

# GCSRL - A Logic for Stochastic Reward Models with Timed and Untimed Behaviour

Matthias Kuntz

University of Twente, Faculty EEMCS  
<http://wwwhome.cs.utwente.nl/~kuntzwm/>

Boudewijn R. Haverkort

University of Twente, Faculty EEMCS  
<http://dacs.ewi.utwente.nl/staff/brh/>

## Abstract

In this paper we define the logic GCSRL (generalised continuous stochastic reward logic) that provides means to reason about systems that have states which sojourn times are either greater zero, in which case this sojourn time is exponentially distributed (tangible states), or zero (vanishing states). In case of generalised stochastic Petri nets (GSPNs) and stochastic process algebras it turned out that these vanishing states can be very useful when it comes to define system behaviour. In the same way these states are useful for defining system properties using stochastic logics. We extend both the semantic model and the semantics of CSRL such that it allows to attach impulse rewards to transitions emanating from vanishing states. We show by means of a small example how model checking GCSRL formulae works.

## I. INTRODUCTION

Distributed hard- and software systems have become part of our daily life and it becomes more and more important to assert that they are working correctly and that they meet high performance and dependability requirements (performability, cf. [1], [2]). In order to carry out performance and dependability analysis, it is necessary to have both a model and a number of measures of interest, such as utilisation, mean number of jobs, mean time to failure, etc.

In the realm of functional verification, temporal logics such as CTL [3] provide powerful means to specify complex requirements that a system has to satisfy. In the recent years big efforts have been made to provide similar means for the specification of system properties in the area of performance analysis. One result of these efforts is the logic CSL (continuous stochastic logic) [4], [5].

Very recently, the relatively new but established technique of stochastic model checking has been extended to the verification of performability properties. This extension required a new semantic model (Markov Reward models (MRM)), new logics (continuous stochastic reward logic (CSRL)) and new model checking algorithms [6], [7].

In this note, we extend CSRL with means to specify and verify properties of models that contain both timed (Markovian) and untimed (immediate) transitions. Untimed transitions are very useful for modelling synchronisation, decision or cooperation schemes that can be assumed to consume no or only negligible time. As an example, think of an unreliable transmission channel, where the transmission takes measurable time, which is best modelled by a Markovian transition; since the channel is unreliable, a received packet can be either error-free or could have been corrupted during transmission, this is best modelled by two untimed transitions, representing the error-free resp. the error case, which leads to different successive behaviour. It can be useful to enrich untimed transitions with impulse rewards, in connection with an extension of CSRL, one then can reason about, e.g., the probability that the number of corrupted data packet arrivals in a certain time interval is below or above a certain threshold.

For stochastic logics *without* rewards, there are two logics, that allows us to reason about systems with both timed and untimed behaviour. In [8] an extension of the logic CSL is described that allows to specify and verify CSL properties over Markov chains with both timed and untimed transitions. In [9] an extension of the logic SPDL [10], IM-SPDL, having the same aim was described. For stochastic *reward* logics, such an extension has not been proposed so far.

## II. EXTENDED MARKOV REWARD MODELS

The semantic model of the logic GCSRL is an extended Markov reward model (EMRM). An EMRM has two types of transitions, immediate and Markovian transitions. Immediate transitions are untimed transitions, whereas Markovian transitions are associated with an exponentially distributed delay.

*Definition 2.1 (Extended Markov Reward Model):* An extended Markov reward model is an eight-tuple  $\mathcal{M} := (s, S, \text{AP}, L, \rho_Z, \rho_J, R_I, R_M)$ , where:

- $s$  is the unique initial state,
- $S$  is a finite set of states,
- $\text{AP}$  is the set of atomic propositions,
- $L : S \mapsto 2^{\text{AP}}$  is the state labelling function, that associates with every state  $s \in S$  the set of atomic propositions which hold in that state,
- $\rho_Z : S \mapsto \mathbb{R}_{>0}$ , is the state reward function, that associates with every state a reward rate,
- $\rho_J : (R_I \cup R_M) \mapsto \mathbb{R}_{>0}$  is the impulse reward function, that relates to every transition in  $\mathcal{M}$  an impulse reward,
- $R_I : S \times \mathbb{P} \times S$  is the immediate transition relation, where  $\mathbb{P} = (0, 1]$ , If  $(s, p, s') \in R_I$ , we will write  $s \xrightarrow{p, j} s'$ , where  $p \in \mathbb{P}$  is a probability and  $\rho_J((s, p, s')) = j$  is the impulse reward attached to that transition.
- $R_M : S \times \mathbb{R} \times S$  is the Markovian transition relation. If  $(s, \lambda, s') \in R_M$ , we write  $s \xrightarrow{\lambda, j} s'$ , with  $\rho_J((s, \lambda, s')) = j$  the impulse reward attached to that transition.

*Example 2.1:* In Fig. 1 the EMRM for a simple processing unit is given. A job that is processed can be in different phases  $s_1$  to  $s_3$ . The completion of each phase is delayed according to the Markovian transitions with rates  $\lambda_1$  to  $\lambda_3$ . Each phase can be interrupted by a failure of the processing unit, with rates  $\mu_1$  to  $\mu_3$ . The failures can be either disastrous, with probability  $1 - p_1$  or non-disastrous, with probability  $p_1$ . In case of a disastrous failure, (leading to state  $s_{12}$ ), the complete previous work is lost, and the job has to be processed from the beginning. If the error is non-disastrous, (leading to states  $s_7, s_9$ , and  $s_{11}$ ), a rollback to the last error-free state can be made. After the job completes, the processing unit either goes into a stand-by mode (state  $s_5$ ) with probability  $p_2$ , or starts directly with the processing of a new job (with probability  $1 - p_2$ ). The states bear the following atomic propositions:

$$\begin{aligned} L(s_1) &= L(s_2) = L(s_3) = L(s_4) = \text{oper} \\ L(s_4) &= \text{finished} \\ L(s_5) &= \text{standby} \end{aligned}$$

where oper indicates an error-free state of the system, and finished the successful completion of a job. The state reward function  $\rho_Z$  is defined as follows:

$$\rho_Z(s_1) = \dots = \rho_Z(s_3) = 1$$

All other states have state reward zero. The transitions from vanishing states  $s_6, s_8$  resp.  $s_{10}$  to state  $s_{12}$  have impulse reward 1, which is useful when we want to “count” the number of non-correctable errors (cf. Example 3.1).

## III. GCSRL - SYNTAX AND SEMANTICS

In this section we will give the syntax and semantics of GCSRL, mainly in an informal style.

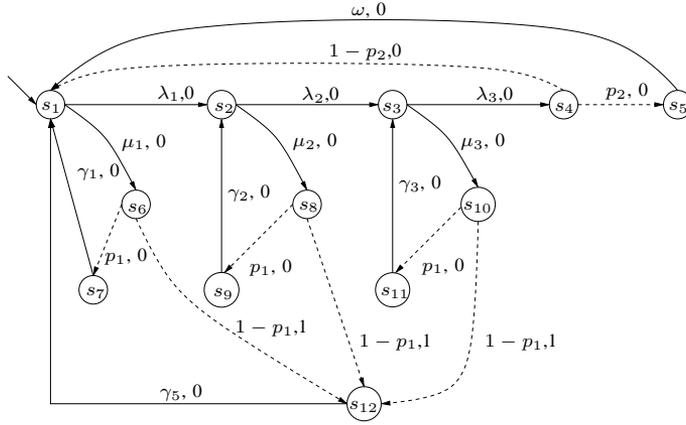


Fig. 1. EMRM for processing unit

### A. Syntax of GCSRL

The syntax of GCSRL formulae is defined by the following grammar, which is slightly different from the syntactic definition in [6], where only state rewards were considered.

*Definition 3.1 (Syntax of GCSRL):* Let  $p \in [0, 1]$  be a probability,  $q \in \text{AP}$  be an atomic proposition, and  $\bowtie \in \{<, \leq, >, \geq\}$  a comparison operator. GCSRL state formulae are then defined as stated below.

$$\Phi := q \mid \neg\Phi \mid \Phi \vee \Psi \mid \mathcal{S}_{\bowtie p}(\Phi) \mid \mathcal{P}_{\bowtie p}(\phi) \mid (\Phi),$$

where  $\phi$  is a GCSRL path formula:

$$\phi := X_{J,Z}^I \Phi \mid \Phi U_{J,Z}^I \Psi,$$

where  $I = [t, t']$  is the real time interval, with  $t \in \mathbb{R}_{\geq 0}$  and  $t' \in \mathbb{R}_{> 0} \cup \{\infty\}$ ,  $J = [j, j']$  is the real impulse reward interval, with  $j \in \mathbb{R}_{\geq 0}$  and  $j' \in \mathbb{R}_{> 0} \cup \{\infty\}$ , and  $Z = [y, y']$  is the real state reward interval, with  $y \in \mathbb{R}_{\geq 0}$  and  $y' \in \mathbb{R}_{> 0} \cup \{\infty\}$ .

### B. Semantics of GCSRL

Except for  $\mathcal{P}_{\bowtie p}(\Phi U_{J,Z}^I \Psi)$ , we will explain the semantics of GCSRL in an informal style. Depending on the lower resp. upper bounds of the intervals  $I$ ,  $J$ , and  $Z$  the semantics of GCSRL path formulae can vary. We will give the semantics of path formulae only for the cases  $I = [0, t]$ ,  $J = [0, j]$ , and  $Z = [0, y]$ . In the sequel, the notion of a *path* is very important.

*Definition 3.2 (Paths in EMRMs):* A path  $\sigma$  of an EMRM  $\mathcal{M}$  is a sequence of transitions of the form

$$s_1 \xrightarrow{t_1, j_1} s_2 \xrightarrow{t_2, j_2} \dots,$$

where  $t_i \in \mathbb{R}_{\geq 0}$  is the real sojourn time in  $s_i$  before passing to  $s_{i+1}$ , and  $j_i$  is the impulse reward gained, when going from  $s_i$  to  $s_{i+1}$ .  $s_i = \sigma[i]$  is the  $(i+1)$ st state of path  $\sigma$ .

1) *Informal GCSRL Semantics:* The meaning of GCSRL formulae can informally be described as follows.

- 1) The semantics of atomic propositions ( $q$ ), negation ( $\neg\Phi$ ), disjunction ( $\Phi \vee \Psi$ ) is defined the usual way [11].
- 2)  $\mathcal{S}_{\bowtie p}(\Phi)$  asserts that the steady-state probability of the set of  $\Phi$ -states, i.e. the probability to reside in a  $\Phi$ -state, once the system has reached stationarity, satisfies the bounds as imposed by  $\bowtie p$ .

- 3)  $\mathcal{P}_{\bowtie p}(\phi)$  asserts that the (transient) probability measure of the paths that satisfy  $\phi$  is within the bounds as given by  $\bowtie p$ .
- 4) A path  $\sigma$  satisfies  $X_{J,Z}^I \Phi$  (“next”), iff  $\sigma[1]$  satisfies  $\Phi$ , the sojourn time in  $\sigma[0]$  does not exceed  $t$  time units, the state reward, accumulated in  $\sigma[0]$  is not greater than  $y$ , and finally the impulse reward, gained when transiting from  $\sigma[0]$  to  $\sigma[1]$  lies within the specified interval  $J$ .
- 5) A path  $\sigma$  satisfies  $\Phi U_{J,Z}^I \Psi$  (“until”), iff within  $t$  time units a state  $\sigma[k]$  is reached that satisfies  $\Psi$ , all preceding states  $\sigma[i]$ ,  $0 \leq i < k$  must satisfy  $\Phi$ , the state reward accumulated in states  $\sigma[i]$  is not above the upper bound of  $Z$ , and the overall impulse reward, gained when taking the path from  $\sigma[0]$  to  $\sigma[k]$  lies within  $J$ .

2) *Formal Semantics for  $\phi = \Phi U_{J,Z}^I \Psi$* : The formal semantics of  $\phi = \Phi U_{J,Z}^I \Psi$  is defined as follows and characterises paths  $\mathcal{M}$  in an EMRM  $\mathcal{M}$  that satisfy  $\phi$ .

*Definition 3.3 (Semantics of  $U_{J,Z}^I$  path formulae)*:

$$\begin{aligned} \sigma \models \Phi U_{J,Z}^I \Psi &\iff \exists k \geq 0 (\sigma[k] \models \Psi \wedge \\ &\forall i < k (\sigma[i] \models \Phi \wedge \sum_{l=0}^{k-1} t_l \leq t \wedge \mathcal{SR}_t \leq y \wedge \mathcal{IR}_t \leq j)), \end{aligned}$$

where  $\mathcal{SR}_t$  is the accumulated state reward up to time  $t$ , and  $\mathcal{IR}_t$  the impulse reward gained up to time  $t$ :

$$\begin{aligned} \mathcal{SR}_t &:= \sum_{l=0}^{k-1} \rho_Z(\sigma[l]) \cdot t_l + (t - \sum_{l=0}^{k-1} t_l) \cdot \rho_Z(\sigma[k]), \\ \mathcal{IR}_t &:= \sum_{l=0}^{k-1} j_l. \end{aligned}$$

*Example 3.1*: Returning to Example 2.1, using GCSRL we can express the following properties the system should satisfy:

- $\Phi_1 := \mathcal{P}_{<0.0001}(\text{oper}U_{[0,2],\emptyset}^{[0,75]} \text{finished})$ : Is the probability that the job finishes within 75 time units smaller than 0.0001, given that at most 2 disastrous failures occurred within the given time interval?
- $\Phi_2 := \mathcal{P}_{\geq 0.75}(\text{oper}U_{[0,0],\emptyset}^{[35,50]} \text{finished})$ : Is the probability that, when the job needs between 35 and 50 time units until completion, no disastrous failure event occurs, at least 75 percent?
- $\Phi_3 := \mathcal{S}_{\geq 0.9999}(\text{oper})$ : In steady-state, is the probability of the system is being operational at least 99.99 percent?
- $\Phi_4 := \mathcal{P}_{>0.9}(\text{true}U_{\emptyset,[0,75]}^{[150,200]} \text{standby})$ : Is the probability to reach the standby-state after at least 150, but at most 200 time units with probability greater 0.9, thereby having accumulated state rewards of at most 75?

## IV. MODEL CHECKING GCSRL

### A. General Idea and Classification of States

In principle, our aim is to reduce model checking GCSRL to model checking CSRL. To do so, we have to transform the EMRM  $\mathcal{M}$  into an MRM  $\mathcal{C}$ . Therefore, we will remove the vanishing states from  $\mathcal{M}$  and adopt the remaining transitions accordingly. It is useful to characterise more precisely the set of states of an EMRM.

*Definition 4.1 (States of an EMRM)*: An EMRM  $\mathcal{M}$  possesses two state classes, vanishing and tangible states. A state is called *vanishing* if it has at least one outgoing untimed transition. A state with only Markovian transitions is called *tangible*.

We will denote the set of vanishing states by  $S_{Van}$  and the set of tangible states by  $S_{Tan}$ . It holds  $S = S_{Van} \cup S_{Tan}$  and  $S_{Van} \cap S_{Tan} = \emptyset$ .

## B. Model Checking $\mathcal{P}_{\bowtie p}(\Phi \cup_{J,Z}^I \Psi)$

Here, we will briefly introduce the basic model checking procedure for GCSRL formulae of the kind  $\mathcal{P}_{\bowtie p}(\Phi \cup_{J,Z}^I \Psi)$ . Following [12], the procedure of transforming an EMRM  $\mathcal{M}$  into an MRM  $\mathcal{C}$  can roughly be described as follows.

- 1) Make  $\neg\Phi$  and  $\Psi$  states in  $\mathcal{M}$  absorbing:  $\mathcal{M}[\neg\Phi \vee \Psi]$  [5].
- 2) Compute  $\mathcal{C}$  from  $\mathcal{M}[\neg\Phi \vee \Psi]$  [12]:
  - a) While  $S_{Van}$  is not empty
    - i) choose a state  $s_v$  from  $S_{Van}$
    - ii) incoming transitions to  $s_v$  have to be redirected to its successor:
      - $s \xrightarrow{\lambda, j_1} s_v \wedge s_v \xrightarrow{p, j_2} s' \Rightarrow s \xrightarrow{p \cdot \lambda, j_1 + j_2} s'$
      - $s \xrightarrow{p_1, j_1} s_v \wedge s_v \xrightarrow{p_2, j_2} s' \Rightarrow s \xrightarrow{p_1 \cdot p_2, j_1 + j_2} s'$

Generally, the algorithm computes the transitive hull over the untimed transitions.

On  $\mathcal{C}$  we then have to compute the transient probability:

$$\mathcal{P}_{\bowtie p}(\Phi \cup_{J,Z}^I \Psi) = \sum_{s' \in S_{Tan}} \mathcal{Y}_{ss'}^{\mathcal{C}}(t, y, j)$$

$\mathcal{Y}_{ss'}^{\mathcal{C}}(t, y, j)$  is the joint probability to be at time instant  $t$  in state  $s'$ , having accumulated state resp. impulse rewards of at most  $y$  resp.  $j$  and having  $s$  as initial state:

$$\mathcal{Y}_{ss'}^{\mathcal{C}}(t, y, j) = Pr(\sigma@t = s', \mathcal{SR}_t \leq y, \mathcal{JR}_t \leq j | \sigma@0 = s),$$

where  $\sigma@t$  resp.  $\sigma@0$  characterise the states of  $\mathcal{M}$  at time instant  $t$  resp. 0. The computation of  $\mathcal{Y}_{ss'}^{\mathcal{C}}(t, y, j)$  is the crucial point of model checking CSRL, to which we have reduced the original model checking problem of GCSRL. In [6] a number of algorithms for the computation of  $\mathcal{Y}_{ss'}^{\mathcal{C}}(t, y, j)$  is described. We will now illustrate the transformation process by means of a small example.

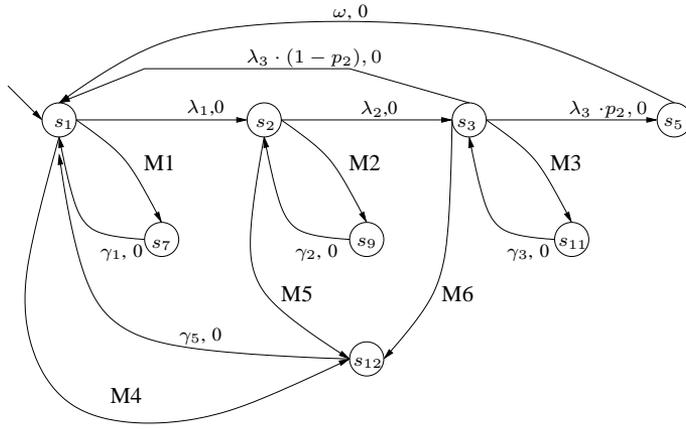
*Example 4.1:* Consider the EMRM from Example 2.1 and take formula  $\Phi_4$  from Example 3.1. In Fig. 2 we find the result of transforming the EMRM from Fig. 1 into an MRM. In the EMRM  $\mathcal{M}$  of Fig. 1 there was a transition  $s_4 \xrightarrow{-p_2} s_5$ . State  $s_4$  is a vanishing state, and has to be deleted, its incoming transitions are redirected to the successor states  $s_5$  resp.  $s_1$ , the rate of the incoming transition of state  $s_4$ ,  $\lambda_3$  is thereby weighted with the appropriate probabilities (cf. Fig. 2):

$$\begin{aligned} s_3 &\xrightarrow{\lambda_3 \cdot p_2} s_5 \\ s_3 &\xrightarrow{\lambda_3 \cdot (1-p_2)} s_1 \end{aligned}$$

Similarly, states  $s_6$ ,  $s_8$ , and  $s_{10}$  are vanishing, and must be deleted. For example, transition  $s_6 \xrightarrow{-p_1, 0} s_7$  is replaced by  $s_1 \xrightarrow{\mu_1 \cdot p_1, 0} s_7$  and  $s_6 \xrightarrow{-1-p_1, 1} s_{12}$  is replaced by  $s_1 \xrightarrow{\mu_1 \cdot (1-p_1), 1} s_{12}$ . For model checking  $\Phi_4$  we only have to make state  $s_5$  absorbing, which is the only state that is assumed to satisfy the atomic property standby (cf. Example 2.1). That means, transition  $s_5 \xrightarrow{\omega} s_1$  has to be deleted (cf. Fig. 3). Formula  $\neg\text{true} = \text{false}$  is not satisfied by any state, therefore, no further state has to be made absorbing. The result of this procedure is the MRM  $\mathcal{C}^{\text{[standby]}}$ . Finally, on  $\mathcal{C}^{\text{[standby]}}$  the reward distribution  $\mathcal{Y}_{ss'}^{\mathcal{C}^{\text{[standby]}}}(t, y, j)$  can be computed, using appropriate CSRL model checking algorithms [6].

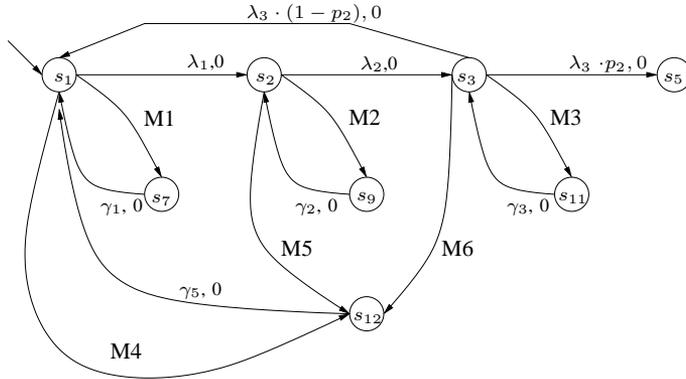
## C. Numerical Results

For all the formulae  $\Phi_1$  to  $\Phi_4$ , from Example 3.1, we will give some numerical results. The results were computed using the tool MRMC [13], using the Tijms-Veldman discretisation algorithm [14]. The experiments were run on a Intel Pentium IV, 3.2 GHz, with 1 GB RAM, running SuSe Linux 10.0 as operating system.



$$\begin{aligned}
 M1 &= \mu_1 \cdot p_1, 0 & M4 &= \mu_1 \cdot (1 - p_1), 1 \\
 M2 &= \mu_2 \cdot p_1, 0 & M5 &= \mu_2 \cdot (1 - p_1), 1 \\
 M3 &= \mu_3 \cdot p_1, 0 & M6 &= \mu_3 \cdot (1 - p_1), 1
 \end{aligned}$$

Fig. 2. MRM for processing unit



$$\begin{aligned}
 M1 &= \mu_1 \cdot p_1, 0 & M4 &= \mu_1 \cdot (1 - p_1), 1 \\
 M2 &= \mu_2 \cdot p_1, 0 & M5 &= \mu_2 \cdot (1 - p_1), 1 \\
 M3 &= \mu_3 \cdot p_1, 0 & M6 &= \mu_3 \cdot (1 - p_1), 1
 \end{aligned}$$

Fig. 3. MRM with  $s_5$  made absorbing for  $\Phi_4$

Transforming the EMRM to an MRM took only negligible time, the bottleneck of the analysis is the computation of  $\mathcal{Y}_{ss'}^c(t, y, j)$ , as can be seen from Table I.

We assume that state  $s_1$  is our initial state, thus, not satisfying  $\Phi_i$  ( $\not\checkmark$ ), resp. satisfying  $\Phi_i$  ( $\checkmark$ ) must be seen with respect to this initial state.

## V. CONCLUSION

In this note we have presented the basic idea of extending the logic CSRL to GCSRL such that we can also reason about reward-based properties of systems that have both timed and untimed behaviour.

Property:	M.C. Time:	Satisfied:
$\Phi_1$	22.67 sec.	✗
$\Phi_2$	14.29 sec.	✗
$\Phi_3$	< 1 msec.	✗
$\Phi_4$	71.85 sec.	✓

TABLE I  
MODEL CHECKING TIMES FOR FORMULAE  $\Phi_1$  TO  $\Phi_4$

Currently, we are defining the semantics of GCSRL path formulae for lower time and reward bounds other than zero. We also plan to define an appropriate notion of bisimulation for GCSRL and check whether the validity of GCSRL formulae is preserved under bisimulation, as it is the case for CSL [5].

Due to the high numerical complexity of computing  $\mathcal{J}_{ss'}^C(t, y, j)$ , we have parallelised some of the algorithms of [6] for running on a traditional cluster system, for first results see [15]. In the future we plan also to do parallelisation on new multi-core processors.

#### REFERENCES

- [1] J. Meyer, "On Evaluating the Performability of Degradable Computing Systems," *IEEE Transactions on Computer Systems*, vol. C-29, no. 8, pp. 720–731, August 1980.
- [2] —, "Performability: A retrospective and some pointers to the future," *Performance Evaluation*, vol. 14, no. 3-4, pp. 139–156, February 1992.
- [3] E. Clarke, E. Emerson, and A. Sistla, "Automatic verification of finite state concurrent systems using temporal logic specifications: A practical approach," in *10th ACM Annual Symp. on Principles of Programming Languages*, 1983, pp. 117–126.
- [4] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton, "Verifying continuous time Markov chains," in *Computer-Aided Verification*, R. Alur and T. Henzinger, Eds., vol. LNCS 1102. Springer, 1996, pp. 146–162.
- [5] C. Baier, B. Haverkort, H. Hermans, and J. Katoen, "Model-Checking Algorithms for Continuous-Time Markov Chains," *IEEE Trans. Software Eng.*, vol. 29, no. 7, pp. 1–18, July 2003.
- [6] L. Cloth, "Model Checking Algorithms for Markov Reward Models," Ph.D. dissertation, University of Twente, Enschede, Netherlands, 2006.
- [7] C. Baier, B. Haverkort, J.-P. Katoen, and H. Hermans, "Model Checking Continuous-Time Markov Chains by Transient Analysis," in *Computer Aided Verification*, E. Emerson and A. Sistla, Eds., vol. LNCS 1855. Springer, 2000, pp. 358–372.
- [8] D. Cerotti, S. Donatelli, A. Horvath, and J. Sproston, "CSL Model Checking for Generalized Stochastic Petri Nets," in *QEST '06: Proceedings of the 3rd international conference on the Quantitative Evaluation of Systems*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 199–210.
- [9] M. Kuntz and M. Siegle, "Symbolic Model Checking of Stochastic Systems: Theory and Implementation," in *13th International SPIN Workshop*. Springer, LNCS 3925, 2006, pp. 89–107.
- [10] M. Kuntz, "Symbolic Semantics and Verification of Stochastic Process Algebras," Ph.D. dissertation, Universität Erlangen-Nürnberg, Institut für Informatik 7, 2006.
- [11] M. Manzano, *Model Theory*. Oxford University Press, 1999.
- [12] M. Siegle, *Behaviour analysis of communication systems: Compositional modelling, compact representation and analysis of performability properties*. Aachen: Shaker Verlag, 2002.
- [13] J.-P. Katoen, M. Khattri, and I. S. Zapreev, "A Markov reward model checker," in *Quantitative Evaluation of Systems (QEST)*. Los Alamos, CA, USA: IEEE Computer Society, 2005, pp. 243–244.
- [14] H. Tijms and R. Veldman, "A Fast Algorithm for the Transient Reward Distribution in Continuous-time Markov Chains," *Operations Research Letters*, vol. 26, no. 4, pp. 155–158, 2000.
- [15] B. Haverkort and M. Kuntz, "Parallel CSRL Model Checking: First Results and Pointers to Future Research," in *6th International Workshop on Parallel and Distributed Methods in verification, PDMC 2007*. to appear, 2007.