) \]

\[ = X_j \]

\[ \mathbb{E}(\mathbb{E}_B(X \mid A, B) \mid A) = \mathbb{E}(X \mid A) \]
Method of Bounded Differences

\[ f(W_1, W_2, \ldots, W_n) \]

\[ \mathbb{P} \left( f(w_1, \ldots, w_n) \begin{cases} \geq \mathbb{E}(f) + t \\ \leq \mathbb{E}(f) - t \end{cases} \right) \leq ?? \]

\( W_1, \ldots, W_n \) are independent

Lipschitz Condition:

\[ |f(w_1, \ldots, w_j, \ldots, w_n) - f(w_1, \ldots, \hat{w}_j, \ldots, w_n)| \leq c_j \]

Azuma Inequality Applied to Doob Martingale:

\[ \mathbb{P} \left( f(w_1, \ldots, w_n) \begin{cases} \geq \mathbb{E}(f) + t \\ \leq \mathbb{E}(f) - t \end{cases} \right) \leq \exp \left( \frac{-t^2}{2 \sum_{j=1}^{n} c_j^2} \right) \]
Application to Programs

Random Inputs \((w_0, w_1, \ldots, w_m)\)

Probabilistic Program

Output Quantity \((y)\)

Output Quantity \((y)\)

\[ y_n = f(w_0, w_1, \ldots, w_m) \]

\[ P(y_n \geq t) \leq ? \]

1. Estimate Lipschitz bounds for each variable.
   - How? [Open Problem].

2. Apply Method of Bounded Differences.
Direct Application of Azuma’s Theorem
Concentration of Measure: Experiment #2

Vehicle on a road. \((x, y, \theta)\)

\[
\begin{align*}
\theta(t) \geq 0 \land y(t) \geq L \\
\theta(t) \leq 0 \land y(t) \leq -L
\end{align*}
\]

\[
y(t + 1) = y(t) + 0.1\theta \\
\theta(t + 1) = 0.99\theta(t) + w \\
w \in [-0.01, 0.01] \\
\mathbb{E}(w) = 0 \\
M(t) : y(t) + 10\theta(t)
\]

\[
\begin{align*}
\theta(t) \leq 0 \land y(t) \geq L & \Rightarrow y(t) + 10\theta(t) \geq L \\
\theta(t) \leq 0 \land y(t) \leq -L & \Rightarrow y(t) + 10\theta(t) \leq -L
\end{align*}
\]
Experiment #2: Azuma’s Inequality

\[ M(t) : y(t) + 10\theta(t) \]
\[ M(0) : y(0) + 10\theta(0) \]

\[ \mathbb{E}(M(t)) = \mathbb{E}(M(0)) = 0 \]

Lipschitz Condition:

\[
|M(t + 1) - M(t)| = |y(t + 1) - y(t) + 10(\theta(t + 1) - \theta(t))| \\
= |0.1\theta(t) + 10(0.99\theta(t) + w(t + 1) - \theta(t))| \\
= |10w(t + 1)| \\
\leq 0.1
\]
Experiment #2: Proving Bounds

\[ \theta(t) \leq 0 \land y(t) \geq L \Rightarrow y(t) + 10\theta(t) \geq L \]

\[ \mathbb{P}(M(t) \geq L) \leq \exp \left( \frac{-L^2}{0.02t} \right) \]

Fix \( t = 100 \)

<table>
<thead>
<tr>
<th>( L )</th>
<th>Azuma Inequality</th>
<th>Chernoff-Hoeffding</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.38</td>
<td>0.93</td>
<td>0.48</td>
</tr>
<tr>
<td>1.5</td>
<td>0.32</td>
<td>7.7 \times 10^{-5}</td>
</tr>
<tr>
<td>3.0</td>
<td>0.011</td>
<td>9.5 \times 10^{-14}</td>
</tr>
<tr>
<td>3.8</td>
<td>0.0073</td>
<td>3.8 \times 10^{-19}</td>
</tr>
</tbody>
</table>


Automatic Inference of Martingales
Concentration of Measure: Experiment #2

Vehicle on a road. \((x, y, \theta)\)

\[
y(t + 1) = y(t) + 0.1\theta \\
\theta(t + 1) = 0.99\theta(t) + w \\
w \in [-0.01, 0.01] \\
\mathbb{E}(w) = 0
\]

\[
M(t) : y(t) + 10\theta(t)
\]

How do we find martingales?
Super Martingales of Probabilistic Programs

\[
x := F(x, w)
\]

Pre-Expectation of \( f \) w.r.t \( S \)

\[
\text{preE}(f, S) : \mathbb{E}(f(x') | x)
\]

\( (x, y) := 2 \times x + \text{Uniform}(-1, 2), -y + \text{Uniform}(-1, 1) \)
Pre-Expectation Example #1

\[(x, y) := 2 \times x + \text{Uniform}(-1, 2), -y + \text{Uniform}(-1, 1)\]

\[\text{preE}(x - y, S) = \mathbb{E}(x' - y' \mid x, y)\]
Pre-Expectation Example #2

\[
(x', y') = \begin{cases} 
(x + U(-1, 2), y - 1) & \text{if } x \geq 0 \\
(2x - U(-1, 1), y - 2) & \text{if } x < 0
\end{cases}
\]

\[
[\varphi] : \begin{cases} 
1 & \text{if } \varphi \\
0 & \text{otherwise}
\end{cases}
\]

\[
\text{preE}(x, \mathcal{S}) = \mathbb{E}(x' \mid x, y)
\]
Loop Supermartingales

\begin{align*}
\text{var } x_1, \ldots, x_n \\
\text{while } (C) \\
\quad \text{do} \\
\quad \quad S \\
\quad \text{od}
\end{align*}

\begin{align*}
f(x_1, \ldots, x_n) \text{ is a martingale expression iff} \\
(\forall x_1, \ldots, x_n) \text{ preE}(f, S) &= f(x_1, \ldots, x_n)
\end{align*}

\begin{align*}
g(x_1, \ldots, x_n) \text{ is a super martingale expression iff} \\
(\forall x_1, \ldots, x_n) \text{ preE}(g, S) &\leq g(x_1, \ldots, x_n)
\end{align*}
Concentration of Measure: Experiment #2

Vehicle on a road. $(x, y, \theta)$

$$y(t + 1) = y(t) + 0.1\theta$$
$$\theta(t + 1) = 0.99\theta(t) + w$$

$w \in [-0.01, 0.01]$  
$\mathbb{E}(w) = 0$

$M(t) : y(t) + 10\theta(t)$

preE$(y + 10 \times \text{th}, S) = y + 10 \times \text{th}$
Automatic Inference of (Super) Martingale

1. Fix an unknown template form of the desired function.
   \[ c_1 y + c_2 \theta \]
2. Use Farkas’ Lemma (theorem of the alternative) to derive constraints \([Colon+S+Sipma’03]\)
3. Solve to obtain (super) martingales.
   \[ c_1 : 1, \quad c_2 : 10 \]
Automatic Inference (Example)

\[
x := x + 0.1(1 - \frac{1}{2}\theta^2)
\]
\[
y := y + 0.1\theta
\]
\[
\theta := 0.99\theta + w
\]
\[
\mathbb{E}(w) = 0
\]

\[
c_1 x^2 + c_2 y^2 + c_3 \theta^2 + c_4 \theta y
\]
\[
+ c_5 x + c_6 y + c_7 \theta + c_8 n
\]

Vehicle on a road. \((x, y, \theta)\)

<table>
<thead>
<tr>
<th>Martingale</th>
<th>Martingale</th>
<th>Martingale</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.985n + 150\theta^2 - 2.985x</td>
<td>10\theta + y</td>
<td>2000\theta y - 199n + 100y^2 + 1990x</td>
</tr>
<tr>
<td>49n - 500x</td>
<td>1000\theta - n</td>
<td>10x - n</td>
</tr>
<tr>
<td>-n - 1000\theta</td>
<td>Supermartingale</td>
<td>Supermartingale</td>
</tr>
<tr>
<td>Supermartingale</td>
<td>Supermartingale</td>
<td>Supermartingale</td>
</tr>
</tbody>
</table>
Further Work on Martingale Inference #1

• Using Doob decomposition [Barthe et al. CAV 2016].

• Start from an given expression and iteratively derive a martingale.
  • Can derive very complex expressions.
  • Lots of avenues for future refinements here.
Further Work on Martingale Inference #2

• Exponential Supermartingales [Tedrake+Steinhardt’ IJRR 2012]
  • Using Sum-of-Squares Inequalities and Semi-Definite Programming.
  • Clever tricks to avoid solving bilinear matrix inequalities.
  • Comparison with Azuma’s inequality may be interesting.
Probabilistic Program Analysis and Concentration of Measure

Part III: *Termination, Persistence and Recurrence, Almost Surely!*

Sriram Sankaranarayanan

University of Colorado, Boulder
Quantitative vs. Qualitative Questions

Random Inputs

Program

What is the probability of blah?

Does the program terminate?
Qualitative Questions

• Almost Sure Termination/Reachability.
  • The program terminates \textit{with probability 1}.
  • All executions eventually reach a desired set \textit{with probability 1}.

• Almost Sure Persistence.
  • The program executions reach a set $S$ and remain in $S$ forever.

• Almost Sure Recurrence
  • The program executions visit $S$ infinitely often.
Almost Sure Termination

```
while (x >= y)
    x := x + Uniform(-1, 1)
y := y + Gaussian(1, 2.0)
```

Does this loop terminate?

Nonterminating execution

\[(10,8) \rightarrow (11,8) \rightarrow (12,8) \rightarrow (13,8) \rightarrow \ldots\]

Almost Sure Termination.
  Terminates with probability 1.
  Measure of samples leading to non-termination is 0.
Proving Termination

while (x  \geq y)
    x := x
    y := y + 1

Ranking Function: x − y

• Decreases by 1 on each loop iteration.
• When negative, loop terminates.

Supermartingale Ranking Function: x − y

\[ \text{preE}(x − y, S) = \mathbb{E}(x' − y'|x, y) \]
\[ = \mathbb{E}(x + U(-1,1) − y − N(1,2)|x, y) \]
\[ = x − y − 1 \]
Supermartingale Ranking Functions (SMRF)

Function of program state: \[ f(x_1, \ldots, x_n) \]

- \[ f \leq 0 \Rightarrow (\neg C) \]
- \[ \text{preE}(f, S) \leq f - \epsilon \]

- “Foster” Lyapunov Criteria (for discrete time Markov Chains).
- Ranking function analogues [Mclver + Morgan]
Main Result

• Let \( f(x_1, \ldots, x_n) \) be a SMRF.

• If \( f \) is positive over the initial state.

• Then \( f \) becomes negative almost surely upon repeatedly executing the loop body.

Corollary of Martingale Convergence Thm. (+ technicalities).
Example # 1

```
real h, t
// h is hare position
// t is tortoise position
while (t <= h)
    if (flip(0.5))
        h := h + uniformRandom(0,2)
    t := t + 1
// Almost sure termination?
```

\[ h - t \]

“Slow and steady wins the race almost surely”
Example #2 : Betting Strategy For Roulette

i := 0;
money := 10, bet
while (money >= 10 ) {
    bet := rand(5,10)
    money := money - bet
    if (flip(36/37)) // bank lost
        if flip(1/3) // col. 1
            if flip(1/2)
                money := money + 1.6*bet // Red
            else
                money := money + 1.2*bet // Black
        elseif flip(1/2) // col. 2
            if flip(1/3)
                money := money + 1.6*bet // Red
            else
                money := money + 1.2*bet // Black
        else // col. 3
            if flip(2/3)
                money := money + 0.4*bet // Red
            i := i + 1
}
Obtaining Completeness

• SMRFs are not complete for proving termination.

```plaintext
x = 0
while (x != 1 and x != -1)
    if (flip(0.5))
        x := x + 1
    else
        x := x - 1
// Almost sure termination
```

The program can be shown to terminate almost surely.

No SMRF exists.

Completeness assuming the time taken to terminate (stopping time) is integrable [Fioriti, Hermanns et al.’15].
Proving bounds on time taken to terminate. [Chatterjee et al.’16, Kaminski et al’16].
Complexity of proving almost sure termination. [Kaminski + Katoen ‘15].
A note of caution...

while C do
  S
  (not C) holds?

  \[ x := 0 \]
  while (x != 1)
    if (flip(0.5))
      x := x + 1
    else
      x := x - 1
  \]

x = 1 holds

x is a martingale of the program
E(x) = 0 at initial state.
E(x) = 0 after each loop iteration.
E(x) = 0 holds when program terminates?

Facts about expected values at each loop iteration
are not necessarily true when the program terminates.

Doob’s Optional Stopping Theorem: Provides condition when we can transfer.

[ Fioriti, Hermanns POPL’15].
Persistence (and Recurrence)
Beyond Termination..

• We are often interested in proving more complex temporal properties.

• Two papers in the same conference!
  • [Chakarov+Voronin+S’ TACAS 16]
  • [Dimitrova+Fioriti+Hermanns+Majumdar’TACAS 16]
  • Both based on ideas using martingale theory.
Room Heater Example [Abate et al. 2010]

\[
x_1' = x_1 + b_1(x_0 - x_1) + a(x_2 - x_1) + c_1(1 - \sigma(x_1)) + \nu_1
\]

\[
x_2' = x_2 + b_2(x_0 - x_2) + a(x_1 - x_2) + c_2(1 - \sigma(x_2)) + \nu_2
\]

![Graph of Room 1, U(-0.01,0.01)]

![Graph of Room 1, N(0, 0.25)]
Almost surely, all behaviors enter $T$ eventually and stay in there forever.
From Termination to Persistence

Target Set

\[ V(x) < 0 \Rightarrow x \in T \]

Decrease Rule

\[ \mathbb{E}(V(x')|x) \leq V(x) - \epsilon \]

Unsound!
Unsoundness of SMRFs for Persistence

V(x) = x satisfies the conditions for SMRF.

The chain visits 0 infinitely often almost surely!
Bounded Increase Condition

Target Set

\[ V(x) < 0 \implies x \in T \]

Decrease Rule

\[ \mathbb{E}(V(x')|x) \leq V(x) - \epsilon \]

Bounded Decrease Condition

\[ V(x') - V(x) \leq M \]

If \( V \) satisfies conditions above, then

\[ \Diamond \Box (T') \text{ almost surely} \]
Room Heater Example [Abate et al. 2010]

\[ x_1' = x_1 + b_1(x_0 - x_1) + a(x_2 - x_1) + c_1(1 - \sigma(x_1)) + \nu_1 \]
\[ x_2' = x_2 + b_2(x_0 - x_2) + a(x_1 - x_2) + c_2(1 - \sigma(x_2)) + \nu_2 \]

Using Sum-Of-Squares Programming

\[ (x_1 - 18.3)^2 + (x_2 - 18.8)^2 \]
Open Directions
Challenge #1: Symbolic Domains

- Incorporate Booleans, Graphs and other domains.
- Common in randomized algorithms.
- Benefit by careful mechanization.

- Application areas:
  - Dynamics on graphs and social networks.
  - Graph rewriting systems (Graph Grammars).
  - Self-assembling systems.
Challenge #2: Concentration of Measure Inequalities

• Understanding when concentration of measure inequalities work.
  • Using more properties about the underlying distributions.

• Designer Inequalities.
  • Symbolic inference of property specific inequalities.
Thank You!

Thank you to University of Minho and the Organizers of the Summer School.

Work supported by US NSF under award # CCF-1320069 (primary source of support), CNS-0953941, CNS-1016994 and CPS-1035845. All opinions expressed are those of the authors and not necessarily of the NSF.