Brandenburgische Technische Universität Cottbus

Simulative Model Checking of Colored Stochastic Petri Nets

Internationale Graduiertenschule der BTU Cottbus
Fachklasse ZUSYS

Motivation

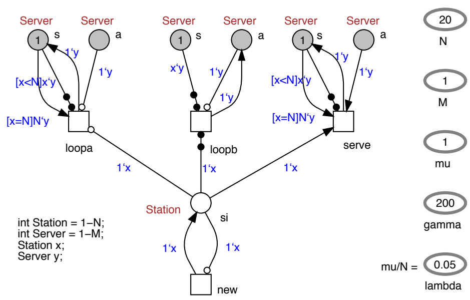
- Stochastic models used to model
  - Technical systems
  - Biochemical networks
- Compulsory, if the stochastic noise is crucial
- Systems are getting more and more complex
  → Number of states grows
- Unbounded models
  → Number of states is ∞
  ⇒ Simulative approach to handle such models

Colored Stochastic Petri Net

- $SPN_C = \langle P, T, F, \Sigma, C, g, f, v, m_0 \rangle$
- P = finite, non-empty set of places ◯
- T = finite, non-empty set of transitions ☐
- F = finite set of directed arcs ⟶
  $F \subseteq (P \times T) \cup (T \times P)$
- Σ = finite, non-empty set of color sets
- $C : P \longrightarrow \Sigma$ assign each place a color set
- $g : T \longrightarrow EXP$ assign each transition a guard expression
- $f : F \longrightarrow EXP$ assign each arc an arc expression
- $v : T \longrightarrow H$ assign a stochastic hazard function to each transition
  $H := \bigcup_{t \in T} \{h_t \mid h_t : \mathbb{N}_0^{|\bullet t|} \rightarrow \mathbb{R}^+\}$
- $m_0 : P \longrightarrow EXP$ assign each place an initialization expression
- EXP = expression over colors $\varsigma \in \Sigma$
- Semantic: Continuous Time Markov Chain

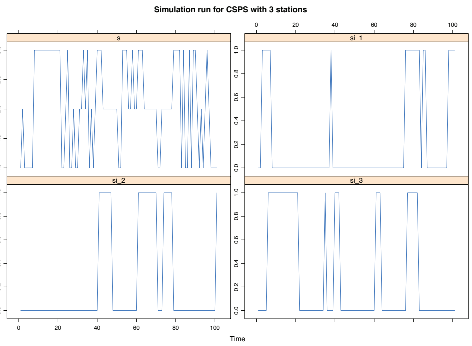
Cyclic Server Polling System

- Cyclic server polling system (CSPS) by Ibe & Trivedi (1990)
- Comprises a set of waiting lines that receive arrivals from the external world
- One server that cyclically visits the queues, providing service to customers
- Use N to denote the number of stations handled by the polling server



Stochastic Simulation

- Creates a single finite path through the possibly infinite CTMC
- Direct method introduced by Gillespie 1977, computational complexity is $\mathcal{O}(T)$
- Next reaction method by Gibson & Bruck 2000, computational complexity is $\mathcal{O}(\log T)$
- Probability that 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.



Model Checking

- 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
- Transient analysis
  - N = 10 stations, time point T = 10
  - Probability that at time point 10 station x is awaiting service
    $$P_{=?}[F_{10}\,[s = \$x \wedge a = 1]]$$
    $$x_1 = 0.0760892732,\ x_2 = 0.0768200225$$
    $$x_3 = 0.0764394239,\ x_4 = 0.0761958408$$
    $$x_5 = 0.0762871845,\ x_6 = 0.0744907591$$
    $$x_7 = 0.0763024084,\ x_8 = 0.0750540450$$
    $$x_9 = 0.0761197211,\ x_{10} = 0.0766677831$$
- Steady state analysis
  - N = 10 stations
  - Probability that in the long run station 1 is awaiting service
    $$S_{=?}[si_1 = 1 \wedge \neg(s = 1 \wedge a = 1)]$$
    $$S = 0.14021$$
  - Probability that in the long run station 1 is idle
    $$S_{=?}[si_1 = 0]$$
    $$S = 0.78162$$

Comparision

- MARCIE – BTU Cottbus
  - http://www-dssz.informatik.tu-cottbus.de
- PRISM – University of Oxford
  - http://www.prismmodelchecker.org
- Total run-time for 66,348,303 simulation runs

Model	N	t	Threads	MARCIE	PRISM
FMS	14	1	1 8	40m43s 5m4s	3h24m13s n.a.
Kanban	10	10	1 8	13m48s 1m58s	2h23m58s n.a.
CSPS	20	1	1 8	4h10m4s 31m53s	62h32m33s n.a.
AKAP	6	1	1 8	2m37s 25s	10m23s n.a.
ANG	6	1	1 8	38m48s 5m20s	8h26m57s n.a.

Conclusions

- 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

- simulation & model checking at the colored level
- numerical PLTLc model checking
- model checking of hybrid Petri nets
- reinvestigate GPGPU for simulative model checking

References

(1) W. Marwan, C. Rohr and M. Heiner: Petri nets in Snoopy: A unifying framework for the graphical display, computational modelling, and simulation of bacterial regulatory networks; In Methods in Molecular Biology – Bacterial Molecular Networks, (Jv. Helden, A. Toussaint and D. Thieffry, Eds.), Humana Press, pages 409–437, 2012.
(2) C. Rohr, W. Marwan and M. Heiner: Snoopy – a unifying Petri net framework to investigate biomolecular networks; Bioinformatics, 26(7), 2010.
(3) M. Heiner, C. Rohr, M. Schwarick and S. Streif: A Comparative Study of Stochastic Analysis Techniques; In Proc. 8th CMSB, Trento, IEEE CS Press, 2010.
(4) M. Schwarick , C. Rohr and M. Heiner: MARCIE – Model checking And Reachability analysis done effiCIEntly; In Proc. 8th QEST, Aachen, ACM Digital Library, 2011.
(5) M. Heiner, M. Herajy, F. Liu, C. Rohr and M. Schwarick: Snoopy - a unifying Petri net tool; In Proc. PETRI NETS 2012, Hamburg, Springer, LNCS, volume 7347, pages 398–407, June 2012.
(6) C. Rohr: Simulative Model Checking of Steady-State and Time-Unbounded Temporal Operators; In Proc. BioPPN 2012, satellite event of Petri Nets 2012, CEUR-WS.org, CEUR Workshop Proceedings, June 2012.

Kontakt
Name: Christian Rohr
Email: Christian.Rohr@tu-cottbus.de