4-3-2007

Symbolic Analysis of GSMP Models With One Stateful Clock

Mikhail Bernadsky
University of Pennsylvania

Rajeev Alur
University of Pennsylvania, alur@cis.upenn.edu

Recommended Citation

Mikhail Bernadsky and Rajeev Alur, "Symbolic Analysis of GSMP Models With One Stateful Clock", *Lecture Notes in Computer Science: Hybrid Systems: Computation and Control* 4416, 90-103. April 2007. http://dx.doi.org/10.1007/978-3-540-71493-4_10

From the 10th International Workshop, HSCC 2007, Pisa, Italy, April 3-5, 2007.

Symbolic Analysis of GSMP Models With One Stateful Clock

Abstract

We consider the problem of verifying reachability properties of stochastic real-time systems modeled as generalized semi-Markov processes (GSMPs). The standard simulation-based techniques for GSMPs are not adequate for solving verification problems, and existing symbolic techniques either require memoryless distributions for firing times, or approximate the problem using discrete time or bounded horizon. In this paper, we present a symbolic solution for the case where firing times are random variables over a rich class of distributions, but only one event is allowed to retain its firing time when a discrete change occurs. The solution allows us to compute the probability that such a GSMP satisfies a property of the form "can the system reach a target, while staying within a set of safe states". We report on illustrative examples and their analysis using our procedure.

Disciplines

Computer Sciences

Comments

From the 10th International Workshop, HSCC 2007, Pisa, Italy, April 3-5, 2007.

Symbolic Analysis for GSMP Models with One Stateful Clock[⋆]

Mikhail Bernadsky and Rajeev Alur

Department of Computer and Information Science
University of Pennsylvania
{mbernads, alur}@cis.upenn.edu

Abstract. We consider the problem of verifying reachability properties of stochastic real-time systems modeled as generalized semi-Markov processes (GSMPs). The standard simulation-based techniques for GSMPs are not adequate for solving verification problems, and existing symbolic techniques either require memoryless distributions for firing times, or approximate the problem using discrete time or bounded horizon. In this paper, we present a symbolic solution for the case where firing times are random variables over a rich class of distributions, but only one event is allowed to retain its firing time when a discrete change occurs. The solution allows us to compute the probability that such a GSMP satisfies a property of the form "can the system reach a target, while staying within a set of safe states". We report on illustrative examples and their analysis using our procedure.

1 Introduction

Engineering of complex systems such as hardware devices, communication protocols, multimedia systems and networks requires accurate reliability modeling and performance evaluation at many stages of development [8, 10]. For such systems, it is often the case that the event occurrence times, which determine the evolution of a system, interactions between components and between a system and its environment can be described by probabilistic assumptions. This observation has led to extensive research on *probabilistic model checking* of probabilistic and stochastic models. The goal of probabilistic model checking is to check algorithmically that a model of a system satisfies a probabilistic correctness property, for example, "every message is delivered within 1ms with probability 0.9."

Recently, results were obtained on model checking of discrete and continuous time Markov chains (DTMCs and CTMCs), with specifications written in temporal logics such as PCTL and CSL [5, 11, 12]. While CTMCs can be used as building blocks to approximate distributions with unbounded support [7], approximation of distributions whose support is bounded, for instance, uniform or beta distributions, is problematic. It may be a serious restriction in modeling of real-time systems with mutually exclusive events. To circumvent this

[⋆] This research was supported by the US National Science Foundation via grants CCR-0410662 and CSR-EHS 0509143.

restriction non-Markovian formalisms were proposed, but they either require that non-exponential distributions are deterministic [13], or that at any given moment there is at most one active event with a general distribution [6].

Our goal is to develop algorithms for the probabilistic model checking problem for systems modeled as *Generalized Semi-Markov Processes* (GSMPs) [8, 9, 15]. In our model of *finite-state* GSMPs, the system can be in one of the finitely many states, and can have a finite number of scheduled events. When the event with the least remaining firing time happens, the state is updated probabilistically, and new events can be scheduled at times chosen randomly according to the specified distributions. In [2], the authors show how to check *qualitative* probabilistic properties, that is, whether a GSMP satisfies a property with probability 0 or 1, and this analysis is based on the so-called region graph introduced for analysis of non-probabilistic real-time systems modeled using timed automata [3]. In a recent paper, we showed that if we are given a bound on the number of events, then exact symbolic analysis for verifying quantitative probabilistic properties of GSMPs is possible [1]. None of these techniques, however, suggest a general method for symbolic analysis of GSMPs.

In this paper, we present a symbolic analysis technique for the class of GSMPs where firing times are random variables over a rich class of distributions, but only one event is allowed to retain its firing time when a discrete change occurs. We call this class of processes 1GSMPs. In particular, we focus on model checking of *until* properties: given a set of destination and safe locations, we wish to compute the probability that an execution of the 1GSMP will reach a destination location while staying within the set of safe locations.

In our solution, we first derive a system of integral equations of harmonic functions such that each function is associated with a region of clock values in a particular location, and gives the probability of satisfying the until property as a function of the firing time of the stateful clock upon entry. This step can be easily generalized to work for all GSMPs.

In [14], the integral equations of similar structure were proposed for a related problem for semi-Markov processes. The authors cited [8], noting that solving these equations either by using numerical methods for integral equations or by applying Laplace transforms is not practical and works only for small models. In this paper, we describe a novel method that directly transforms the system of the integral equations into a system of ordinary differential equations. Each integral term in an integral equation is converted into a sum of differentials of newly introduced unknown functions (we call such functions 'auxiliary') and GSMP density functions. The original unknown functions and the auxiliary functions are linked by differential equations. The algorithm that constructs such equations uses the characterization of the density functions as solutions of linear homogeneous ordinary differential equations. The resulting system can then be solved to compute the desired probability for any given initial state.

We illustrate the proposed modeling and analysis techniques using classical examples of component failures and of a $GI/G/1$ queue [4].

2 Generalized Semi-Markov Processes

Let $\mathbb{N}$ be the set of all natural numbers, $\mathbb{N}_0$ be $\mathbb{N} \cup \{0\}$, $\mathbb{R}$ be the set of reals, and $\mathbb{R}_+$ be the set of all non-negative reals.

We start by reviewing some facts from the theory of differential equations and the probability theory. The solutions to the class of differential equations that we will describe form a class of expressions that we will use later to define a class of density functions, that, in turn, will be used in the definition of GSMPs.

2.1 Preliminaries

Linear homogeneous ordinary differential equation with constant coefficients (LHODE) of order n is an equation of the form

$$a_n \frac{\mathrm{d}^n}{\mathrm{d}x^n} y(x) + a_{n-1} \frac{\mathrm{d}^{n-1}}{\mathrm{d}x^{n-1}} y(x) + \cdots + a_1 \frac{\mathrm{d}}{\mathrm{d}x} y(x) + a_0 y(x) = 0,$$

where $a_0, \ldots, a_n \in \mathbb{R}$.

Characteristic equation for this LHODE is $a_n\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0 = 0$. This equation has exactly n (complex, possibly repeated) roots and they determine, up to constants, all solutions of the LHODE.

Specifically, if there are $\lambda_1 = \cdots = \lambda_k$, $k \geq 1$ real repeated roots, then LHODE has a solution $y(x) = e^{\lambda_1 x}(c_1 + \cdots + c_k x^{k-1})$, where $c_1, \ldots, c_k \in \mathbb{R}$ are arbitrary constants. If $\lambda_1 = \cdots = \lambda_k$ are complex repeated roots equal to $\alpha + \beta i$, $\alpha, \beta \in \mathbb{R}$ and $\beta \neq 0$, then the equation should also have k repeated conjugate roots $\bar{\lambda}_1 = \cdots = \bar{\lambda}_k$ equal to $\alpha - \beta i$. All these $2k$ roots correspond to a solution $y(x) = e^{\alpha x}(c_1 + \cdots + c_k x^{k-1}) \sin \beta x + e^{\alpha x}(d_1 + \cdots + d_k x^{k-1}) \cos \beta x$, where $c_1, \ldots, c_k, d_1, \ldots, d_k \in \mathbb{R}$ are arbitrary constants. Summing solutions for such groups of roots we obtain the general solution for the LHODE.

We say that $expr(x)$ is a *DESOL expression* iff it is a sum of terms, such that each term is either of the form $cx^m e^{\mu x} \sin(\alpha x)$ or of the form $cx^m e^{\mu x} \cos(\alpha x)$, where $c, \mu, \alpha \in \mathbb{R}$ and $m \in \mathbb{N}_0$. For every DESOL expression it is possible to construct an LHODE that has this expression as its solution. This 'encoding' of the DESOL expressions with LHODEs will be used later in Section 4 to convert a system of integral equations into a system of differential equations.

Let $Expr(x)$ be the set of all DESOL expressions. Consider a partition $R_a = \cup_{i=1}^{a} \{(i-1, i]\}$ of $(0, a]$, which consists of a unit intervals. The constant a is the *width* of R_a. Let $Int(x)$ be a function defined for all $x \in (0, a]$, such that if $x \in (i-1, i]$, then $Int(x) = i$. We say that a function $f(x)$ is *piecewise DESOL function*, with finite support on R_a, if there exists a map $M_f \colon \{1, \ldots, a\} \to Expr(x)$, such that for all $x \in (0, a]$, $f(x) = M_f(Int(x))(x')$, where $x' = x - Int(x) + 1$. Thus to compute $f(x)$, we determine the interval of R_a to which x belongs, find DESOL expression for this interval and then evaluate this expression at x'. Notice, that $x' \in (0, 1]$, so every expression is evaluated only in that interval. This leads to simplifications in our algorithm. By $f^j(x)$, $1 \leq j \leq a$ we will denote the expression of $f(x)$ that corresponds to the interval $(j-1, j]$.

In a GSMP the time between scheduling an event and its occurrence (or firing time) is modeled as a positive random variable. A random variable X is characterized by its *cumulative distribution function* (cdf) $distr(x) = \Pr(X < x)$, and if $distr(x)$ is continuous then also by *probability density function* (pdf) $dens(x)$ defined by the equation $distr(x) = \int_0^x dens(y)\, dy$.

A unidimensional random variable X has a *DESOL distribution* of width a, if there exists a piecewise DESOL function $dens(x) \geq 0$ on R_a, such that for all $t \in \mathbb{R}_+$, $\Pr(X < t) = \int_0^t dens(y)\, \mathrm{d}y$ [1].

2.2 Modeling Stochastic Processes

A finite-state generalized semi-Markov process (GSMP) with the firing distributions of width a is a tuple $G = (Q, \Sigma, E, init, distr, next)$ where:

- Q is a finite set of locations;
- Σ is a finite set of events;
- $E\colon Q \to 2^{\Sigma}$ assigns to each location $q \in Q$ a set of events that are *active* in q. A location q is *absorbing* iff $E(q) = \emptyset$;
- $init\colon Q \to [0, 1]$ is a probability measure on Q, which for each location $q \in Q$ gives the probability that q is the initial location of G;
- $distr\colon \Sigma \to (\mathbb{R}_+ \to [0, 1])$ assigns to each event its *firing time distribution*, which is a DESOL distribution of width a. For a cdf $distr(e)$, $dens_e$ denotes the corresponding pdf.
- $next\colon Q \times \Sigma \to 2^{\Sigma} \times (Q \to [0, 1])$ defines transitions between the locations of G. This function takes as its arguments a source location q and an active event e of q, and returns a set of events $E^{q,e}_{\text{reset}}$ and a probability measure $P^{q,e}_{\text{next}}$ on Q. For each location q', $P^{q,e}_{\text{next}}(q')$ gives the probability that a run of G will move from q to q' if e fires, and $E^{q,e}_{\text{reset}}$ is the set of events that are reset when the transition occurs. We require that $\sum_{q' \in Q} P^{q,e}_{\text{next}}(q') = 1$, $E^{q,e}_{\text{reset}} \subseteq E(q)$, and $E^{q,e}_{\text{reset}} \subseteq E(q'')$, for every location q'' which can be reached with strictly positive probability[2].

Notice that since we use random variables with density functions that do not have mass points and are discontinuous only at a finite number of points, we do not consider the possibility that several events would fire at the same time.

It is convenient to think that a clock is assigned to each event e. The clock, denoted t_e, shows the time remaining until the next occurrence of e. Upon scheduling/resetting of e we update its clock to a new value chosen independently at random according to $distr(e)$. All clocks of the current active events run down with the same rate equal to 1.

Let us say that $\nu\colon \Sigma \to \mathbb{R}_+$ is a clock valuation (or simply valuation) if ν maps events to the values of their clocks. If an event is not active in the current location we assume that its value is undefined.

[1] Assume that $dens(y)$ is 0 for $y \in (a, +\infty)$.

[2] Adding a possibility that some events can be reset in a transition from one location to another does not make our model more powerful, but it is useful for modeling, and we add it as a “syntactic sugar”.

A *configuration* of the GSMP G is a pair $s = (q, \nu)$, where $q \in Q$ and ν is a clock valuation. Given a configuration $s = (q, \nu)$, let $t^*(s) = \min\{\nu(e), e \in E(q)\}$ be the time until the next transition, and $e^*(s) = \arg\min(\nu(e), e \in E(q))$ be the event that causes the transition. For any $t \leq t^*(s)$ we denote by $\nu - t$ the valuation ν' such that for all $e \in E(q)$, $\nu'(e) = \nu(e) - t$. We say that $s \xrightarrow{t} s'$ is a *timed transition* between the configurations $s = (q, \nu)$ and $s' = (q, \nu')$ if $\nu' = \nu - t$. If $t^*(s) = 0$, and e^* is such that $\nu(e^*) = 0$, then $s \xrightarrow{\mu} s'$ denotes a *discrete transition* between configurations $s = (q, \nu)$ and $s' = (q', \nu')$, where q' is chosen according to the probability measure $\mu = P^{q,e^*}_{\text{next}}$, and the valuation ν' is constructed as follows:

- if an event $e \in E_{\text{inherited}}(q, e^*, q')$, where $E_{\text{inherited}}(q, e^*, q') = E(q') \cap [E(q) \setminus \{e^*\} \setminus E_{\text{reset}}]$ is the set of events that were active in q and continue to be active in q', excluding e^* and excluding the events that were reset, then $\nu'(e) = \nu(e)$;
- if $e \in E_{\text{new}}(q, e^*, q')$, where $E_{\text{new}}(q, e^*, q') = E(q') \setminus E_{\text{inherited}}(q, e^*, q')$, then valuations $\nu'(e)$ are chosen independently at random according to *distr*(e) (i.e. the events in $E_{\text{new}}(q, E^*, q')$ are (re-)scheduled);
- if $e \in E_{\text{cancelled}}(q, e^*, q')$, where $E_{\text{cancelled}}(q, e^*, q') = E(q) \setminus E(q')$ is the set of canceled events that were active in q but no longer active in q', then $\nu'(e)$ is undefined.

A *run* σ of G is a sequence of alternating timed and discrete transitions:

$$\sigma = s_0 \xrightarrow{t^*(s_0)} s'_0 \xrightarrow{\mu_0} s_1 \xrightarrow{t^*(s_1)} s'_1 \xrightarrow{\mu_1} s_2 \xrightarrow{t^*(s_2)} s'_2 \xrightarrow{\mu_2} \dots$$

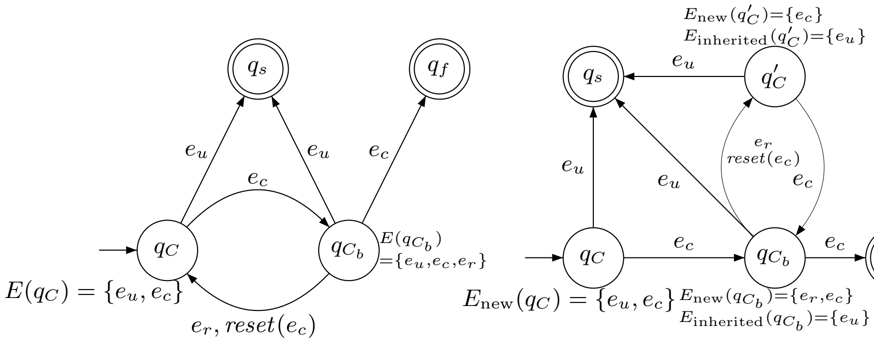
The run σ starts at the initial configuration $s_0 = (q_0, \nu_0)$, q_0 is the initial location, which is chosen according to *init*, and ν_0 is the initial valuation of the events in $E(q_0)$, scheduled according to the corresponding firing time distributions. A run can have a finite or infinite number of transitions; a run that has reached an absorbing location will stay in that location forever.

We say that a GSMP is *normalized* iff (i) for every q_{pred}, e^* and q, such that $P^{q_{\text{pred}},e^*}_{\text{next}}(q) > 0$, $E_{\text{inherited}}$ and E_{new} do not depend on q_{pred} and e^*, i.e. $E_{\text{inherited}}(q_{\text{pred}}, e^*, q) = E_{\text{inherited}}(q)$ and $E_{\text{new}}(q_{\text{pred}}, e^*, q) = E_{\text{new}}(q)$. And (ii) if q is an initial location, i.e. $init(q) > 0$, then $E_{\text{inherited}}(q_{\text{pred}}, e^*, q) = \emptyset$.

Condition (i) states that partition of a location's events into sets of inherited and new events is the same for every run that visits that location. Condition (ii) requires that the set of inherited events for every initial location should be empty.

We modify the definition of GSMP, and say that normalized GSMP is a tuple $G_1 = (Q, \Sigma, E_{\text{inherited}}, E_{\text{new}}, init, distr, next)$ to emphasize that partition into sets of inherited and new events is fixed for every location.

Given a GSMP G, we can, for every location, determine all possible event partitions and then by cloning that location for every found partition and connecting it to the corresponding (clones of) predecessor and successor locations, create a normalized GSMP G_1.



(a) GSMP G

(b) GSMP G_1

Fig. 1. Sample GSMP and its normalized version

G and G_1 are equivalent in the sense that the answers to the questions we are interested in this paper are the same for G and G_1.

A normalized GSMP is called *1GSMP* if for every location q, $E_{\text{inherited}}(q)$ consists of at most one event.

In 1GSMP at most one event can retain the value of its clock upon a transition. The other active events are either reset or canceled in the transition target location. In the sequel, we will be interested only in 1GSMPs.

We say that a clock is *new* if it is the clock of a new event, and it is *inherited* otherwise. We know the distributions of values of all new clocks upon reaching the current location q, but the distribution of the inherited clock is unknown and depends on the path to q.

A *history* π of length n of a run σ is a sequence of locations and transitions between them marked by the events that have fired:

$$q_0 \xrightarrow{e_1^*} q_1 \xrightarrow{e_2^*} \dots \xrightarrow{e_n^*} q_n.$$

2.3 Computing Probabilities of Until Properties

Suppose that we are given a 1GSMP G with firing time distributions of size a. The locations of G are partitioned into three disjoint sets: Q_s, Q_u and Q_d, which are called the set of *safe locations*, the set of *unsafe locations*, and the set of *destination locations*, respectively. We require that from every $q_s \in Q_s$, a location in $Q_u \cup Q_d$ is reachable with probability one. This property can checked by a method presented in [2].

Let $\mathit{\Pi}^n_{\text{until}} \subseteq \mathit{\Pi}$ be a set of histories of length less than or equal to n, such that only the last location of every history in $\mathit{\Pi}^n_{\text{until}}$ is in Q_u, while the other locations are in Q_s. Let $\mathit{\Pi}_{\text{until}} = \cup_{n \geq 0} \mathit{\Pi}^n_{\text{until}}$.

We consider the following *model-checking problem*:

- *What is the probability* P_{until} *that a run* σ *of* G *has a history* $\pi \in \mathit{\Pi}_{\text{until}}$*?*

In addition, our approach will enable us to determine the probability of reaching an unsafe location before any of the destination locations from any location q, if we specify the clock valuation of the event in $E_{\text{inherited}}(q)$.

2.4 Illustrative Example

We illustrate the given definitions by an example that we will use throughout the paper. Consider a system that crucially depends on a component C. To ensure an uninterrupted service, the system has a back-up component C_b. While C is active, C_b is in standby mode, but if C fails, C_b is activated immediately. We are interested to know the probability of the system failure.

The system can be modeled as a GSMP G depicted in Figure 1(a). Each location is marked with the active events. Each transition is marked with the event that causes this transition, and $\mathit{reset}(e)$ indicates that event e was reset.

Location q_C is the initial location, and it models the configuration in which C is operating and C_b is in standby mode. Location q_{C_b} represents the configuration when C has failed and C_b is active. There are two absorbing locations q_s and q_f, the former is the destination location, it models successful completion of the task, the latter is the unsafe location of G and it models the state of the system when both components have failed.

Two events e_u and e_c are active in q_C. The event e_u, which is scheduled only upon the first visit to q_C, models the time interval the system should be up. Firing of this event indicates that the system has completed its task successfully and has reached q_s; e_u is scheduled using a random variable whose density function is $\mathit{dens}_{e_u}(x) = \frac{1}{1-e^{-1}} e^{-x}$ on $(0, 1]$ and 0 otherwise[3]. On $(0, 1]$ this density function is a solution of the differential equation $\frac{\mathrm{d}}{\mathrm{d}x} \mathit{dens}_{e_u}(x) + \mathit{dens}_{e_u}(x) = 0$. The second active event of q_C is e_c, it is scheduled every time a run reaches q_C, and it models a crash event of C. If it fires a run of G moves to q_{C_b}. The event e_c is scheduled using a random variable with the beta density function $\mathit{dens}_{e_c}(x) = \frac{1}{2}x$ on $(0, 1]$, LHODE for $\mathit{dens}_{e_c}(x)$ is $\frac{\mathrm{d}^2}{\mathrm{d}x^2} \mathit{dens}_{e_c}(x) = 0$.

Every time location q_{C_b} is reached, two events e_r and e_c are scheduled. The firing time of e_r is determined by a random variable with the uniform density function $\mathit{dens}_{e_r}(x) = 1$ on $(0, 1]$ and 0 otherwise (its LHODE is $\frac{\mathrm{d}}{\mathrm{d}x} \mathit{dens}_{e_r}(x) = 0$), and firing of this event indicates that C was replaced and the run returns to q_C. But if in q_{C_b} the event e_c fires before e_r, it means that C_b had failed before C was replaced, the system failed and the run moves to the location q_f.

[3] The densities in this example are chosen to make clear our approach and not to model accurately a real system.

G is not a normalized GSMP — upon the first visit to q_C both e_u and e_c are scheduled, but upon every subsequent visit e_c is reset, but the clock of e_u keeps its value. The normalized GSMP G_1 constructed from G is shown in figure 1(b). Location q_C of G was split into two locations q_C and q'_C, and, as in G, q_C is the only initial location of 1GSMP G_1.

3 System of Integral Equations for Harmonic Functions

To solve the model-checking problem we will use a method similar to the "first step analysis" for the Markov chains. For every location we introduce a set of harmonic functions. If a location q does not have an inherited clock, then its set consists of one constant function that we denote by H_q, otherwise it consists of a functions $H_q^1(t), \ldots, H_q^a(t)$.

Each $H_q^i(t)$ is defined on $(0, 1]$. The interpretation of the functions is the following — if location q was reached at the moment when the value of the inherited clock t satisfies $i - 1 < t \leq i$, then the probability to reach an unsafe location before any of the destination locations is $H_q^i(t - (i - 1))$ [4]. If there is no inherited clock in q, then all clocks are rescheduled (or reset) upon reaching q and the constant H_q is the desired probability.

Our method constructs integral equations that express dependence between harmonic functions of a location and all its immediate successor locations. This is a general method and it works not only for 1GSMPs but also for all normalized GSMPs, however restriction to 1GSMPs allows us a transformation from the system of constructed integral equations into a system of ordinary differential equations.

Before describing our algorithm we need to introduce a class of partitions of clock valuations that we will use. The same class of partitions was used in [1] for solving bounded model-checking problem.

3.1 Diagonal Mesh Partitions

For a set of variables $t_1, \ldots, t_n$, an n-dimensional *diagonal mesh partition* $R_a(t_1, \ldots, t_n)$ of width $a \in \mathbb{N}$ is a partition of $\mathbb{R}_+^n$ into regions such that each region is described by:

- *mesh constraints:* for each variable t, by a constraint of the form $b - 1 < t \leq b$, where $b \in \mathbb{N}$ and $1 \leq b \leq a$;
- *diagonal constraints:* for every pair of distinct variables t and t' by an ordering on the fractional parts of the variables, i.e. by a constraint of the form $(t - \lfloor t \rfloor) \sim (t' - \lfloor t' \rfloor)$, where $\sim \in \{<, >\}$, and $\lfloor s \rfloor$ denotes the largest integer not greater than s.

For a region r and a variable t, let $Int_r(t)$ be the function that returns the mesh constraint constant of t in r.

[4] The reason for all harmonic functions to be defined on the same interval $(0, 1]$ will become clear later when we present the algorithm that constructs integral equations.

Given a region r we consider a (total) *region ordering* $\prec_r$ of fractional parts of $t_1, \ldots, t_n$, i.e. $t_i \prec_r t_j$ iff $(t_i - \lfloor t_i \rfloor) < (t_j - \lfloor t_j \rfloor)$. Thus, each region r in R_a can be described by the order $\prec_r$, and the unit intervals for every variable.

3.2 Algorithm

Suppose we are given a 1GSMP $G = (Q, \Sigma, E_{\text{inherited}}, E_{\text{new}}, \mathit{init}, \mathit{distr}, \mathit{next})$, Q_d — the set of destination locations and Q_u — the set of unsafe locations. We assume that all locations in $Q_d \cup Q_u$ are absorbing. We describe the algorithm in two steps. We start by discussing the main loop, and then we describe construction of the right-hand sides of the integral equations.

In the loop, the algorithm goes over all locations of G. For each destination location q_d it outputs equation $H_{q_d} = 0$, which states that being in a destination location, the probability to reach an unsafe location before any of destination locations is zero. For each unsafe location q_u, the algorithm outputs $H_{q_u} = 1$. If q is neither a destination nor an unsafe location, then, in case q has an inherited event $e_q^{\text{inherited}}$, the algorithm constructs a equations for each of the functions $H_q^1, \ldots, H_q^a$ of the same argument $t_{e_q^{\text{inherited}}}$. In case q does not have an inherited event, a single equation for H_q is constructed.

Algorithm 1 returns the right-hand side for the equation that defines the harmonic function $H_q^i(t_{e_q^{\text{inhereted}}})$, where $e_q^{\text{inhereted}}$ is the only inherited event of q (the algorithm for H_q, such that q has new events only is very similar).

Let us assume that the number of active events in q is n. For every non-constant function $H_p^j(t)$ let $\tilde{H}_p^j(t) = H_p^j(1-t)$.

At line 2 we have the loop that goes through all regions in the diagonal mesh partition $R_a(t_{e_1}, \ldots, t_{e_n})$ for which $t_{e_q^{\text{inhereted}}} \in (i-1, i]$. The restriction is required because we are constructing RHS for $H_q^i(t_{e_q^{\text{inhereted}}})$, which implies that $t_{e_q^{\text{inhereted}}} \in (i-1, i]$. At line 3 we determine the clock that should fire, i.e. the clock which is minimal in respect to $\prec_r$ among all the clocks that have the minimal value returned for them by Int_r. At line 4 we ensure that we consider every transition that may be caused by firing of e_r^* along with its probability. At lines $8 - 10$ we create a product of all new clock densities, each enters with its own variable. At line 11 we check if the target location has an inherited clock. If it does then at line 15 or 18 we determine to which interval this inherited clock belongs. This is uniquely determined by the region r. At line 22 we construct the integrand.

From line 24 to line 37 we have the loop that goes over all active event clocks of q. For each new clock we integrate over all possible values that this clock can have in r. The limits of integration are constructed in such a way that they respect $\prec_r$. For example, suppose that in the ordering $t_{e_1} \prec_r t_{e_2} \prec_r t_{e_3} \prec_r t_{e_4} \prec_r t_{e_5} \prec_r t_{e_6} \prec_r \cdots \prec_r t_{e_n}$, t_{e_3} is the inherited clock and the others are new clocks. Then we know that t_{e_1} can have values between 0 and t_{e_2}, t_{e_2} between 0 and t_{e_3}, t_{e_4} between t_{e_3} and t_{e_5}, t_{e_5} between t_{e_3} and t_{e_6} and so on. The last clock t_{e_n} can have values between t_{e_3} and 1. This idea is captured in the algorithm.

Algorithm 1 Generate RHS(q, i)

```
$RHS \leftarrow 0$
for all $r\colon (r \in R_a(t_{e_1}, \ldots, t_{e_n}), e_i \in E(q)) \wedge Int_r(t_{e_q^{\text{inherited}}}) = i$ do
    $e_r^* \leftarrow$ the firing clock of $r$
    for all $tran \in next(q, e_r^*)$ do
        $q_{\text{target}} \leftarrow$ the target location of the transition $tran$
        $Prob \leftarrow$ the probability of the transition $tran$ which is $P_{\text{next}}^{q,e_r^*}(q_{\text{target}})$
        $DensProduct \leftarrow 1$
        for all $e \in E_{\text{new}}(q)$ do
            $DensProduct \leftarrow DensProduct * dens_e^{Int_r(t_e)}(t_e)$
        end for
        if $E_{\text{inherited}}(q_{\text{target}}) = \emptyset$ then
            $HarmonicFunction \leftarrow H_{q_{\text{target}}}$
        else
            if $e_r^* \prec_r e_{q_{\text{target}}}^{\text{inherited}}$ then
                $index \leftarrow Int_r(t_{e_{q_{\text{target}}}^{\text{inherited}}}) - Int_r(t_{e_r^*}) + 1$
                $HarmonicFunction \leftarrow H_{q_{\text{target}}}^{index}(t_{e_r^*} - t_{e_{q_{\text{target}}}^{\text{inherited}}})$
            else
                $index \leftarrow Int_r(t_{e_{q_{\text{target}}}^{\text{inherited}}}) - Int_r(t_{e_r^*})$
                $HarmonicFunction \leftarrow \tilde{H}_{q_{\text{target}}}^{index}(t_{e_r^*} - t_{e_{q_{\text{target}}}^{\text{inherited}}})$
            end if
        end if
        $Integrand = HarmonicFunction * DensProduct$
        $LowerLimit \leftarrow 0$
        for $i = 1$ to $n$ do
            $e_{\text{cur}} \leftarrow e_i$, where $t_{e_i}$ is the $i^{\text{th}}$ element in $t_{e_1} \prec_r t_{e_2} \prec_r \cdots \prec_r t_{e_n}$
            if $e_{cur} \in E_{\text{inherited}}(q)$ then
                $LowerLimit \leftarrow t_{e_{\text{cur}}}$
            else
                if $i < n$ then
                    $UpperLimit \leftarrow t_{e_{i+1}}$, where $t_{e_{i+1}}$ is the $(i+1)^{\text{th}}$ element in $t_{e_1} \prec_r t_{e_2} \prec_r \cdots \prec_r t_{e_n}$
                else
                    $UpperLimit \leftarrow 1$
                end if
                $Integrand = \int_{LowerLimit}^{UpperLimit} Integrand \, \mathrm{d}t_{e_{\text{cur}}}$
            end if
            $RHS \leftarrow RHS + Prob * Integrand$
        end for
    end for
end for
return $RHS$
```

At line 36 we add the constructed integral to RHS, and at line 40 we output the entire constructed expression.

Let us return to our sample GSMP G_1. For q_{C_b}, for example, the algorithm generates the following integral equation:

$$
\begin{aligned}
&H^1_{q_{C_b}}\left(t_{e_u}\right) = \int_{t_{e_u}}^{1} \int_{0}^{t_{e_u}} dens_{e_c}\left(t_{e_c}\right) dens_{e_r}\left(t_{e_r}\right) \mathrm{d}t_{e_c}\, \mathrm{d}t_{e_r} \\
&+ \int_{0}^{t_{e_u}} \int_{0}^{t_{e_r}} dens_{e_c}\left(t_{e_c}\right) dens_{e_r}\left(t_{e_r}\right) \mathrm{d}t_{e_c}\, \mathrm{d}t_{e_r} \\
&+ \int_{t_{e_u}}^{1} \int_{0}^{t_{e_u}} H^1_{q'_C}\left(t_{e_u} - t_{e_r}\right) dens_{e_c}\left(t_{e_c}\right) dens_{e_r}\left(t_{e_r}\right) \mathrm{d}t_{e_r}\, \mathrm{d}t_{e_c} \\
&+ \int_{0}^{t_{e_u}} \int_{0}^{t_{e_c}} H^1_{q'_C}\left(t_{e_u} - t_{e_r}\right) dens_{e_c}\left(t_{e_c}\right) dens_{e_r}\left(t_{e_r}\right) \mathrm{d}t_{e_r}\, \mathrm{d}t_{e_c}
\end{aligned}
$$

The first term is for the region r_1 with ordering $t_{e_c} \prec_{r_1} t_{e_u} \prec_{r_1} t_{e_r}$, the second is for r_2, $t_{e_c} \prec_{r_2} t_{e_r} \prec_{r_2} t_{e_u}$, the third is for r_3, $t_{e_r} \prec_{r_3} t_{e_u} \prec_{r_3} t_{e_c}$, and the last is for r_4, $t_{e_r} \prec_{r_4} t_{e_c} \prec_{r_4} t_{e_u}$. The first two terms do not include harmonic functions because $H_{q_f} = 1$. There are no terms for the orderings that start with t_{e_u} because $H_{q_s} = 0$.

4 System of Differential Equations

In this section we show how to convert the system of integral equations constructed in the previous section into a system of differential equations such that the solution of the former system is a solution of the latter. Compared to the existing methods [16], our method converts every intergral term individually and can handle equations with terms that contain multiple integrals.

Due to the lack of space, we give intuition only for the main step. The other steps use the fact that the class of DESOL expressions is closed under addition, multiplication, integration and differentiation [17]. The resulting system will consist of the equations obtained after converting integral equaitons, equations that define auxiliary functions and additional equations described in Section 4.1.

We show how to convert, for example, a term $T = \int_a^b H(t-t')dens(t)\,\mathrm{d}t$, where the integration limits a and b can be 0, 1 or clock variables, $H(t-t')$ is a harmonic function and $dens(t)$ is a function which is a solution of a LHODE:

$$
\sum_{l=1}^{n} a_l \frac{\mathrm{d}^l dens(t)}{\mathrm{d}t^l} + a_0 dens(t) = 0, \tag{1}
$$

Given a harmonic function $H(t-t')$, we introduce an auxiliary function $A(t-t')$. The differential equation that links $A(t-t')$ to $H(t-t')$ is constructed from (1):

$$
H(t-t') = \sum_{l=1}^{n} (-1)^l a_l \frac{\mathrm{d}^l A(t-t')}{\mathrm{d}t^l} + a_0 A(t-t').
$$

Now replacing $H(t-t')$ in T with this equation, we will obtain that

$$T = \sum_{l=1}^{n} (-1)^l a_l \int_a^b \frac{\mathrm{d}^l A(t-t')}{\mathrm{d}t^l} \mathit{dens}(t)\,\mathrm{d}t + a_0 \int_a^b A(t-t')\mathit{dens}(t)\,\mathrm{d}t. \tag{2}$$

Consider the summand for $l = n$ in the equation above. Let us apply the integration by parts method to it:

$$(-1)^n a_n \int_a^b \frac{\mathrm{d}^n A(t-t')}{\mathrm{d}t^n} \mathit{dens}(t)\,\mathrm{d}t = (-1)^n a_n \int_a^b \mathit{dens}(t)\,\mathrm{d}\left(\frac{\mathrm{d}^{n-1} A(t-t')}{\mathrm{d}t^{n-1}}\right)$$

$$= (-1)^n a_n \left(\mathit{dens}(b) \left.\frac{\mathrm{d}^{n-1} A(t-t')}{\mathrm{d}t^{n-1}}\right|_{t=b} - \mathit{dens}(a) \left.\frac{\mathrm{d}^{n-1} A(t-t')}{\mathrm{d}t^{n-1}}\right|_{t=a} \right)$$

$$- (-1)^n a_n \int_a^b \frac{\mathrm{d}^{n-1} A(t-t')}{\mathrm{d}t^{n-1}} \frac{\mathrm{d}\,\mathit{dens}(t)}{\mathrm{d}t}\,\mathrm{d}t$$

Next we apply the integration by parts method $n-1$ more times:

$$\begin{aligned}
&(-1)^n a_n \left(\mathit{dens}(b) \left.\frac{\mathrm{d}^{n-1} A(t-t')}{\mathrm{d}t^{n-1}}\right|_{t=b} - \mathit{dens}(a) \left.\frac{\mathrm{d}^{n-1} A(t-t')}{\mathrm{d}t^{n-1}}\right|_{t=a} \right) \\
&+ (-1)^n a_n \sum_{k=2}^{n-1} (-1)^{n-k} \left(\left.\frac{\mathrm{d}^{n-k} \mathit{dens}(t)}{\mathrm{d}t^{n-k}} \frac{\mathrm{d}^{k-1} A(t-t')}{\mathrm{d}t^{k-1}}\right|_{t=a}^{t=b} \right) \qquad (3) \\
&+ a_n \int_a^b A(t-t') \frac{\mathrm{d}^n \mathit{dens}(t)}{\mathrm{d}t^n}\,\mathrm{d}t = TD_n + a_n \int_a^b A(t-t') \frac{\mathrm{d}^n \mathit{dens}(t)}{\mathrm{d}t^n}\,\mathrm{d}t,
\end{aligned}$$

where TD_n is a sum of differentials. Now we apply the same conversion to the remaining summands of (2). Let consider the sum of all converted summands:

$$T = \sum_{l=1}^{n} TD_l + \sum_{l=1}^{n} a_l \int_a^b A(t-t') \frac{\mathrm{d}^l \mathit{dens}(t)}{\mathrm{d}t^l}\,\mathrm{d}t + a_0 \int_a^b A(t-t')\mathit{dens}(t)\,\mathrm{d}t$$

$$= \sum_{l=1}^{n} TD_l + \int_a^b A(t-t') \left(\sum_{l=1}^{n} a_l \frac{\mathrm{d}^l \mathit{dens}(t)}{\mathrm{d}t^l} + a_0 \mathit{dens}(t) \right) \mathrm{d}t = \sum_{l=1}^{n} TD_l.$$

The last step follows from (1).

4.1 Additional Equations

Consider a product $\left.\frac{\mathrm{d}^{n-k} \mathit{dens}(t)}{\mathrm{d}t^{n-k}} \frac{\mathrm{d}^{k-1} A(t-t')}{\mathrm{d}t^{k-1}}\right|_{t=b}$ from (3), and let us assume that $b = 1$, then using the calculus chain rule:

$$\left.\frac{\mathrm{d}^{n-k} \mathit{dens}(t)}{\mathrm{d}t^{n-k}} \frac{\mathrm{d}^{k-1} A(t-t')}{\mathrm{d}t^{k-1}}\right|_{t=1} = (-1)^{k-1} \left.\frac{\mathrm{d}^{n-k} \mathit{dens}(t)}{\mathrm{d}t^{n-k}} \frac{\mathrm{d}^{k-1} A(t-t')}{\mathrm{d}t'^{k-1}}\right|_{t=1}$$

$$= (-1)^{k-1} C_{\mathit{dens}} \frac{\mathrm{d}^{k-1} A(1-t')}{\mathrm{d}t'^{k-1}} = (-1)^{k-1} C_{\mathit{dens}} \frac{\mathrm{d}^{k-1} \tilde{A}(t')}{\mathrm{d}t'^{k-1}},$$

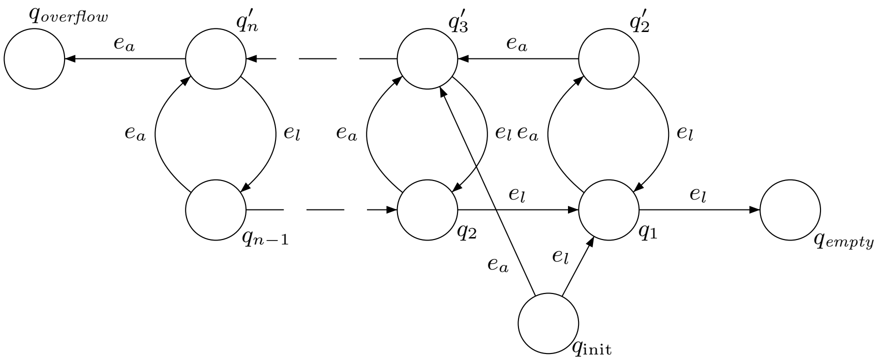


Fig. 2. Queue

where $C_{dens} = \left.\frac{\mathrm{d}^{n-k}\,dens(t)}{\mathrm{d}t^{n-k}}\right|_{t=1}$ and $\tilde{A}(t) = A(1-t)$.

Recall that our system may also include unknowns of the form $\tilde{H}(t) = H(1-t)$. To ensure that the number of unknowns coincide with the number of equations we need to add new differential equations for $\tilde{H}(t)$ and $\tilde{A}(t)$. For that we take the equation with $H(t)$ on its left-hand side and do change of variable from t to $t' = 1-t$. For $\tilde{A}(t)$ we do change of variables for the equation that defines $A(t)$. We add the equations for all such functions to our system and solve it.

5 Implementation

To demonstrate our tool we consider an example motivated by research on power-aware devices. Suppose that a device processes requests. Unprocessed requests can be stored in the device's queue of a finite length n. To save power it is preferable to accumulate as many requests as possible in the queue and then process them in one batch until the queue is empty. We know the distribution of the interarrival time between two successive requests, and the distribution on the time it takes for the device to process a request. These distributions are not exponential, so we are dealing with $GI/G/1$ queue.

Suppose we given the number k of requests in the queue at the moment the device starts batch processing. We want to know what is the probability that the queue overflows before it gets empty.

The 1GSMP for our example (when k is set to 2) is shown in Figure 2. A numeric index of a location is also the number of requests in the queue while a run is in that location. The firing of events e_a and e_l indicate a new request arrival and request completion, respectively. The density for e_a is t_{e_a} on $(0,1]$ and $t_{e_a} - 1$ on $(1,2]$. The density for e_l is $2/3$ on $(0,1]$ and $1/3$ on $(1,2]$.

Experiments were conducted on a Windows XP computer with a Pentium D processor running at 2.80 GHz with 2 GB of RAM. The results are presented in the table below.

Parameters		**Results**	
n	k	P_{overflow}	Running time
16	1	7.5401361×10^{-8}	5 min. 4 sec.
16	8	0.00010769	5 min. 2 sec.
16	16	0.53083234	5 min. 9 sec.
32	1	$7.4714055 \times 10^{-16}$	47 min. 56 sec.
32	16	1.8367065×10^{-8}	41 min. 38 sec.
32	32	0.53083236	42 min. 46 sec.

References

1. R. Alur and M. Bernadsky. Bounded model checking for GSMP models of stochastic real-time systems. In *Hybrid Systems: Computation and Control, Proc. of 9th Int. Workshop*, LNCS 3927, pages 19–33. Springer, 2006.
2. R. Alur, C. Courcoubetis, and D.L. Dill. Model-checking for probabilistic real-time systems. In *Automata, Languages and Programming: Proceedings of the 18th ICALP*, LNCS 510, pages 115–136. Springer-Verlag, 1991.
3. R. Alur and D.L. Dill. A theory of timed automata. *TCS*, 126:183–235, 1994.
4. S. Asmussen. *Applied Probability and Queues.* Springer, 2006.
5. A. Aziz, K. Sanwal, V. Singhal, and R.K. Brayton. Model-checking continuous-time Markov chains. *ACM Trans. on Computational Logic*, 1(1):162–170, 2000.
6. H. Choi, V.G. Kulkarni, and K. Trivedi. Markov regenerative stochastic Petri nets. Performance Evaluation, 20:337–357, 1994.
7. A. Cumani. Esp — A package for the evaluation of stochastic Petri nets with phase-type distributed transition times. In *Proc. of Int. Workshop on Timed Petri Nets*, pages 144–151, Torino (Italy), 1985. IEEE Computer Society Press no. 674.
8. R. German. *Performance analysis of communication systems: Modeling with non-Markovian stochastic Petri nets.* J. Wiley & Sons, 2000.
9. P.W. Glynn. A GSMP formalism for discrete event systems. *Proceedings of the IEEE*, 77(1):14–23, 1988.
10. B. Haverkort. *Performance of computer-communication systems: A model-based approach.* Wiley & Sons, 1998.
11. M.Z. Kwiatkowska. Model checking for probability and time: from theory to practice. In *Proc. of the 18th IEEE Symposium on Logic in CS*, pages 351–360, 2003.
12. M.Z. Kwiatkowska, G. Norman, and D. Parker. Probabilistic symbolic model checking with PRISM: a hybrid approach. *STTT*, 6(2):128–142, 2004.
13. C. Lindemann and A. Thümmler, Numerical Analysis of Generalized Semi-Markov Processes. Research Report No. 722, Dept. of CS, University of Dortmund, 1999.
14. Gabriel G. lnfante López, Holger Hermanns and Joost-Pieter Katoen Beyond Memoryless Distributions: Model Checking Semi-Markov Chains. In *Proceedings of PAPM-PROBMIV*, LNCS 2165, pages 57–70, 2001.
15. G.S. Shedler. *Regenerative stochastic simulation.* Academic Press, 1993.
16. H. Ye and R. Corless. Solving linear integral equations in Maple. In *Proc. of the Int. Symposium on Symbolic and Algebraic Computation*, pages 95–102, 1992.
17. D. Zeilberger. Holonomic Systems Approach To Special Functions. *J. Computational and Applied Math*, 32:321–368, 1990.