COSMOS: a Tool For Statistical Model-Checking

Benoît Barbot, Hilal Djafri
LSV-ENS de Cachan
MeFoSyLoMa Seminar
Cachan, March 23rd 2011
Outline

1. Introduction
2. Model
3. Properties
4. Algorithms
5. Importance Sampling
6. Tools
7. Experiments
8. Future Works
Stochastic Model Checking

Stochastic Model $\mathcal{M}$
CTMC, DTMC, MDP
SPN, SAN, PEPA

Check $\phi$ over $\mathcal{M}$

$\text{Prob}(\mathcal{M} | = \phi)$

$\Delta \triangleleft p(\phi)$

(MeMoSyLoMa)

COSMOS
Cachan, March 23rd 2011
<table>
<thead>
<tr>
<th>Stochastic Model $\mathcal{M}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTMC, DTMC, MDP</td>
</tr>
<tr>
<td>SPN, SAN, PEPA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stochastic Logic $\mathcal{L}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCTL, CSL</td>
</tr>
<tr>
<td>asCSL, CSL\textsuperscript{TA}</td>
</tr>
</tbody>
</table>
Stochastic Model Checking

**Stochastic Model** $\mathcal{M}$
- CTMC, DTMC, MDP
- SPN, SAN, PEPA

**Stochastic Logic** $\mathcal{L}$
- PCTL, CSL
- asCSL, CSL$^{TA}$

Check $\phi$ over $\mathcal{M}$
Stochastic Model Checking

Stochastic Model $\mathcal{M}$
- CTMC, DTMC, MDP
- SPN, SAN, PEPA

Stochastic Logic $\mathcal{L}$
- PCTL, CSL
- asCSL, CSL$^{TA}$

Check $\phi$ over $\mathcal{M}$

$\text{Prob}(\mathcal{M} \models \phi)$
$P_{\phi}(\phi) \bowtie p$
Numerical Methods

- **Principles**
  - Generate a stochastic process from the high level description.
  - Compute some measures from the process: numerical analysis, solving systems of equations.

- **Advantages**
  - Accuracy of results

- **Drawbacks**
  - Require huge memory
  - The stochastic process must be Markovian or more generally semi-regenerative

- **Tools**: PRISM, MRMC, MC4CSTLA
Statistical Methods

- **Principles**
  - Generate sufficient number of trajectories.
  - Discrete event simulation, statistical techniques: confidence interval, hypothesis testing

- **Advantages**
  - No problem of memory
  - General class of stochastic processes

- **Drawbacks**
  - Execution time can be very important.
  - Nested formulas are not considered.
  - Steady state properties are difficult to compute.

- **Tools:** PRISM, MRMC, APMC, VESTA, YMER
Generalized Stochastic Petri Net (GSPN)

- Petri nets
- Distribution of the delay of firing a transition.
- Policies: selection, memory, service.

Shared Memory System
Some GSPN Implementation Details

1 **Arcs**
   - Type: in-arcs, out-arcs, inhibitor-arcs.
   - Valuations: integer, marking dependent.

2 **Transitions**
   - Attributes: distribution, priority and weight.
   - Distribution: *Dirac*, *Geometric*, *Exponential*, *Erlang*, etc.
   - Exponential parameter can be marking dependent.

3 **Policies**
   - Service: single, multiple, infinite.
   - Memory: enabled, age-memory.
Outline

1 Introduction
2 Model
3 Properties
4 Algorithms
5 Importance Sampling
6 Tools
7 Experiments
8 Future Works
A HASL formula has two components:

1. A deterministic hybrid automaton, with a set of variables whose rates are constants or state dependent.

2. An expression on the automaton variables, built with numerical operators and expectation.
\[ \phi = (P_1 + P_2 \geq 4) \]

\[ AVG(\text{Last}(x)) / A = \text{Prob}(\neg \phi U^{[0,8]}(\phi)) \]
\[ \phi = (P_1 + P_2 \geq 4) \]

\[ \begin{align*}
E, t \leq 8 & & \rightarrow & & l_1, \phi \\
\text{\#}, t = 8, x := 0 & & \rightarrow & & l_2, \neg \phi
\end{align*} \]

\[ \text{AVG} \left( \text{Last}(x) \right) / A = \text{Prob} \left( (\neg \phi) U^{[0,8]}(\phi) \right) \]
Main Algorithm

The main algorithm launches the generation of trajectories and compute on the fly the expression and a confidence interval around it.

**Algorithm 1:**

```plaintext
begin
    \( \hat{H} = 0; \ K = 0; \ K_{succ} = 0; \ w = \infty; \ Z = NormalPercentile(1 - \alpha/2) \)
    while \((w > width \ and \ K < maxpaths)\) do
        \( i = 0 \)
        while \((i < batch \ and \ K < maxpaths)\) do
            \( (success, val) = SimulateSinglePath(); \ K = K + 1 \)
            if \((success)\) then
                \( K_{succ} = K_{succ} + 1; \ i = i + 1; \ UpdateStatistics(\hat{H}, val, K_{succ}) \)
            end
        end
        \( UpdateWidth(Var, K_{succ}, Z) \)
    end
    return \( \hat{H} \ and \ CI(\hat{H}) \)
end
```

1 Data to be maintained in memory
   - The current marking of the Petri net.
   - The current location of the automaton.
   - The current value of variables and expression.
   - The list of enabled events.

2 A step of trajectory generation consists of
   - Determine the enabled arc of the automaton.
   - Fire the enabled arc.
   - If the fired arc is a synchronized one, then update the Petri net marking and the events list.
   - The algorithm terminates when:
     - The automaton reaches a final location.
     - No synchronization with net is possible and no autonomous arc is enabled.
Events List

- Data structure: binary min-heap.
- A node $e$ in the heap is tuple of: $(t, pr, w)$
- $e_1(t_1, pr_1, w_1) \prec e_2(t_2, pr_2, w_2)$ if:
  \[
  \begin{cases}
  t_1 < t_2, \\
  or \\
  t_1 = t_2 \ and \ pr_1 > pr_2 \\
  or \\
  t_1 = t_2 \ and \ pr_1 = pr_2 \ and \ w_1 < w_2
  \end{cases}
  \]
- When a Petri net transition $tr$ is fired the heap is updated by examining transitions which may be enabled and those which may be disabled.
Outline

1. Introduction
2. Model
3. Properties
4. Algorithms
5. Importance Sampling
6. Tools
7. Experiments
8. Future Works
Setting

- Given a Markov chain with two absorbing states $s_+$ and $s_-$. 

- **Goal:**
  Computation of the probability to reach the state $s_+$. 

- **Hypothesis:**
  Those states are reached with probability 1.
Inputs

- We want to estimate the probability of reaching $s_+$
- The probability of reaching $s_+$ is about $10^{-15}$.
- The threshold value $10^{-6}$.
- We compute $10^9$ trajectories.
Rare Event Problem

• Inputs
  • We want to estimate the probability of reaching \( s_+ \)
  • The probability of reaching \( s_+ \) is about \( 10^{-15} \).
  • The threshold value \( 10^{-6} \).
  • We compute \( 10^9 \) trajectories.

• Possible outcomes
  • No trajectory reaches \( s_+ \) with probability \( \approx 1 - 10^{-6} \)
    We obtain the following confidence interval: \([0; 7.03 \times 10^{-9}]\)
    \( \Rightarrow \) Confidence interval too large
  • One trajectory reaches \( s_+ \) with probability smaller than \( 10^{-6} \)
    We obtain the following confidence interval: \([6.83 \times 10^{-9}; 1.69 \times 10^{-8}]\)
    \( \Rightarrow \) Value outside the confidence interval
  • More than one trajectory reaches \( s_+ \)
    \( \Rightarrow \) Value outside the confidence interval
Importance Sampling

Principle: Substitute $W_s$ to $V_s$ with same expectancy but reduced variance.

1. Substitute $P'$ to $P$ such that $P(s, s') > 0 \Rightarrow P'(s, s') > 0 \vee s = s_-$

2. For each trajectory $\sigma = s \rightarrow s_1 \rightarrow s_2 \cdots s_k \rightarrow s_\pm$

   We define

   $$W_s = \begin{cases} 
   \frac{P(s, s_1)}{P'(s, s_1)} \cdot \frac{P(s_1, s_2)}{P'(s_1, s_2)} \cdots \frac{P(s_k, s_\pm)}{P'(s_k, s_\pm)} & \text{if } \sigma \text{ ends in state } s_+ \\
   0 & \text{if } \sigma \text{ ends in state } s_- 
   \end{cases}$$

3. Statistically estimate $E(W_{s_0})$
Importance Sampling

Principle: Substitute $W_s$ to $V_s$ with same expectancy but reduced variance.

1. Substitute $P'$ to $P$ such that $P(s, s') > 0 \Rightarrow P'(s, s') > 0 \lor s = s_-$

2. For each trajectory $\sigma = s \rightarrow s_1 \rightarrow s_2 \cdots s_k \rightarrow s_\pm$
   We define
   \[
   W_s = \begin{cases} 
   \frac{P(s, s_1)}{P'(s, s_1)} \cdot \frac{P(s_1, s_2)}{P'(s_1, s_2)} \cdots \frac{P(s_k, s_\pm)}{P'(s_k, s_\pm)} & \text{if } \sigma \text{ ends in state } s_+ \\
   0 & \text{if } \sigma \text{ ends in state } s_-
   \end{cases}
   \]

3. Statistically estimate $E(W_{s_0})$

This method is unbiased

\[
\forall s \in S, \ E(W_s) = E(V_s)
\]

Objective

\[
V(W_{s_0}) \ll V(V_{s_0})
\]
Methodology

1. Specify a reduced model $\mathcal{M}^\bullet$ and a reduction function $f$.

2. Establish using analysis of $\mathcal{M}$ and $\mathcal{M}^\bullet$ that the reduction “guarantees the variance reduction”.

3. Compute with a numerical model checker the probability of for each state of $\mathcal{M}^\bullet$ to reach $s_+$.

4. Compute statistically the probability to reach $s_+$ in $\mathcal{M}$ using the importance sampling induced by $\mathcal{M}^\bullet$. 
Outline

1. Introduction
2. Model
3. Properties
4. Algorithms
5. Importance Sampling
6. Tools
7. Experiments
8. Future Works
- **Programming Language:** C++

- **Interface:**
  - **Inputs:** A GSPN (textual, CosyVerif, GreatSpn GUI), A HASL formula [LHA, EXP] (textual, CosyVerif)
  - **Output:** Evaluation of EXP

- **Technical Details:**
  - Model compilation
  - Random numbers generation: BOOST library
1 Introduction
2 Model
3 Properties
4 Algorithms
5 Importance Sampling
6 Tools
7 Experiments
8 Future Works
Without rare events

<table>
<thead>
<tr>
<th>N</th>
<th>Numerical Prism</th>
<th>Statistical Prism</th>
<th>Cosmos</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T(sec) Mem Value</td>
<td>T(sec) Trajectories Value</td>
<td>T(sec) Trajectories Value</td>
</tr>
<tr>
<td>50</td>
<td>0.2 128Ko 0.35</td>
<td>2.8 2406 0.35</td>
<td>1     2500 0.36</td>
</tr>
<tr>
<td>100</td>
<td>1.2 387Ko 0.34</td>
<td>5.6 2345 0.33</td>
<td>5     2400 0.34</td>
</tr>
<tr>
<td>200</td>
<td>8   1.3Mo 0.34</td>
<td>13 2425 0.35</td>
<td>14    2400 0.34</td>
</tr>
<tr>
<td>500</td>
<td>174 5.7Mo 0.34</td>
<td>44 2370 0.34</td>
<td>50    2500 0.35</td>
</tr>
<tr>
<td>1000</td>
<td>1375 20Mo 0.34</td>
<td>99 2434 0.34</td>
<td>105   2400 0.35</td>
</tr>
</tbody>
</table>

We see that Cosmos and statistical Prism are equivalent w-r-t time computation.
We see that:

- Our rare event method is able to deal with tiny probabilities but not statistical Prism.
- With huge models, in particular on the last line ($N=5000$), the Prism numerical model checker is not able to perform the computation whereas our tool is.
Outline

1. Introduction
2. Model
3. Properties
4. Algorithms
5. Importance Sampling
6. Tools
7. Experiments
8. Future Works
Conclusion and Future Works

- Extension of the model to high-level Petri nets.
- Generalization of the importance sampling method.
- Automation of the construction of the reduced model.
- Tool box for more complex systems.