Probabilistic Analysis of Time Sensitive Systems

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Jeffery Hansen
October 28, 2014
<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>28 OCT 2014</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE AND SUBTITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probabilistic Analysis of Time Sensitive Systems</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. AUTHOR(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hansen / Jeffery</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>12. DISTRIBUTION/AVAILABILITY STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved for public release, distribution unlimited.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13. SUPPLEMENTARY NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>The original document contains color images.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>14. ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>15. SUBJECT TERMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16. SECURITY CLASSIFICATION OF:</th>
<th>17. LIMITATION OF ABSTRACT</th>
<th>18. NUMBER OF PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. REPORT</td>
<td>SAR</td>
<td>14</td>
</tr>
<tr>
<td>unclassified</td>
<td>unclassified</td>
<td></td>
</tr>
<tr>
<td>b. ABSTRACT</td>
<td>unclassified</td>
<td></td>
</tr>
<tr>
<td>c. THIS PAGE</td>
<td>unclassified</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>19a. NAME OF RESPONSIBLE PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

Standard Form 298 (Rev. 8-98)  
Prescribed by ANSI Std Z39-18
What is the Problem?

Time-sensitive systems in uncertain environments have complex behaviors. How do we validate correct timing in such systems?

- Exact probabilistic verification is infeasible due to model size
- Black box testing does not yield bounded predictions
- Need formal approach for dealing with uncertainty
  - Accurate, bounded, probabilistic results
  - In reasonable time even for rare outcomes

Use statistical model checking to do a “smart sampling of the world”

- Simulation captures both random variables and timing (scheduling)
- Importance sampling “tilts” input distributions for efficient probability estimation of “rare” events

Note: We use “probability estimation” based statistical model checking. There is also a “hypothesis testing” based version.
Statistical Model Checking (SMC)

Any system $\mathcal{M}$ that takes random inputs

Probabilistic Temporal Logic Formula $\phi$

Statistical Model Checker

Relative Error $= \frac{\text{Std. Dev.}}{\text{Mean}}$

Estimated Probability that $\mathcal{M} \models \phi$ with relative error $RE$

Based on Monte-Carlo Simulation

- System properties described in formal language (UTSL, BLTL, etc.)
- Property is tested on “sample trajectories” (sequence of states)
- Each outcome can be treated as a Bernoulli random variable (i.e., coin flip)
Probability Estimation with SMC

SMC Basics
• Indicator function $I(\vec{x}) = 1$ iff property holds for input $\vec{x}$.
• Relative Error $RE(\hat{p}) = \frac{\sqrt{\text{var}(\hat{p})}}{E[\hat{p}]}$ is measure of accuracy.
• Draw random samples from input distribution $f(\vec{x})$ until target Relative Error is met.
• Estimated probability that property holds is:
  \[
  \hat{p} = \frac{1}{N} \sum_{i=1}^{N} I(\vec{x}_i) = \frac{1}{10} = 0.1
  \]
  \[
  RE(\hat{p}) = \frac{0.32}{0.1} = 3.2
  \]

Importance Sampling
• Modify input distribution to make rare properties more visible.
• Weighting function $W(\vec{x})$ maps solution back to original problem.
• Reduced relative error with same number of samples.
  \[
  \hat{p} = \frac{1}{N} \sum_{i=1}^{N} I(\vec{x}_i)W(\vec{x}_i) = \frac{0.2 + 0.5 + 0.3}{10} = 0.1
  \]
  \[
  RE(\hat{p}) = \frac{0.18}{0.1} = 1.8
  \]
Osmosis SMC Tool

Osmosis is a tool for Statistical Model Checking (SMC) with Semantic Importance Sampling.

- Input model is written in subset of C.
- ASSERT() statements in model indicate conditions that must hold.
- Input probability distributions defined by the user.
- Osmosis returns the probability that at least one of the ASSERT() statements does not hold.
- Uses dReal\(^1\) solver to build \(I^*(\vec{x})\).
- Simulation halt condition based on:
  - Target relative error, or
  - Set number of simulations

\(^1\)http://dreal.cs.cmu.edu/

Osmosis Main Algorithm

1. Generate approximation of fault region

\(I(\vec{x})\): Indicator Function defines fault region
\(I^*(\vec{x})\): Abstract Indicator Function defines over-approximation

2. Conduct SMC and calc. failure prob.

\[ \hat{p}_{\text{raw}} = \frac{5}{20} \quad \text{# in Fault} \]
\[ p^* = \frac{6}{64} \quad \text{Fraction of input } I^*(\vec{x}) \text{ covers} \]
\[ \hat{p} = \hat{p}_{\text{raw}} p^* = 0.23 \]
$I^*(\hat{x})$ Generation Algorithm

Algorithm:

1. Set the current “cube” as the full range of all inputs.
2. Apply dReal to the current cube.
3. If the result is “SAT”, split cube into two equal probability cubes on one variable, and recursively apply at Step 2.
Example: Air Hockey Problem

Air Hockey Problem
• Table with a moving puck and a fixed target.
• Puck rebounds without friction.

Inputs
• Angle – Initial angle at which puck is hit.
• Distance – Total distance of travel for puck.

Failure Condition
• Puck stops on target (red dot).

Challenges
• Multiple failure areas in input space.
• Complex structure of failure area.
Fault Map for Air Hockey Problem

Fault map shows area of input space where faults are located.

- Plotted in CDF space.
- **Green** area indicates input space included in $I^*(x)$.
- **Red** area indicates input space included in $I(x)$.

Recursion Depth: 12
Sample Size vs Recursion Depth (Air Hockey)

Simulation effort with SIS decreases exponentially with recursion depth.
$p^*$ vs Recursion Depth (Air Hockey)

Upper-bound $p^*$ becomes more accurate as recursion depth increases.
Effect of SIS Optimizations

Optimization 1
No call to dReal if first child call is unsat.

Optimization 2
Use counter-example from parent to avoid dReal calls on children.

Optimization 2 results in greatest benefit with factor of two reduction in number of calls to dReal. Small additional benefit by combining both methods.
Conclusion

Semantic Importance Sampling

• Create approximation of fault region using abstraction.
• Create an alternate input distribution for importance sampling.
• Level of approximation (recursion depth) is user tunable.
• Can reduce SMC sample size by orders of magnitude.

Osmosis tool

• Applies semantic importance sampling on a C-like specification.
• Uses the dReal SMT solver to build approximate fault region model.
• Can be applied when there are multiple fault regions.
• Optimization techniques can nearly halve number of dReal tests required.
Contact Information Slide Format

Presenter / Point of Contact
Jeffery Hansen, SSD
Telephone: +1 412-268-9565
Email: jhansen@sei.cmu.edu

U.S. Mail
Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Web
www.sei.cmu.edu
www.sei.cmu.edu/contact.cfm

Customer Relations
Email: info@sei.cmu.edu
Telephone: +1 412-268-5800
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257