Simulative Model Checking of Steady State and Time-Unbounded Temporal Operators

Christian Rohr

Department of Computer Science
Brandenburg University of Technology Cottbus

June 25, 2012

Outline

Outline

1. Introduction
2. Model checking
3. MARCIE
4. Example
5. Conclusion & Outlook

Introduction

- Stochastic models are used for a long time to model technical systems, and become increasingly popluar in Systems Biology.
- They are compulsory, if the stochastic noise is crucial for the behavioural properties to be investigated.
- Systems are getting more and more complex $\rightarrow$ number of states grows
- Unbounded models $\rightarrow$ number of states is ∞

$\Rightarrow$ simulative approach to handle such models

Definition

A stochastic Petri net $\mathcal{SPN} = (P, T, f, v, s_0)$ is defined as followed

- P a finite, non empty set of places ◯
- T a finite, non empty set of transitions □
- f: $((P \times T) \cup (T \times P)) \to \mathbb{N}_0$ (weighted directed arcs)
- s_0: initial state
- v: $T \to H$ (stochastic firing rate functions) with
  $H := \bigcup_{t \in T} \left\{ h_t | h_t : \mathbb{N}_0^{|\bullet t|} \to \mathbb{R}^+ \right\}$
  $v(t) = h_t$ for all transitions $t \in T$

Semantic: Continuous Time Markov Chain (CTMC)

Definition

CTMC is a 3-tuple $(S, \mathbf{R}, s_0)$ with S denoting the state space of the underlying net and s_0 the initial state.

$$\mathbf{R}: S \times S \rightarrow \mathbb{R}_{\geq 0}$$

$$\mathbf{R}(s, s') = \begin{cases} h_j(s) & \exists\, t_j \in T : s \xrightarrow{t_j} s' \\ 0 & \text{otherwise} . \end{cases}$$

$E(s) = \sum_{s' \in S} \mathbf{R}(s, s')$, exit rate of state s

- The transient probability $\pi(\boldsymbol{\alpha}, s, \tau)$ to be in a certain state s at a certain time point τ starting with a probability distribution $\boldsymbol{\alpha}$.
- Generalized to probability $\pi(\boldsymbol{\alpha}, \tau)$ for all reachable states at time point τ.
- The transient probability for infinite time $\pi(\boldsymbol{\alpha})$ is called the steady state property.

- creates a single finite path through the possibly infinite CTMC
- direct method introduced by Gillespie[1]
- next reaction method by Gibson & Bruck[2]
- The probability that a transition $t_j \in T$ will occur in the infinitesimal time interval $[\tau, \tau + \Delta\tau)$ is given by:

$$P(\tau + \Delta\tau, t_j \mid s) = h_j(s) \cdot e^{-E(s) \cdot \Delta\tau}$$

- So, the enabled transitions in the net compete in a race condition. The fastest one determines the next state and the simulation time elapsed. In the new state, the race condition starts anew.

[1] Gillespie 1977a.
[2] Gibson et al. 2000.

Basically, each variant performs the following steps:

1. Initialise time $t = t_0$ and the system's state X at time t_0.
2. Repeat:
    1. determine time increment $\tau \in \mathbb{R}$
    2. select next reaction type j depending on the current state $X(t)$
    3. perform state transition imposed by reaction of type j and update state vector X
    4. update time $t = t + \tau$.

    until simulation time is reached.

Introduction

Stochastic Simulation

- To achieve an appropriate accuracy of the results, several simulation runs need to be done.
- The method of our choice is the confidence interval[3].
- Assuming an accuracy of the results of 10^{-5} and a confidence level of 95% leads to $\approx 38,000,000$ runs.
- The number of required simulation runs increases exponentially with the accuracy.
- speed up simulation by running several simulations in parallel (multiple threads, multiple processes)

[3] Sandmann et al. 2008.

Outline

Model checking

- A more sophisticated analysis approach.
- automatically determines whether a system satisfies a specific property expressed in some kind of temporal logic
- Probabilistic Linear-time Temporal Logic with numerical constraints — PLTLc

Example

"What is the probability to reach a state satisfying some state property σ?"

- $\mathcal{P}_{=?}[\mathrm{F}(\sigma)]$
- state property σ i.e. $A = B, A + B > C, B \leq 5, \ldots$

Linear-time Temporal Logic (LTL)[4] is the fragment of full Computational Tree Logic (CTL)[5] without path quantifiers, implicitly quantifying universally over all paths.

Definition

$$\begin{aligned}
\phi :=\ & \mathrm{X}^I\,\phi \mid \mathrm{F}^I\,\phi \mid \mathrm{G}^I\,\phi \mid \phi\,\mathrm{U}^I\,\phi \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \sigma \\
& I = \{x \in \mathbb{R}^+ \mid x_1 \leq x \leq x_2\}\,,\ \ omit\ I = [0, \infty) \\
\sigma :=\ & \neg\sigma \mid \sigma \wedge \sigma \mid \sigma \vee \sigma \mid value \trianglelefteq value \mid true \mid false \\
& \trianglelefteq\, \in \{<, \leq, \geq, >, =, \neq\} \\
value :=\ & value \sim value \mid Place \mid Int \mid Real \mid function \\
& \sim\, \in \{+, -, *, /\}
\end{aligned}$$

LTL formulae return true if the trace fulfils the formula, false otherwise.

[4] Pnueli 1981.
[5] Clarke et al. 2001.

Definition

Temporal operators:

- $\mathrm{X}\,\phi$: "next", for at the next time point
- $\phi_1\,\mathrm{U}\,\phi_2$: "until", indicates that a future state where its second argument (ϕ_2) holds is reached while its first argument (ϕ_1) continuously holds
- $\mathrm{F}\,\phi$: "finally", for at some time point in the future
- $\mathrm{G}\,\phi$: "globally", for at all time points in the future

Dualities:

$$\neg\,\mathrm{X}\,\phi \equiv \mathrm{X}\,\neg\phi;$$

$$\mathrm{F}\,\phi \equiv \mathit{true}\,\mathrm{U}\,\phi;\ \mathrm{G}\,\phi \equiv \neg\,\mathrm{F}\,\neg\phi;$$

$$\neg\,\mathrm{F}\,\phi \equiv \mathrm{G}\,\neg\phi;\ \neg\,\mathrm{G}\,\phi \equiv \mathrm{F}\,\neg\phi$$

Model checking

LTL with numerical constraints

Numerical constraints[6] are added to the language by including “free variables" *fVariable* in the definition of *value*.

Definition

$$value := fVariable \mid value \sim value \mid Place \mid Int \mid Real \mid function$$

$$\sim \in \{+, -, *, /\}$$

LTLc formulae return the domain $\mathcal{D}_\phi \subset \mathbb{N}^n$ of the free variables so that ϕ becomes true.

[6] Fages et al. 2007.

Probabilistic LTLc[7] enhances LTLc by the inclusion of a probability operator, and the probabilistic interpretation of the domains for the free variables.

Definition

$$
\begin{aligned}
\psi := \; & \mathcal{P}_{\bowtie x}\,[\phi] \mid \mathcal{P}_{=?}\,[\phi] \\
& \bowtie \in \{<, \leq, \geq, >\}\,, \; x \in [0, 1]
\end{aligned}
$$

PLTLc formulae of type

- $\mathcal{P}_{=?}\,[\phi]$ return the probability that ϕ is true,
- $\mathcal{P}_{\leq 0.5}\,[\phi]$ return true if the probability that ϕ holds is lower or equal to 0.5, false otherwise.

[7] Donaldson et al. 2008.

- When to stop the simulation trace?
  - a fixed, large number of simulation steps
  - a fixed, long end time for simulation trace
  - reaching the steady state is a reasonable stopping criteria
- in steady state numerous properties don't change in time
- for any property p of the system

$$\frac{\partial p}{\partial t} = 0$$

- recently observed behaviour of the system will continue into the future
- the probabilities that various states will be repeated will remain constant

- When to stop the simulation trace?
  - a fixed, large number of simulation steps
  - a fixed, long end time for simulation trace
  - reaching the steady state is a reasonable stopping criteria
- in steady state numerous properties don't change in time
- for any property p of the system

$$\frac{\partial p}{\partial t} = 0$$

- recently observed behaviour of the system will continue into the future
- the probabilities that various states will be repeated will remain constant

- When to stop the simulation trace?
  - ~~a fixed, large number of simulation steps~~
  - a fixed, long end time for simulation trace
  - reaching the steady state is a reasonable stopping criteria
- in steady state numerous properties don't change in time
- for any property p of the system

$$\frac{\partial p}{\partial t} = 0$$

- recently observed behaviour of the system will continue into the future
- the probabilities that various states will be repeated will remain constant

- When to stop the simulation trace?
  - ~~a fixed, large number of simulation steps~~
  - a fixed, long end time for simulation trace
  - reaching the steady state is a reasonable stopping criteria
- in steady state numerous properties don't change in time
- for any property p of the system

$$\frac{\partial p}{\partial t} = 0$$

- recently observed behaviour of the system will continue into the future
- the probabilities that various states will be repeated will remain constant

- When to stop the simulation trace?
  - ~~a fixed, large number of simulation steps~~
  - ~~a fixed, long end time for simulation trace~~
  - reaching the steady state is a reasonable stopping criteria
- in steady state numerous properties don't change in time
- for any property p of the system

$$\frac{\partial p}{\partial t} = 0$$

- recently observed behaviour of the system will continue into the future
- the probabilities that various states will be repeated will remain constant

- When to stop the simulation trace?
  - ~~a fixed, large number of simulation steps~~
  - ~~a fixed, long end time for simulation trace~~
  - reaching the steady state is a reasonable stopping criteria
- in steady state numerous properties don't change in time
- for any property p of the system

$$\frac{\partial p}{\partial t} = 0$$

- recently observed behaviour of the system will continue into the future
- the probabilities that various states will be repeated will remain constant

- When to stop the simulation trace?
  - ~~a fixed, large number of simulation steps~~
  - ~~a fixed, long end time for simulation trace~~
  - reaching the steady state is a reasonable stopping criteria
- in steady state numerous properties don't change in time
- for any property p of the system

$$\frac{\partial p}{\partial t} = 0$$

- recently observed behaviour of the system will continue into the future
- the probabilities that various states will be repeated will remain constant

- When to stop the simulation trace?
  - ~~a fixed, large number of simulation steps~~
  - ~~a fixed, long end time for simulation trace~~
  - reaching the steady state is a reasonable stopping criteria
- in steady state numerous properties don't change in time
- for any property p of the system

$$\frac{\partial p}{\partial t} = 0$$

- recently observed behaviour of the system will continue into the future
- the probabilities that various states will be repeated will remain constant

- When to stop the simulation trace?
  - ~~a fixed, large number of simulation steps~~
  - ~~a fixed, long end time for simulation trace~~
  - reaching the steady state is a reasonable stopping criteria
- in steady state numerous properties don't change in time
- for any property p of the system

$$\frac{\partial p}{\partial t} = 0$$

- recently observed behaviour of the system will continue into the future
- the probabilities that various states will be repeated will remain constant

- When to stop the simulation trace?
  - ~~a fixed, large number of simulation steps~~
  - ~~a fixed, long end time for simulation trace~~
  - reaching the steady state is a reasonable stopping criteria
- in steady state numerous properties don't change in time
- for any property p of the system

$$\frac{\partial p}{\partial t} = 0$$

- recently observed behaviour of the system will continue into the future
- the probabilities that various states will be repeated will remain constant

Model checking
Steady State

Steady state simulation generally poses two problems:

1. The existence of a transient phase may cause the estimate to be biased.
2. The simulation runs are long, and usually one cannot afford to carry out many independent simulations.

Algorithms:

1. batch means method
2. method of independent replicas
3. regeneration method

Skart[8]

- an automated sequential procedure for on-the-fly steady-state simulation output analysis
- exhibit competitive sampling efficiency
- substantially closer conformance to the given CI coverage probabilities than other procedures [9]
- sample batch means algorithm
  - make one long run and partition into *batches*
  - compute an average statistic for each batch
  - construct an interval estimate using the batch means
  - sum up occupation time of fulfilling state
  - $\mathcal{S} = occupation\ time / simulation\ time$

but: several runs needed, because of multiple strongly connected components

[8] Tafazzoli et al. 2010.

[9] ASAP3, WASSP, SBatch, ABATCH, LBATCH

Extension of PLTLc with steady state operator $\mathcal{S}$.

Definition

$$
\begin{aligned}
\psi &:= \mathcal{P}_{\bowtie x}\,[\phi] \mid \mathcal{P}_{=?}\,[\phi] \mid \mathcal{S}_{\bowtie x}\,[\sigma] \mid \mathcal{S}_{=?}\,[\sigma] \\
& \quad \bowtie \in \{<, \leq, \geq, >\}\,,\ x \in [0,1] \\
\phi &:= \mathrm{X}^{I}\,\phi \mid \mathrm{F}^{I}\,\phi \mid \mathrm{G}^{I}\,\phi \mid \phi\,\mathrm{U}^{I}\,\phi \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \sigma \\
& \quad I = \{x \in \mathbb{R}^{+} \mid x_1 \leq x \leq x_2\}\,,\ \mathit{omit}\ I = [0, \infty) \\
\sigma &:= \neg\sigma \mid \sigma \wedge \sigma \mid \sigma \vee \sigma \mid \mathit{value} \trianglelefteq \mathit{value} \mid \mathit{true} \mid \mathit{false} \\
& \quad \trianglelefteq \in \{<, \leq, \geq, >, =, \neq\} \\
\mathit{value} &:= \mathit{value} \sim \mathit{value} \mid \mathit{Place} \mid \mathit{Int} \mid \mathit{Real} \mid \mathit{function} \\
& \quad \sim \in \{+, -, *, /\}
\end{aligned}
$$

Outline

MARCIE

- symbolic and simulative analysis tool for stochastic Petri nets
- efficient Interval Decision Diagram (IDD) implementation
- symbolic state space analysis including efficient saturation-based state space generation
- evaluation of standard Petri net properties as well as CTL model checking
- symbolic Continuous Stochastic Reward Logic (CSRL) model checking

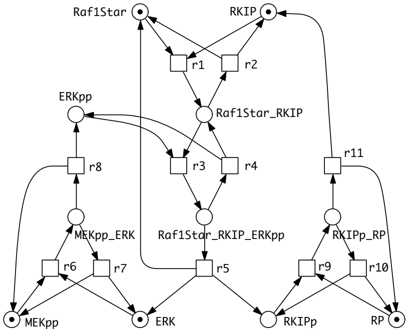
MARCIE

- two exact simulation algorithms
  - direct method[10]
  - next reaction method[11]
- parallelised simulation engine using multiple threads or Message Passing Interface (MPI)
- simulative PLTLc model checking (on-the-fly, offline)
- available at http://www-dssz.informatik.tu-cottbus.de/marcie.html

[10] Gillespie 1977b.

[11] Gibson et al. 2000.

Outline



- non-linear ODE model[12], Petri net model[13]
- stochastic Petri net $\mathcal{SPN}_{ERK}$ with 11 places and 11 transitions connected by 34 arcs
- mass action kinetics with original parameter values

[12] Cho et al. 2003.

[13] Gilbert et al. 2006; Heiner et al. 2010b.

Example

RKIP inhibited ERK pathway

State space

The size of the state space for different initial markings of $\mathcal{SPN}_{ERK}$ computed with MARCIE's symbolic state space generation. All places which carry one token in $\mathcal{SPN}_{ERK}$ have now initially N tokens.

N	|states|	N	|states|	N	|states|	N	|states|
5	1,974	20	1,696,618	40	79,414,335	100	1.591×10^{10}
10	47,047	25	5,723,991	50	2.834×10^{8}	250	3.582×10^{12}
15	368,220	30	15,721,464	60	8.114×10^{8}	500	2.231×10^{14}

Example

RKIP inhibited ERK pathway

Reachability

We first check the reachability of a state at some time in the future, such that the number of tokens on place *MEKpp* is between 60% and 80% of N:

$$\mathcal{P}_{=?}\left[\mathrm{F}\left[MEKpp \geq N \cdot 0.6 \land MEKpp \leq N \cdot 0.8\right]\right].$$

This is the same property as in Heiner et al. 2010a, but without time bounds.

Result

Reachability analysis for different initial markings N of $\mathcal{SPN}_{ERK}$.
The total time is given for different numbers of workers.

N	1	2	4	8	16	32	64	result
20	0m56s	0m28s	0m14s	0m7s	0m3s	0m2s	0m0s	[1,1]
30	1m17s	0m38s	0m19s	0m10s	0m4s	0m3s	0m1s	[1,1]
40	1m38s	0m48s	0m24s	0m12s	0m6s	0m3s	0m1s	[1,1]
50	1m57s	0m59s	0m30s	0m15s	0m7s	0m4s	0m2s	[1,1]
60	2m23s	1m9s	0m35s	0m17s	0m8s	0m5s	0m3s	[1,1]

In any case such a state was reached, therefore the probability of the formula is 1.

Example

RKIP inhibited ERK pathway

Steady State

Since we know now that such a state is eventually reached, we want to compute the steady state probability of being in such a state, where the number of tokens on place *MEKpp* is between 60% and 80% of N:

$$\mathcal{S}_{=?}\left[MEKpp \geq N \cdot 0.6 \wedge MEKpp \leq N \cdot 0.8\right].$$

This is the same property as in Heiner et al. 2010a, so we can verify the correctness of the results.

Example

RKIP inhibited ERK pathway

Result

Steady state analysis for different initial markings N of $\mathcal{SPN}_{ERK}$. The total time is given for different numbers of workers.

N	1	2	4	8	16	32	64	result
20	7m31s	3m46s	1m53s	0m57s	0m27s	0m17s	0m10s	[0.77482, 0.77534]
30	7m34s	3m43s	1m51s	0m57s	0m28s	0m17s	0m11s	[0.83277, 0.83325]
40	7m40s	3m43s	1m56s	0m57s	0m28s	0m18s	0m11s	[0.87416, 0.87470]
50	7m43s	3m57s	1m57s	1m0s	0m30s	0m17s	0m11s	[0.90437, 0.90486]
60	7m53s	3m59s	1m56s	1m1s	0m28s	0m17s	0m12s	[0.92641, 0.92696]

The resulting confidence interval covers the probability computed by the Jacobi method in Heiner et al. 2010a. The algorithm scales nearly linear with the number of worker processes.

Outline

Conclusion

- efficient stochastic simulation (direct, next reaction method)
- simulative PLTLc model checking
- simulative steady state computation
- time-unbounded temporal operators
- on-the-fly and offline
- multi-threaded and distributed computation

Outlook

- extensive comparisons to other simulative/statistical model checking methods/tools
- simulation & model checking at the colored level
- simulative model checking of hybrid Petri nets
- reinvestigate GPGPU for simulative model checking

Thank you for your attention!

http://www-dssz.informatik.tu-cottbus.de/marcie.html