2006

# Bounded Model Checking of GSMP Models of Stochastic Real-Time Systems

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

Mikhail Bernadsky
*University of Pennsylvania*

# Bounded Model Checking of GSMP Models of Stochastic Real-Time Systems

**Abstract**
Model checking is a popular algorithmic verification technique for checking temporal requirements of mathematical models of systems. In this paper, we consider the problem of verifying bounded reachability properties of stochastic real-time systems modeled as generalized semi-Markov processes (GSMP). While GSMPs is a rich model for stochastic systems widely used in performance evaluation, existing model checking algorithms are applicable only to subclasses such as discrete-time or continuous-time Markov chains. The main contribution of the paper is an algorithm to compute the probability that a given GSMP satisfies a property of the form "can the system reach a target before time $T$ within $k$ discrete events, while staying within a set of safe states". For this, we show that the probability density function for the remaining firing times of different events in a GSMP after $k$ discrete events can be effectively partitioned into finitely many regions and represented by exponentials and polynomials. We report on illustrative examples and their analysis using our techniques.

**Disciplines**
Computer Sciences

**Comments**
From the 9th International Workshop, HSCC 2006, Santa Barbara, CA, USA, March 29-31, 2006.

# Bounded Model Checking for GSMP Models of Stochastic Real-time Systems*

Rajeev Alur and Mikhail Bernadsky

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

**Abstract.** Model checking is a popular algorithmic verification technique for checking temporal requirements of mathematical models of systems. In this paper, we consider the problem of verifying bounded reachability properties of stochastic real-time systems modeled as generalized semi-Markov processes (GSMP). While GSMPs is a rich model for stochastic systems widely used in performance evaluation, existing model checking algorithms are applicable only to subclasses such as discrete-time or continuous-time Markov chains. The main contribution of the paper is an algorithm to compute the probability that a given GSMP satisfies a property of the form "can the system reach a target before time $T$ within $k$ discrete events, while staying within a set of safe states". For this, we show that the probability density function for the remaining firing times of different events in a GSMP after $k$ discrete events can be effectively partitioned into finitely many regions and represented by exponentials and polynomials. We report on illustrative examples and their analysis using our techniques.

## 1 Introduction

Probabilistic modeling is commonly used in the design and performance evaluation of a wide range of real-time systems such as communication protocols and multi-media systems ([11, 8]). Traditional analysis of probabilistic models involves simulations, and is used to obtain estimates of quality-of-service metrics such as mean delivery time for a message. In contrast, formal verification techniques are aimed at checking whether or not a system model satisfies a functional correctness property such as "every message is eventually delivered." Model checking has emerged as a viable method for formal verification for debugging critical components in industrial settings ([6, 5, 12]). The goal of probabilistic model checking is to integrate the two approaches so that a probabilistic model of a real-time system can be algorithmically checked against a specification such as "every message is delivered within 1ms with probability 0.9."

Early work on probabilistic model checking considers discrete models such as finite-state Markov chains or Markov decision processes, and requirements given by temporal logics or automata, and shows how to algorithmically compute the probability that a model satisfies the requirement ([19, 7, 10]). More recent work allows modeling using

* This research was supported by the US National Science Foundation via grants CCR-0410662 and ITR/SY 0121431.

continuous-time Markov chains (CTMCs), and specifications written in temporal logics such as CSL and PCTL that allow requirements with time and probability ([3, 15, 16]). Issues concerning symbolic representation and efficient implementation have also been studied leading to a number of probabilistic model checkers ([13, 17]). In particular, the model checker PRISM has been applied to a number of case studies in distributed protocols and embedded systems (see `http://www.cs.bham.ac.uk/~dxp/prism`).

In this paper, we consider the probabilistic model checking problem for systems modeled as *Generalized Semi-Markov Processes* (GSMPs) ([9, 18, 8]). 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(s) with the least remaining firing time happens, the state is updated probabilistically, and new events can be scheduled at times chosen randomly according to distributions described by exponential and polynomial density functions with finitely many discontinuities, which we call *expolynomial region distributions* (ERDs). Unlike CTMCs, such distributions need not be memoryless, and the class of ERDs includes uniform or polynomial distributions over finite intervals, point distributions over finitely many constant values, and exponentials.

The classical way to analyze GSMP models involves Monte Carlo simulations. In [1], 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 [2]. Region graph, however, is not adequate for computing *quantitative* probabilistic properties as different configurations in the same region have different probabilities of satisfaction of properties. In [14], the authors show that by refining the region graph, one can approximate the satisfaction of quantitative probabilistic properties, while [20] shows that statistical sampling can be adopted to compute estimates for the model checking problem. The literature on stochastic Petri nets shows how GSMPs can be approximated by Markovian models [8]. In this paper, we show 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. More specifically, given a finite-state GSMP $M$, a target set $F$, a safety set $S$, a bound $k$ on the number of discrete events, we show how to compute the probability that $M$ will reach $F$, while staying within the set $S$, within $k$ discrete events (and also, within a time bound $T$, if specified). The bound $k$ is analogous to the bound on the lengths of paths used in recent work on *bounded model checking* of discrete Boolean systems using SAT solvers [4].

For quantitative analysis of a GSMP, we need to effectively represent and compute the distribution on the remaining firing times of scheduled events when the event(s) with the least firing time happens. For this purpose, we consider multidimensional expolynomial region distributions: the space of configurations is divided into finitely many regions using axis-parallel and diagonal constraints similar to the region graph, and with each region, the density function is continuous represented by a combination of exponential and polynomial functions. Our main technical construction shows that the class of ERDs is effectively closed under expiration of events and scheduling of new events. This leads to an iterative symbolic algorithm which computes the probability distribution after each discrete step.

We are implementing our modeling and analysis approach in a tool called *Event Horizon Verifier*, and we illustrate it using a classical example from queuing networks. Consider a buffer for which the interarrival time between successive messages from the producer, and the processing time for a message by the consumer, are described by ERDs. Given a capacity $N$, suppose we want to calculate the probability that the number of unprocessed messages exceeds $N$. Then, our analysis allows us to compute this probability, given a bound on the total number of events.

## 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.

In a GSMP the time between scheduling an event and its occurrence (or firing time) is modeled as a positive random variable. For this reason we briefly review related terminology. 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 $distr(x) = \int_0^x dens(y)\, dy$. For many modeling purposes, however, it is convenient to use random variables whose cdf's are not continuous. For instance, it may be necessary to model the firing time of an event by a random variable that takes only a finite number of possible values. We say that $x \in \mathbb{R}_+$ is a *mass point* of $X$ if $\Pr(X = x) > 0$[1]. We will see that for random variables with a finite number of mass points it is still possible to define a function with properties similar to those of the pdf of a random variable with continuous cdf.

We say that an expression $e(x)$ is *expolynomial* if it can be written as $\sum_{k=1}^{r} c_k x^{m_k} e^{\lambda_k x}$, where $c_k, \lambda_k \in \mathbb{R}$, $m_k \in \mathbb{N}_0$, for all $k = 1, \ldots, r$. Let $Expr(x)$ be the set of all expolynomial expressions. Consider a partition $R_a$ of $\mathbb{R}_+$, which consists of $a$ bounded intervals followed by an unbounded interval and the points between them: $R_a = \cup_{i=0}^{a-1} \{i, (i, i+1)\} \cup \{a, (a, +\infty)\}$. The constant $a$ is the *width* of $R_a$. We say that a function $f(x)$ is *expolynomial* with finite support on $R_a$ if there exists a map $M_f \colon R_a \to Expr(x)$, such that for all $x \in \mathbb{R}_+$, $f(x) = M(r)(x)$, where $x \in r$ and $r \in R_a$ (i.e. $r$ is either an interval or a point).

**Definition 1.** *A (unidimensional) random variable $X$ has an expolynomial region distribution of width $a$, if there exists an expolynomial function $dens(x) \geq 0$ on $R_a$, such that for all $t \in \mathbb{R}_+$, $\Pr(X < t) = \sum_{I \in I_{R_a}} \int_{I \cap (y<t)} dens(y)\, dy + \sum_{i=1}^{\min(a, \lfloor t \rfloor)} dens(i) = \int_0^t dens(y)\, dy + \sum_{i=1}^{\min(a, \lfloor t \rfloor)} dens(i)$, where $I_{R_a}$ is the set of all intervals in $R_a$, and $\lfloor t \rfloor$ denotes the largest integer no greater than $t$.*

We call $dens(x)$ the pdf of $X$. Notice, that $X$ has a mass point at $i$ iff $dens(i) > 0$ and $i \in \{1, \ldots, a\}$.

Uniform distributions, exponential and truncated exponential distributions, finite discrete distributions are all examples of expolynomial region distributions. Many other

[1] Mass points can also be treated using Dirac delta function $\delta(x)$. We have chosen not to do so because this approach leads to cumbersome expressions in the multidimensional settings.

distributions with continuous and discrete components can be approximated by expolynomial distributions. Our definition requires finite intervals to be of the unit length and mass points to occur at a finite number of points in $\mathbb{N}$, however this is done only to simplify the presentation of the results. In general, it is sufficient if a distribution is defined by expolynomial expressions on a finite number of intervals with rational endpoints, and has only a finite number of mass points.

Now we are ready to give a formal definition of the class of stochastic processes that we study in this paper.

**Definition 2.** *A finite-state generalized semi-Markov process (GSMP) is a tuple* $A = (Q, \Sigma, E, init, distr, next)$ *where:*

- $Q$ *is a finite set of locations;*
- $\Sigma$ *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* $A$*;*
- $distr\colon \Sigma \to (\mathbb{R}_+ \to [0,1])$ *assigns to each event its* firing time distribution*, which is an expolynomial region distribution. For a cdf* $distr(e)$*,* $dens(e)$ *denotes the corresponding pdf.*
- $next\colon Q \times (2^{\Sigma} \setminus \{\emptyset\}) \to (Q \to [0,1])$ *defines transitions between the locations of* $A$*. This function takes as its arguments a source location* $q$ *and a non-empty subset* $G$ *of the active events of* $q$*, and returns a probability measure on* $Q$*. For each location* $q'$*, this measure gives the probability that* $A$ *will move from* $q$ *to* $q'$ *if all events in* $G$ *occur simultaneously; we require that* $\sum_{q' \in Q} next(q, G)(q') = 1$ *for all* $G \subseteq 2^{E(q)} \setminus \{\emptyset\}$*.*

It is convenient to think that a clock is assigned to each event $e$. Upon (re-)scheduling of $e$ we update its clock to a new valuation chosen independently at random according to $distr(e)$.The clock shows the time remaining until the next occurrence of $e$. Every clock runs 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 $\nu$ 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.

A *configuration* of the GSMP $A$ is a pair $s = (q, \nu)$, where $q \in Q$ and $\nu$ 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 $E^*(s) = \{e^* \mid e^* = \arg\min\{\nu(e), e \in E(q)\}\}$ be the set of events that causes the transition (the clocks of these events expire simultaneously). For any $t \leq t^*(s)$ we denote by $\nu - t$ the valuation $\nu'$ 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 $t^*(s) = 0$, then $E^* = \{e^* \mid \nu(e^*) = 0\}$, and $s \xrightarrow{\mu} s'$ denotes a *discrete transition* between the configurations $s = (q, \nu)$ and $s' = (q', \nu')$, where $q'$ is chosen according to the probability measure $\mu = next(q, E^*)$, and the valuation $\nu'$ is constructed as follows:

1. if an event $e \in E_{\mathrm{old}}(q, E^*, q')$, where $E_{\mathrm{old}}(q, E^*, q') = E(q') \cap [E(q) \setminus E^*]$ is the set of events, excluding the events in $E^*$, that were active in $q$ and continue to be active in $q'$, then $\nu'(e) = \nu(e)$;

2. if $e \in E_{\text{new}}(q, E^*, q')$, where $E_{\text{new}}(q, E^*, q') = E(q') \setminus E_{\text{old}}(q, E^*, q')$ is the set of events that were not active in $q$ but become active in $q'$ and events that are in $E^* \cap E(q')$ (i.e. events that fired in $q$ and are active in $q'$), 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);
3. 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 cancelled events that were active in $q$ but no longer active in $q'$, then $\nu'(e)$ is undefined.

A *run* $\sigma$ of $A$ 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 $\sigma$ starts at the initial configuration $s_0 = (q_0, \nu_0)$, $q_0$ is the initial location,which is chosen according to $init$, and $\nu_0$ is the initial valuations 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.

The time of the $n^{th}$ transition is the time $T_n(\sigma) = \sum_{i=0}^{n-1} t^*(s_i)$ that elapsed since the start of $\sigma$ and until the $n^{th}$ discrete transition.

*Example 1.* Let us describe a GSMP $A_s$, which we will use as our running example. $A_s$ has six locations, $q_0$ is the initial location (i.e. $init$ picks this location with probability one), and locations $q_2$, $q_3$, $q_4$, and $q_5$ are absorbing. In $q_0$ two events $e_1$ and $e_2$ are active, the initial clock valuations for these events are chosen according to their firing time density functions: $dens(e_1)(t_1) = Dt_1e^{-t_1}$ when $t_1 \in (0, 1)$ and 0 otherwise (the normalizing constant $D$ is equal to $1/(1 - 2e^{-1})$), and $dens(e_2)(t_2) = 1/2$ when $t_2 \in (0, 1) \cup (1, 2)$ and 0 otherwise, i.e. it is uniformly distributed on $(0, 2)$. If $e_1$ fires first, then the process moves to $q_1$ with probability 1, otherwise it moves to $q_2$ and stays there forever. In $q_1$ three events are active — $e_2$ whose clock keeps its valuation from $q_0$ and events $e_1$ and $e_3$ whose clocks obtain new valuations upon entering $q_1$. The firing time density function for $e_3$ is $dens(e_3)(t_3) = 1$ when $t_3 \in (0, 1)$ and 0 otherwise, and it describes the uniform distribution on $(0, 1)$. Firings of $e_1$, $e_2$ and $e_3$ in $q_1$ lead to locations $q_3$, $q_4$ and $q_5$, respectively.

A *history* $\pi$ of the length $n$ of a run $\sigma$ is a sequence of tuples and transitions between them marked with sets of events:

$$\pi = (q_0, \mathcal{O}_0, \mathcal{N}_0) \xrightarrow{E_1} (q_1, \mathcal{O}_1, \mathcal{N}_1) \xrightarrow{E_2} \dots \xrightarrow{E_n} (q_n, \mathcal{O}_n, \mathcal{N}_n)$$

Each tuple $(q_i, \mathcal{O}_i, \mathcal{N}_i)$ consists of a visited location and two sets that partition the set of active events of that location. The set $\mathcal{O}_i$ consists of active events that were not scheduled upon arriving to $q_i$ and the set $\mathcal{N}_i$ consists of active events that were scheduled. For the tuple $(q_0, \mathcal{O}_0, \mathcal{N}_0)$, we have that $\mathcal{O}_0 = \emptyset$ and $\mathcal{N}_0 = E(q_0)$, and for any $i > 0$, $\mathcal{O}_i = E_{\text{old}}(q_{i-1}, E_i, q_i)$ and $\mathcal{N}_i = E_{\text{new}}(q_{i-1}, E_i, q_i)$.

By $last(\pi) = q_n$ we will denote the last visited location in a history $\pi$, and by $\Pi$ we will denote the set of all finite histories.

It easy to see that two runs share the same history $\pi$ of length $n$ if they visit the same sequence of $n$ locations and transitions between those locations are caused by firing of the same sets of events.

We say that a history $\pi'$ is a *successor* of $\pi$ along an edge marked by a set of events $E$ iff there exists a tuple $(q_{l'}, \mathcal{O}_{l'}, \mathcal{N}_{l'})$ such that $\pi' = \pi \xrightarrow{E} (q_{l'}, \mathcal{O}_{l'}, \mathcal{N}_{l'})$.

**Definition 3.** *Let $\pi$ be a history of length $n$ and let $l = |E(last(\pi))|$ be the number of the active events in the last location of $\pi$, then the* event clock valuations of $\pi$ *(abbreviated as* ecv *of $\pi$) is an $l$-dimensional random variable of values of the active clocks in the location $last(\pi)$, immediately after it has been reached by the $n^{\text{th}}$ transition.*

Given a history $\pi$, we denote by $f^{\pi}(x_1, \ldots, x_l)$ the pdf of the event clock valuations of $\pi$. We will show how to use $f^{\pi}(x_1, \ldots, x_l)$ to compute probability $p_{\pi}$, which is called the *occurrence probability* of $\pi$ and which is equal to the probability that a run of $A$ has $\pi$ as its history.

## 3 Computing Probabilities of Bounded Until Properties

Suppose that we are given a GSMP $A$. The locations of $A$ are partitioned into two sets: $Q_s$ and $Q_u$ which are called the sets of *safe* and *unsafe* locations, respectively. Furthermore, a subset $Q_d$ of $Q_s$ is called the set of *destination locations*.

Let $\Pi^n_{\text{until}} \subseteq \Pi$ be a set of histories of length less than or equal to $n$, and such that for every $\pi \in \Pi^n_{\text{until}}$ all locations of $\pi$ belong to $Q_s$ and the only location that belongs to $Q_d$ is $last(\pi)$; let $\Pi_{\text{until}} = \cup_{n>0} \Pi^n_{\text{until}}$.

Given two parameters — a real number $p \in [0, 1]$ and an integer $n > 0$, we consider the *bounded until problem*:

– *Is the probability that a run $\sigma$ of $A$ has a history $\pi \in \Pi^n_{\text{until}}$ greater than $p$?*

Algorithm 1 is a generic algorithm to solve this problem. The algorithm works on tuples $(\pi, f^{\pi}, p_{\pi})$, the first element of a tuple is a history $\pi$, the second element is the ecv density $f^{\pi}$ of $\pi$, and the last element $p_{\pi}$ is the occurrence probability of $\pi$. Given $f^{\pi}$ and $p_{\pi}$, we assume (and we will prove later) that for any successor history $\pi'$ of $\pi$, we are able to compute $f^{\pi'}$ and $p_{\pi'|\pi}$ (which is the occurrence probability of $\pi'$ conditioned on the probabilistic event that $\pi$ has happened).

$HistorySet$ is the set of tuples that the algorithm has to process. The set is initialized with the tuples $(\pi^i_0, f^{\pi^i_0}, p_{\pi^i_0})$, where $\pi^i_0 = (q_i, \mathcal{O}^i_0, \mathcal{N}^i_0)$ for locations $q_i$ of $A$, such that $init(q_i) > 0$. The algorithm also sets to zero two real numbers $P_d$ and $P_u$, which are the lower bounds of reaching a destination location and an unsafe location, respectively. In the main loop, the algorithm picks a history from $HistorySet$ and checks if its last location is a destination or an unsafe location. If this is the case then it increases $P_d$ or $P_u$. If the last location is a safe location but not a destination location and the length of the history is less than $n$, then the algorithm computes $f^{\pi'}$ and $p_{\pi'|\pi}$ for every successor history $\pi'$ of $\pi$ and updates $HistorySet$ with the computed tuples. When the loop is completed, the algorithm outputs "$YES$" if $P_d > p$ and "$NO$" otherwise.

Suppose that in addition to the numbers $p$ and $n$, we are given a positive real number $T$. Then, applying our algorithm, we can also solve the *bounded timed-until problem*:

**Algorithm 1** Genereric iterative algorithm

**for all** $q_i : q_i \in Q \wedge init(q_i) > 0$ **do**
  $HistorySet \leftarrow (\pi_0^i, f^{\pi_0^i}, p_{\pi_0^i})$
**end for**
$P_d \leftarrow 0$, $P_u \leftarrow 0$
**while** $HistorySet \neq \emptyset \wedge P_d \leq p \wedge P_u \leq (1-p)$ **do**
  $pick\ (\pi, f^{\pi}, p_{\pi})\ in\ HistorySet$
  **if** $last(\pi) \in Q_d$ **then**
    $P_d \leftarrow P_d + p_{\pi}$
  **else if** $last(\pi) \in Q_u$ **then**
    $P_u \leftarrow P_u + p_{\pi}$
  **else if** $length\ of\ \pi < n$ **then**
    **for all** $\pi_s : \pi_s\ is\ a\ successor\ of\ \pi$ **do**
      $compute\ f^{\pi_s}\ and\ p_{\pi_s|\pi}$
      $add\ (\pi_s, f^{\pi_s}, p_{\pi} \cdot p_{\pi_s|\pi})\ to\ HistorySet$
    **end for**
  **end if**
**end while**
**if** $P_d > p$ **then**
  **return** $YES$
**else**
  **return** $NO$
**end if**

– *Is the probability that a run $\sigma$ of $A$ has a history $\pi \in \Pi^n_{\text{until}}$ and $T_{|\pi|}(\sigma) < T$ greater than $p$?*

The bounded timed-until problem can be reduced to the bounded until problem by introducing a new event $e_t$ and a new unsafe absorbing location $q_t$. The random variable that models firing time distribution for $e_t$ is equal to $T$ with probability one. For every location $q$ and every set of events $E$, such that $e_t \in E$, $next(q, E)$ returns a probability measure concentrated on $q_t$. Thus, if a destination location is reached then it is reached before time $T$ has elapsed.

### 3.1 A Sample Computation

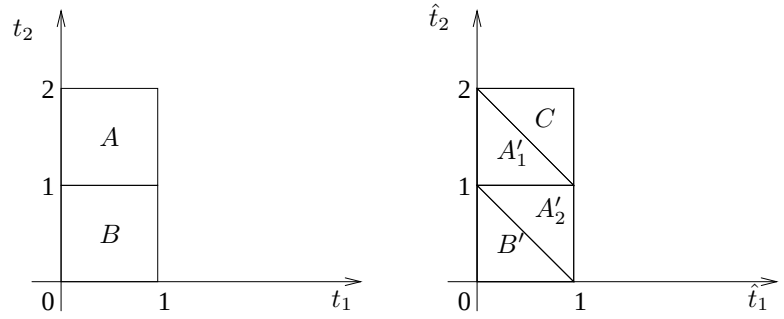
Consider the GSMP $A_s$ from Example 1 of Section 2. Given a history $\pi_1 = \pi_0 \xrightarrow{\{e_1\}} (q_1, \{e_2\}, \{e_1, e_3\})$, $\pi_0 = (q_0, \emptyset, \{e_1, e_2\})$, we want to compute $p_{\pi_1}$ and $f^{\pi_1}$.

Later, in Section 4, we will prove that to find $p_{\pi_1}$ and $f^{\pi_1}$ we need to compute three formulas:

$$\tilde{f}^{\pi_1}(t_2) = \int_0^{+\infty} dens(e_2)(t_1 + t_2) dens(e_1)(t_1)\, dt_1, \qquad p_{\pi_1} = \int_0^{+\infty} \tilde{f}^{\pi_1}(t_2)\, dt_2,$$

$$f^{\pi_1}(t_1, t_2, t_3) = dens(e_1)(t_1) \frac{\tilde{f}^{\pi_1}(t_2)}{p_{\pi_1}} dens(e_3)(t_3).$$

Intuitively, the first formula captures the necessary information on the distribution of values of the clock of $e_2$ in $q_1$, given that $e_1$ has fired before $e_2$. The second formula




(a) $dens(e_2)(t_2)dens(e_1)(t_1)$ (b) $dens(e_2)(\hat{t}_1 + \hat{t}_2)dens(e_1)(\hat{t}_1)$

**Fig. 1.** Computing $f^{\pi_1}$

shows that we can find $p_{\pi_1}$ by integrating $\tilde{f}^{\pi_1}(t_2)$ over all possible values. And the last formula gives an expression for $f^{\pi_1}(t_1, t_2, t_3)$ as a product of three density functions, each corresponds to an active clock of $q_1$. Even though the formulas above use integrals we will not use numerical computations, but instead we will obtain the formulas in an explicit form. This suitability for symbolic computations is a distinctive property of the expolynomial functions and we will use it throughout the paper.

We show two ways to compute the first formula. Let $1_{[a<t<b]}$ denote a function of $t$, which is 1 if $a < t < b$ and 0 otherwise. We know that $dens(e_1)(t_1) = Dt_1e^{-t_1}1_{[0<t_1<1]}$ and $dens(e_2)(t_2) = \frac{1}{2}1_{[0<t_2<2]}$, thus $\tilde{f}^{\pi_1}(t_2) = \frac{D}{2}\int_0^{+\infty} t_1e^{-t_1}1_{[0<t_1<1]}1_{[0<t_1+t_2<2]}\,dt_1 = \frac{D}{2}\int_0^1 t_1e^{-t_1}1_{[0<t_1+t_2<2]}\,dt_1$. We consider two cases:

- if $t_2 \in (0,1)$, then $\tilde{f}^{\pi_1}(t_2) = \frac{D}{2}\int_0^1 t_1e^{-t_1}\,dt_1 = \frac{1}{2}$;
- if $t_2 \in (1,2)$, then $\tilde{f}^{\pi_1}(t_2) = \frac{D}{2}\int_0^{2-t_2} t_1e^{-t_1}\,dt_1 = \frac{D}{2}(t_2e^{t_2-2} - 3e^{t_2-2} + 1)$.

Note that computing $\tilde{f}^{\pi_1}(t_2)$ requires analysis of different possible cases and the number of cases quickly becomes intractable with the increase in the number of active events in a location and complexity of firing time distributions. To deal with these difficulties we present now a more convenient "geometric" way to compute $\tilde{f}^{\pi_1}(t_2)$.

In Figure 1(a), the support for the function $dens(e_2)(t_2)dens(e_1)(t_1)$ is shown. It consists of two squares $A$ and $B$ (without the borders), and in each of these squares the function is equal to $\frac{D}{2}t_1e^{-t_1}$. Now consider a linear transformation $t_1 = \hat{t}_1$, $t_2 = \hat{t}_1 + \hat{t}_2$. Under this transformation the squares $A$ and $B$ are transformed into areas $A'_1 \cup A'_2$ and $B'$, respectively (see Figure 1(b)). The original function did not depend on $t_2$, and after the transformation the function will not depend on $\hat{t}_2$ either — it is equal to $\frac{D}{2}\hat{t}_1e^{-\hat{t}_1}$ in the areas $A'_1$, $A'_2$ and $B'$, and it is 0 in $C$. Now it is easy to see that if $\hat{t}_2 \in (0,1)$ then we have to compute two integrals, one over $B'$ and the other over $A'_2$: $\tilde{f}^{\pi_1}(\hat{t}_2) = \frac{D}{2}\int_0^{1-\hat{t}_2} \hat{t}_1e^{-\hat{t}_1}\,d\hat{t}_1 + \frac{D}{2}\int_{1-\hat{t}_2}^1 \hat{t}_1e^{-\hat{t}_1}\,d\hat{t}_1 = \frac{1}{2}$; and if $\hat{t}_2 \in (1,2)$ then we need to compute only one integral over $A'_1$: $\tilde{f}^{\pi_1}(\hat{t}_2) = \frac{D}{2}\int_0^{2-\hat{t}_2} \hat{t}_1e^{-\hat{t}_1}\,dt_1 = \frac{D}{2}(\hat{t}_2e^{\hat{t}_2-2} - 3e^{\hat{t}_2-2} + 1)$.

Now, using the second formula and renaming the variable $\hat{t}_2$ back to $t_2$, we obtain that

$$p_{\pi_1} = \int_0^2 \tilde{f}^{\pi_1}(t_2)\,dt = \int_0^1 \frac{1}{2}\,dt_2 + \frac{D}{2}\int_1^2 (t_2 e^{t_2-2} - 3e^{t_2-2} + 1)\,dt_2$$
$$= \frac{1}{2} + \frac{D}{2}(3e^{-1} - 1) \approx 0.7$$

Finally, $f^{\pi_1}(t_1,t_2,t_3) = Dt_1e^{-t_1}1_{[0<t_1<1]}\cdot(\frac{1}{2p_\pi}1_{[0<t_2<1]}+\frac{D}{2p_\pi}(t_2e^{t_2-2}-3e^{t_2-2}+1)1_{[1<t_2<2]})\cdot 1_{[0<t_3<1]}$. Again, for this function we can have a convenient geometric representation but this time in three dimensions.

## 4 Multidimensional Expolynomial Region Distributions

In this section we introduce multidimensional expolynomial region distributions. We are interested in this class because it is closed under symbolic computations that we will use. It follows that if the firing time distributions of the events are (one-dimensional) ERDs then all the distributions that we will encounter will also be ERDs. Before giving a formal definition, we describe a class of partitions of the clock valuation space that we will call *diagonal mesh partitions*. These partitions serve as domains for the ERDs — in each region of a diagonal mesh partition, an ERD is given by a multidimensional *expolynomial expression*.

### 4.1 Diagonal and Inverse Diagonal Mesh Partitions

For a set of variables $t_1,\ldots,t_n$ an $n$-dimensional diagonal mesh partition $R_a$ 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 $b-1 < t < b$ (we say that such a constraint is *bounded*), or $t = b$, or $t > a$ (an *unbounded* constraint), where $b \in \mathbb{N}$ and $b \leq a$;
- *diagonal constraints:* for every pair of different variables $t$ and $t'$, such that both of them have bounded mesh constraints in the region, 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 $\sim \in \{<,>,=\}$. Equivalently, if there are constraints $b-1 < t < b$ and $c-1 < t' < c$, then the diagonal constraint can be written as $t \sim t' + (b-c)$.

Given a region $r$ of an $n$-dimensional diagonal mesh partition $R_a$, let $m$ be the number of independent constraints of the form $t = b$ or $t = t' + b$, then we say that the dimension of $r$ is $n-m$. The regions that have dimension $n$ are called *full dimensional regions*, and regions that have less than $n$ dimensions are called *mass regions*.

For technical reasons, we will be also interested in the *inverse diagonal mesh partitions*. Compared to the diagonal mesh partitions these partitions have one designated variable $t^*$, which cannot form diagonal constraints with any other variable but, instead, it forms inverse diagonal constraints. Formally, for a set of variables $t_1,\ldots,t_{n-1},t^*$ an $n$-dimensional inverse diagonal mesh partition $\tilde{R}_a$ 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 $t \in \{t_1, \ldots, t_{n-1}, t^*\}$ by a mesh constraint, as described in the definition of diagonal mesh partition;
– *diagonal constraints:* for every pair of different variables $t \neq t^*$ and $t' \neq t^*$ with bounded mesh constraints, by a diagonal constraint, as described in the definition of diagonal mesh partition;
– *inverse diagonal constraints:* for every pair of variables $t$ and $t^*$, such that for each of them there is a bounded mesh constraint, by a constraint of the form $(t - \lfloor t \rfloor) + (t^* - \lfloor t^* \rfloor) \sim 1$, where $\sim \in \{<, >, =\}$. Equivalently, if there are constraints $b - 1 < t < b$ and $c - 1 < t^* < c$, then the inverse diagonal constraint can be written as $t + t^* + 1 \sim b + c$.

Note that the number of the regions in every diagonal or inverse diagonal mesh partition is finite, and exponential in the number of variables. Note also that the constraints can be seen as hyperplanes in $\mathbb{R}^n_+$.

Next we will consider an important linear transformation $\mathcal{L}\colon \mathbb{R}^n_+ \to \mathbb{R}^n_+$. Let $p = (t_1, \ldots, t_{n-1}, t^*)$ be a point that $\mathcal{L}$ maps to a point $\hat{p} = (\hat{t}_1, \ldots, \hat{t}_{n-1}, \hat{t}^*)$, then coordinates of $p$ and $\hat{p}$ are related by the following equations: $t_i = \hat{t}_i + \hat{t}^*$, for $i = 1, \ldots, n-1$ and $t^* = \hat{t}^*$. We have seen an application of $\mathcal{L}$ in Section 3.1. The properties of the partitions are given by the following lemmas. Due to the lack of space, we omit the proofs.

**Lemma 1.** *Let $R_a$ be an $n$-dimensional diagonal mesh partition of width $a$. Then $\mathcal{L}$ transforms $R_a$ into an $n$-dimensional inverse diagonal mesh partition $\hat{R}_a$ of the same width. The pre-image of any $l$-dimensional region in $\hat{R}_a$, for $l = 0, \ldots, n$, is a (part of) $l$-dimensional region in $R_a$.*

**Lemma 2.** *Let $\hat{R}_a$ be an $n$-dimensional inverse diagonal mesh partition with the variables $(t_1, \ldots, t_{n-1}, t^*)$, then the projection $R'_a$ of $\hat{R}_a$ on the subspace $\mathbb{R}^{n-1}_+$ that corresponds to the variables $(t_1, \ldots, t_{n-1})$ is $(n-1)$-dimensional diagonal mesh partition of width $a$.*

Our interest in diagonal and inverse diagonal mesh partitions is justified by the following example. Let us revisit the GSMP $A_s$ from Example 1. In Section 3.1 we have computed $p_{\pi_1}$ and $f^{\pi_1}(t_1, t_2, t_3)$ and showed that $f^{\pi_1}$ had its support on cubes in $\mathbb{R}^3_+$. Now we want to show that it is necessary to have diagonal constraints too. Consider the history $\pi_2$, which is a successor of $\pi_1$: $\pi_2 = \pi_1 \xrightarrow{\{e_3\}} (q_5, \emptyset, \emptyset)$. We want to compute $p_{\pi_2}$. Similarly to the formula for $\tilde{f}^{\pi_1}(t_2)$, we can write $\tilde{f}^{\pi_2}(t_1, t_2) = \int_0^{+\infty} f^{\pi_1}(t_1 + t_3, t_1 + t_3, t_3)\, dt_3$. Evaluating this formula using, for example, MAPLE we will see that in two regions $(0 < t_1 < 1, 0 < t_2 < 1, t_1 < t_2)$ and $(0 < t_1 < 1, 0 < t_2 < 1, t_1 > t_2)$, $\tilde{f}^{\pi_2}(t_1, t_2)$ is given by two *different* expolynomial expressions.

### 4.2 Expolynomial Expressions, Functions, and Distributions

We say that $e(x_1, \ldots, x_n)$ is an expolynomial expression if it is of the form $\sum_{k=1}^{r} c_k x_1^{m_{k1}} \cdots x_n^{m_{kn}} e^{\lambda_{k1} x_{k1} + \cdots + \lambda_{kn} x_{kn}}$, where $c_k, \lambda_{k1}, \ldots, \lambda_{kn} \in \mathbb{R}$, $m_{k1}, \ldots, m_{kn} \in \mathbb{N}_0$ for all $k = 1, \ldots, r$. By $Expr(x_1, \ldots, x_n)$ we denote the class of all expolynomial expressions in the variables $x_1, \ldots, x_n$.

A function $f_a(x_1, \ldots, x_n)$ is a *multidimensional expolynomial function* of width $a$ with finite support on a diagonal mesh partition $R_a$ if there exists a map $M_f : R_a \to Expr(x_1, \ldots, x_n)$ such that if a point $(x_1, \ldots, x_n) \in r$, $r$ is a region in $R_a$, then $f(x_1, \ldots, x_n) = M_f(r)(x_1, \ldots, x_n)$.

Given an expolynomial function $f(\bar{x}) = f(x_1, \ldots, x_n)$ and an $m$-dimensional region $r \in R_a$, $1 \leq m \leq n$, we want to define the integral of $f$ on $r$ (denoted as $\int_r f$). It is easy to see that due to the region's constraints, each point in $r$ can be determined by only $m$ independent parameters $\bar{y} = (y_1, \ldots, y_m)$, and we can express $\bar{x}$ as a function of $\bar{y}$, i.e. $x_i = x_i(\bar{y})$ for $i = 1, \ldots, n$. Thus we can define $\int_r f$ as a multiple integral $\int_{(x_1(\bar{y}),\ldots,x_n(\bar{y}))\in r} f(\bar{y})\, d\bar{y}$ taken over $m$ variables.

**Definition 4.** *Multidimensional random variable $\bar{X} = (X_1, \ldots, X_n)$ has an* expolynomial region distribution *(ERD) of width $a$ if there exists an expolynomial function $f_a(\bar{x}) = f_a(x_1, \ldots, x_n) \geq 0$ on $R_a$ such that for all $\bar{t} = (t_1, \ldots, t_n) \in \mathbb{R}^n_+$, $\Pr(\bar{X} < \bar{t}) = \Pr(X_1 < t_1, \ldots, X_n < t_n) = \sum_{r\in I_{R_a}} \int_{r\cap(x_1(\bar{y})<t_1,\ldots,x_n(\bar{y})<t_n)} f_a(\bar{y})\, d\bar{y} + \sum_{\substack{(x_1,\ldots,x_n)\in P_{R_a} \\ x_1<t_1,\ldots,x_n<t_n}} f(x_1, \ldots, x_n)$, where $I_{R_a}$ is the set of all regions of dimension one or higher in $R_a$, and $P_{R_a}$ is the set of all zero-dimensional regions (points).*

We call $f_a(\bar{x})$ the pdf of $\bar{X}$. Note, that for every region $r \in R_a$, $\Pr(\bar{X} \in r) = \int_r f$.

Let us give a simple example. Consider two one-dimensional independent random variables with ERDs given by their density functions:

$$f_1(x) = \begin{cases} 0, \text{ if } x = 0 \text{ or } x = 1 \\ 1, \text{ if } 0 < x < 1 \end{cases}, \qquad f_2(y) = \begin{cases} 0, & \text{if } y = 0 \\ 1/2, & \text{if } 0 < y < 1 \text{ or } y = 1 \end{cases}.$$

The first random variable $X$ is uniformly distributed on $(0, 1)$. The second random variable $Y$ is uniformly distributed on $(0, 1)$ and has a mass point at $y = 1$. Then the random variable $Z = XY$ is a two-dimensional random variable with the pdf

$$f_3(x, y) = \begin{cases} 0, & \text{if } (x = i, y = j) \text{ or } (x = i, 0 < y < 1), i, j = 0, 1 \\ 1/2, & \text{if } (0 < x < 1, 0 < y < 1, x \sim y), {\sim} \in \{<, >\} \\ 1/2, & \text{if } (y = 1, 0 < x < 1) \\ 0, & \text{if } (0 < x < 1, 0 < y < 1, x = y) \text{ or } (y = 0, 0 < x < 1) \end{cases}$$

We see that $f_3(x, y)$ is not zero in both full dimensional regions and in one mass region $(y = 1, 0 < x < 1)$.

## 5 Image Computation

In this section we will prove our main technical result.

**Theorem 1.** *Let $A$ be a GSMP, such that the firing time distributions of all events are ERDs of width $a$. Let $\pi \in \Pi$ be a history of $A$, $f^\pi$ be the pdf of the ecv of $\pi$, $\pi' = \pi \xrightarrow{E^*} (q_{l'}, \mathcal{O}_{l'}, \mathcal{N}_{l'})$ be a successor of $\pi$, $f^{\pi'}$ be the pdf of the ecv of $\pi'$, and $m'$ be the number of active events in $q_{l'}$, then:*

1. *$f^{\pi'}$ is an $m'$-dimensional ERD of width $a$;*

2. *given $f^{\pi}$, $f^{\pi'}$ can be computed symbolically.*

The theorem will follow from the steps described below.

To simplify complex expressions we will use a convenient shorthand notation. Suppose that $B = \{b_1, \ldots, b_q\}$ is a set of indices, then instead of writing $f(x_{b_1}, \ldots, x_{b_q}, y)$ we will write $f_{b\in B}(x_b, y)$. We will also slightly abuse notation by writing $f_{b\in B}(x_b + z, y)$ (where $z$ is a variable) instead of writing $f_{b\in B}(\hat{x}_b, y)$, $\hat{x}_b = x_b + z$.

Suppose a non-negative $n$-dimensional random variable $X$ with pdf $f(\bar{x})$ is divided into two random variables $X_1$ and $X_2$, such that $X_1 \in \mathbb{R}_+^s$ and $X_2 \in \mathbb{R}_+^{n-s}$. Then the pdf of $X_1$ is $f_{X_1}(\bar{x}_1) = \int_0^{+\infty} \cdots \int_0^{+\infty} f(y_1, \ldots, y_s, y_{s+1}, \ldots, y_n)\, dy_{s+1} \cdots dy_n$. The function $f_{X_1}(\bar{x}_1)$ is called a *marginal pdf* of $X$.

*Analysis of $f^{\pi'}$:* Notice that $f^{\pi'}$ can be written as

$$f^{\pi'}_{e\in\mathcal{O}_{l'}\cup\mathcal{N}_{l'}}(t_e) = \check{f}^{\pi'}_{e\in\mathcal{O}_{l'}}(t_e) \prod_{e\in\mathcal{N}_{l'}} dens(e)(t_e),$$

where $\check{f}^{\pi'}_{e\in\mathcal{O}_{l'}}(t_e)$ is the joint density function of the clock values of the events in $\mathcal{O}_{l'}$. Thus, obtaining $\check{f}^{\pi'}$ is sufficient for the construction of $f^{\pi'}$. It is also easy to see that if $\check{f}^{\pi'}$ is an expolynomial function of width $a$, then $f^{\pi'}$ is also an expolynomial function of width $a$ (but of a higher dimension).

*Computation of $\tilde{f}^{\pi,e^*}$:* Let us pick any event $e^* \in E^*$ and let $(q_l, \mathcal{O}_l, \mathcal{N}_l)$ be the last tuple of $\pi$. Suppose that $A$ has followed $\pi$ and now is in $q_l$. Let $G = \{t_e \geq t_{e^*} \mid e \in (\mathcal{O}_l \cup \mathcal{N}_l) \setminus \{e^*\}\}$ be a probabilistic event that the clock of the event $e^*$ expires before or simultaneously with the other clocks and $\Pr(G)$ be its probability. Then let $\tilde{f}^{\pi,e^*}_{e\in(\mathcal{O}_l\cup\mathcal{N}_l)\setminus\{e^*\}}(\hat{t}_e)$ be the pdf of clock values of all events in $(\mathcal{O}_l \cup \mathcal{N}_l) \setminus \{e^*\}$ at the moment when $t_{e^*} = 0$, conditioned on occurrence of $G$.

Let $\hat{t}_e = t_e - t_{e^*}$ and define

$$g_{e\in(\mathcal{O}_l\cup\mathcal{N}_l)\setminus\{e^*\}}(\hat{t}_e, t_{e^*}) = f^{\pi}_{e\in(\mathcal{O}_l\cup\mathcal{N}_l)\setminus\{e^*\}}(\hat{t}_e + t_{e^*}, t_{e^*}). \tag{1}$$

Then $g$ can be seen as the joint density function of $t_{e^*}$ and the *differences* between the values of the other event clocks and $t_{e^*}$ (these differences may be positive or negative).

Now let $g'_{e\in(\mathcal{O}_l\cup\mathcal{N}_l)\setminus\{e^*\}}(\hat{t}_e) = \int_0^{+\infty} g_{e\in(\mathcal{O}_l\cup\mathcal{N}_l)\setminus\{e^*\}}(\hat{t}_e, t_{e^*})\, dt_{e^*}$ be a marginal pdf. Then, given the definition of $G$ that states that all differences $\hat{t}_e$ should be non-negative we obtain that

$$\tilde{f}^{\pi,e^*}_{e\in(\mathcal{O}_l\cup\mathcal{N}_l)\setminus\{e^*\}}(\hat{t}_e) = \frac{g'_{e\in(\mathcal{O}_l\cup\mathcal{N}_l)\setminus\{e^*\}}(\hat{t}_e)}{\Pr(G)} = \frac{\int_0^{+\infty} g_{e\in(\mathcal{O}_l\cup\mathcal{N}_l)\setminus\{e^*\}}(\hat{t}_e, t_{e^*})\, dt_{e^*}}{\Pr(G)} \tag{2}$$

If we know how to compute $g'$, it is easy to compute $\Pr(G)$. Since $\tilde{f}^{\pi,e^*}$ is a pdf then if we integrate over all its variables we should obtain 1. Hence, from (2):

$$\Pr(G) = \int_0^{+\infty} \cdots \int_0^{+\infty} g'_{e_i\in(\mathcal{O}_l\cup\mathcal{N}_l)\setminus\{e^*\}}(\hat{t}_{e_i})\, d\hat{t}_{e_1} \cdots d\hat{t}_{e_{m-1}},$$

where $m$ is the number of active events in $q_l$.

It remains to show how, given $f^{\pi}_{e\in(\mathcal{O}_l\cup\mathcal{N}_l)}(t_e)$, we can compute $g'$ and to examine properties of this computation. Let us introduce a new variable $\hat{t}_{e^*} = t_{e^*}$, then from (1) we see that to compute $g$ from $f^{\pi}$ we need to apply the linear transformation $\mathcal{L}$ from Section 4.1. The expolynomial expressions are closed under linear transformations, and we also saw that diagonal mesh partitions are transformed into inverse diagonal mesh partitions of the same width (Lemma 1). So we conclude that $g$ is an expolynomial function on an inverse diagonal partition $\hat{R}^g_a$.

Now we have to obtain $g'$ from $g$. First, notice that by Lemma 2, $g'$ is defined on a diagonal partition $R^{g'}_a$ of the dimension one less than $\hat{R}^g_a$ and of the same width $a$. As in the example of Section 3.1, at each region $r$ it is given as a sum of integrals of expolynomial expressions of regions of $\hat{R}^g_a$ that are projected on $r$. These integrals can be computed symbolically using the formula $\int Dx^m e^{cx}\,dx = D(\frac{1}{c}x^m e^{cx} - \frac{m}{c}\int x^{m-1}e^{cx}\,dx)$, which can be easily derived by applying the integration by parts method. Thus, $g'$ (and therefore $\tilde{f}^{\pi,e^*}$) are computable expolynomial functions of width $a$.

*Computation of $\check{f}^{\pi'}$:* First, we "integrate out" of $\tilde{f}^{\pi,e^*}$ all clocks that were cancelled upon transition from $q_l$ to $q_{l'}$:

$$\tilde{\tilde{f}}^{\pi,e^*}_{e\in\mathcal{O}_{l'}}(t_e) = \int_0^{+\infty}\cdots\int_0^{+\infty} \tilde{f}^{\pi,e^*}_{e\in(\mathcal{O}_l\cup\mathcal{N}_l)\setminus\{e^*\}}(t_e)\,dt_{e_1}\cdots dt_{e_s},$$

where $e_1,\ldots,e_s \in E_{\text{cancelled}}(q,E^*,q')$, thus $\tilde{\tilde{f}}^{\pi}$ is a marginal pdf, and it easy to check that it is also an expolynomial function of width $a$.

We are almost done. It is left to extract from $\tilde{\tilde{f}}^{\pi,e^*}$ information that is pertinent only to the transition that was caused by firing of the events $E^*$ and not to the transitions that are triggered by the sets of events that properly contain $E^*$. Let $\check{E}^* = E^*\setminus\{e^*\}$, then we construct $\check{f}^{\pi\prime}$ from $\tilde{\tilde{f}}^{\pi,e^*}$ by extracting exactly those regions that have a constraint of the form $t_e = 0$ if and only if $e \in \check{E}^*$. For example, if $e^*$ is the only event in $E^*$, then $\check{E}^* = \emptyset$ and we obtain $\check{f}^{\pi\prime}$ from $\tilde{\tilde{f}}^{\pi,e^*}$ by setting to zero all regions that have a constraint $t_e = 0$ for any event $e$. Similarly, if $\check{E}^* = \{e_1\}$ then we extract all those regions that are defined by the constraint $t_{e_1} = 0$ (and set to zero the expolynomial expressions for regions that in addition to $t_{e_1} = 0$ have a constraint $t_{e'} = 0$ for any other event $e'$).

Note, that $\check{f}^{\pi\prime}$ constructed from $\tilde{\tilde{f}}^{\pi,e^*}$ may no longer be a pdf, so we have to divide it by a normalizing constant $0 < H < 1$, which is easily computable.

*Computation of $p_{\pi'|\pi}$:* As a consequence of our previous computations, we obtain the formula for $p_{\pi'|\pi}$:

$$p_{\pi'|\pi} = \Pr(G)\cdot H\cdot next(q_l,E^*,q_{l'}).$$

## 6 Illustrative Example

We are developing a tool called EHV (Event Horizon Verifier) that implements the algorithm of Section 5. The tool is written in JAVA and relies on JSCIENCE open source library for the symbolic computations.

As an application of our method we consider a queueing problem. The producer generates messages and the consumer processes them. The messages that await processing are stored in a buffer of capacity $K$ (initially the buffer is empty). The interarrival time between successive messages is modeled by ERD with the pdf: $f_1(t)$ is $p$ if $t \in (0,1)$, is $a(p) + b(p)x$ if $t \in (1,2)$, and 0 otherwise, where $p \in (0,1)$ is a parameter. With $f_1(t)$ we can model a situation when the interval between any two successive messages are at most two time units, the probability that a message arrives during the first time unit is uniform and the probability that a message would arrive during the second time unit is "skewed" towards the end of the interval.

The time that the consumer needs to process a message is uniformly distributed on $(0,1)$. It is also known that the producer can produce at most $N > K$ messages and we want to find the probability $P_{\text{overflow}}$ that the buffer exceeds its capacity.

Notice, that if the difference between $N$ and $K$ is small, then $P_{\text{overflow}}$ can also be very small. Simulation techniques to estimate small probabilities are involved, they require a large number of simulations and give only statistical guarantees. To the contrary, the running time of our method does not depend on the absolute value of $P_{\text{overflow}}$, and, in fact, performance improves if there are only a few paths that lead to the unsafe locations.

The problem can be reduced to the bounded until problem for the GSMP $B$ defined as follows. The locations of $B$ are encoded with pairs $(k,n)$, where $k = 0,\ldots,K+1$ is the number of messages in the buffer and $n = 0,\ldots,N$ is the total number of messages received so far. The location $(0,0)$ is the initial location. For any $n$, the locations $(K+1,n)$ are "unsafe", and all locations $(k',n')$, such that $N-n' \leq K-k'$ are destinations (if $B$ is in such a location then the buffer cannot overflow). $B$ has two events $e_p$ and $e_c$. For all $n = 0,\ldots,N-1$, the locations $(0,n)$ have $e_p$ as their only active event and upon firing of that event $B$ moves to the location $(1,n+1)$. Unsafe and destination locations are absorbing, and all the other locations have both $e_p$ and $e_c$ as their active events. When $B$ is in such a location $(i,j)$, firing of $e_p$ or $e_c$ causes a transition to $(i+1,j+1)$ or $(i-1,j)$, respectively.

We performed experiments for some sets of parameters. The computer that we used for our experiments was a Linux server equipped with dual Pentium III processors operating at 1400 MHz and with 2 GB of RAM. For each set of parameters we analyzed all histories in $\Pi_{\text{until}}$ (the total number of them is in the "Dest. reached" column) and all histories that end in an unsafe location (the "Unsafe reached" column). Below is the summary of results.

| **Parameter values:** $(\mathbf{K},\mathbf{N}),\mathbf{p}$ | **Results** | | | |
|---|---|---|---|---|
| | $P_{\text{overflow}}$ | Running time | Dest. reached | Unsafe reached |
| $(5,11),1/2$ | $9.0897 \times 10^{-4}$ | 1 min. 23 sec. | 2380 | 1040 |
| $(7,11),1/2$ | $7.5504 \times 10^{-6}$ | 36 sec. | 560 | 185 |
| $(7,11),1/5$ | $4.1124 \times 10^{-9}$ | 3 min. 7 sec. | 560 | 185 |
| $(7,11),1/10$ | $1.9335 \times 10^{-11}$ | 3 min. 17 sec. | 560 | 185 |
| $(30,31),1/10$ | $1.2161 \times 10^{-64}$ | 23 sec. | 30 | 1 |

## References


1. 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.
2. R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.
3. A. Aziz, K. Sanwal, V. Singhal, and R.K. Brayton. Model-checking continuous-time markov chains. *ACM Transactions on Computational Logic*, 1(1):162–170, 2000.
4. A. Biere, A. Cimatti, E. Clarke, M. Fujita, and Y. Zhu. Symbolic model checking using SAT procedures instead of BDDs. In *Proceedings of the 36th ACM/IEEE Design Automation Conference*, pages 317–320, 1999.
5. E.M. Clarke, O. Grumberg, and D.A. Peled. *Model checking*. MIT Press, 2000.
6. E.M. Clarke and R.P. Kurshan. Computer-aided verification. *IEEE Spectrum*, 33(6):61–67, 1996.
7. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.
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. H. Hansson and B. Jonsson. A framework for reasoning about time and reliability. In *Proceedings of the Tenth IEEE Real-Time Systems Symposium*, pages 102–111, 1989.
11. B. Haverkort. *Performance of computer-communication systems: A model-based approach*. Wiley & Sons, 1998.
12. G.J. Holzmann. The model checker SPIN. *IEEE Transactions on Software Engineering*, 23(5):279–295, 1997.
13. M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic symbolic model checker. In *Proceedings of Computer Performance Evaluation/Tools 2002*, LNCS 2324, pages 200–204, 2002.
14. M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Verifying quantitative properties of continuous probabilistic timed automata. In *Proceedings of the 11th International Conference on Concurrency Theory*, LNCS 1877, pages 123–137. Springer, 2000.
15. M.Z. Kwiatkowska. Model checking for probability and time: from theory to pratice. In *Proceedings of the 18th IEEE Symposium on Logic in Computer Science*, pages 351–360, 2003.
16. M.Z. Kwiatkowska, G. Norman, and D. Parker. Probabilistic symbolic model checking with PRISM: a hybrid approach. *Software Tools for Technology Transfer*, 6(2):128–142, 2004.
17. P.D'Argenio, H. Hermanns, J.-P. Katoen, and R. Klaren. Modest - a modeling and description language for stochastic timed systems. In *Proceedings of the PAPM-PROBMIV Joint International Workshop*, LNCS 2165, pages 87–104. Springer, 2001.
18. G.S. Shedler. *Regenerative stochastic simulation*. Academic Press, 1993.
19. M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proceedings of the 26th IEEE Symposium on Foundations of Computer Science*, pages 327–338, 1985.
20. H. Younes and R. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *Computer Aided Verification, 14th International Conference*, LNCS 2404, pages 223–235, 2002.