COSMOS: a Tool For Statistical Model-Checking

Benoît Barbot, Hilal Djafri
LSV-ENS de Cachan

MeFoSyLoMa Seminar

Cachan, March 23^{rd} 2011

Outline

Outline

Stochastic Model Checking

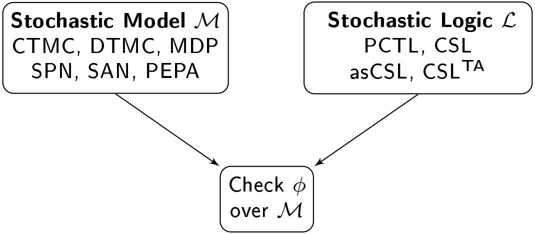
Stochastic Model $\mathcal{M}$
CTMC, DTMC, MDP
SPN, SAN, PEPA

Stochastic Model Checking

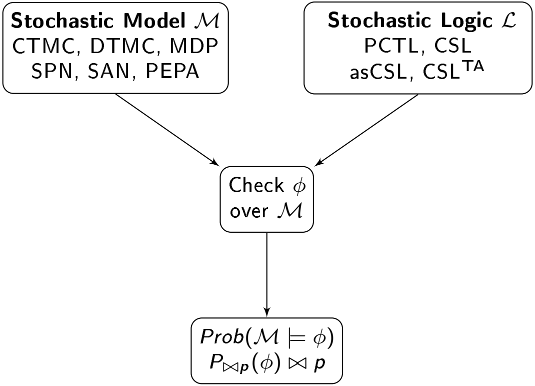
Stochastic Model $\mathcal{M}$
CTMC, DTMC, MDP
SPN, SAN, PEPA

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

Stochastic Model Checking



Stochastic Model Checking



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, MC4CSLTA

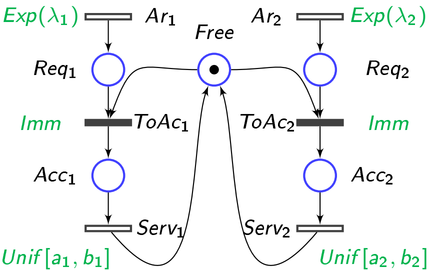
Statistical Methods

- 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

Outline

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

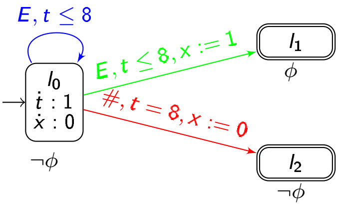
HASL Formula

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.

HASL Formula (2)

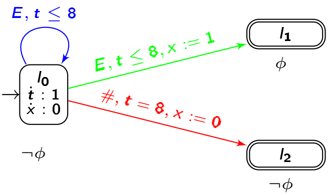
$\phi = (P_1 + P_2 \geq 4)$

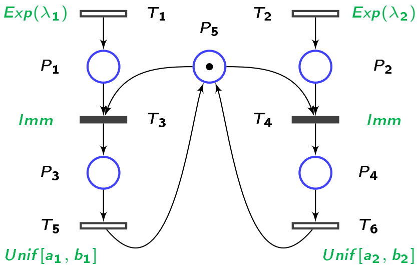


$AVG(Last(x))/\mathcal{A} = Prob((\neg\phi)U^{[0,8]}(\phi))$

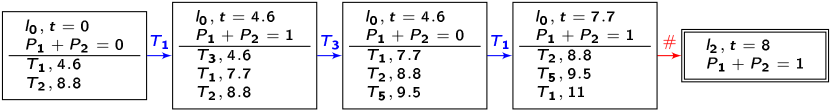
Example

$$\phi = (P_1 + P_2 \geq 4)$$





$$AVG(Last(x))/\mathcal{A} = Prob((\neg\phi)U^{[0,8]}(\phi))$$



Outline

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:

```
begin
    Ĥ = 0;  K = 0;  Ksucc = 0;  w = ∞;  Z = NormalPercentile(1 − α/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
                Ksucc = Ksucc + 1;  i = i + 1;  UpdateStatistics(Ĥ, val, Ksucc)
            end
        end
        UpdateWidth(Var, Ksucc, Z)
    end
    return Ĥ  and  CI(Ĥ)
end
```

Generate A Single Trajectory

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:

$$\begin{cases} t_1 < t_2, \\ or \\ t_1 = t_2 \quad and \quad pr_1 > pr_2 \\ or \\ t_1 = t_2 \quad and \quad pr_1 = pr_2 \quad and \quad w_1 < w_2 \end{cases}$$

- 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

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.

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.

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 $\mathbf{P}'$ to $\mathbf{P}$ such that $\mathbf{P}(s,s') > 0 \Rightarrow \mathbf{P}'(s,s') > 0 \vee s = s_-$
2. For each trajectory $\sigma = s \longrightarrow s_1 \longrightarrow s_2 \cdots s_k \longrightarrow s_\pm$
   We define
   $$W_s = \begin{cases} \frac{\mathbf{P}(s,s_1)}{\mathbf{P}'(s,s_1)} \cdot \frac{\mathbf{P}(s_1,s_2)}{\mathbf{P}'(s_1,s_2)} \cdot \ldots \cdot \frac{\mathbf{P}(s_k,s_+)}{\mathbf{P}'(s_k,s_+)} & \text{if } \sigma \text{ ends in state } s_+ \\ 0 & \text{if } \sigma \text{ ends in state } s_- \end{cases}$$
3. Statistically estimate $\mathbf{E}(W_{s_0})$

Importance Sampling

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

1. Substitute $\mathbf{P}'$ to $\mathbf{P}$ such that $\mathbf{P}(s,s') > 0 \Rightarrow \mathbf{P}'(s,s') > 0 \lor s = s_-$
2. For each trajectory $\sigma = s \longrightarrow s_1 \longrightarrow s_2 \cdots s_k \longrightarrow s_\pm$
   We define

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

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

This method is unbiased

$$\forall s \in S,\ \mathbf{E}(W_s) = \mathbf{E}(V_s)$$

Objective

$$\mathbf{V}(W_{s_0}) \ll \mathbf{V}(V_{s_0})$$

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

- 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

Outline

Without rare events

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

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

With Rare Events

N	Prism num		Prism stat			Cosmos		
	T (s)	Value	T (s)	$\mu(s_0)$	Conf. Int. width	T (s)	$\mu(s_0)$	Conf. Int. width
50	0.3	0.0929	1.45	0.091	0.016	7	0.090	0.017
100	1.6	0.01177	2.7	0.015	0.007	37	0.01156	8.6E-4
500	126	2.06E-12	2.3	0	#	168	2.075E-12	1.72E-13
1000	860	2.87E-25	No path reaches the rare event			376	2.906E-25	2.52E-26

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

Conclusion and Future Works

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