Probabilistic Model Checking of Regenerative Concurrent Systems

Probabilistic Model Checking of Regenerative Concurrent Systems

Marco Paolieri, András Horváth, and Enrico Vicario, *Member, IEEE Computer Society*

Abstract—We consider the problem of verifying quantitative reachability properties in stochastic models of concurrent activities with generally distributed durations. Models are specified as Stochastic Time Petri Nets (STPNs) and checked against Boolean combinations of *interval until operators* imposing bounds on the probability that the marking process will satisfy a goal condition at some time in the interval $[\alpha, \beta]$ after an execution that never violates a safety property. The proposed solution is based on the analysis of *regeneration points* in the model execution: a regeneration is encountered after a discrete event if the future evolution depends only on the current marking and not on its previous history, thus satisfying the Markov property. We analyze systems in which multiple generally distributed timers can be started or stopped independently, but regeneration points are always encountered with probability 1 after a bounded number of discrete events. Leveraging the properties of regeneration points in probability spaces of execution paths, we show that the problem can be reduced to a set of Volterra integral equations, and we provide algorithms to compute their parameters through the enumeration of finite sequences of *stochastic state classes* encoding the joint probability density function (PDF) of generally distributed timers after each discrete event. The computation of symbolic PDFs is limited to the first regeneration epoch, and the repetitive structure of the stochastic process is exploited also before the lower bound α, providing crucial benefits for large time bounds. A case study is presented through the probabilistic formulation of Fischer's mutual exclusion protocol, a well-known real-time verification benchmark.

Index Terms—Probabilistic Model Checking, Reachability, Stochastic Petri Net, Markov Regenerative Process, Markov Renewal Theory.

1 Introduction

In the engineering of non-functional requirements, verification of quantitative properties of stochastic models enables early assessment of design choices and provides model-driven guidance for implementation and integration stages. This becomes particularly valuable in the development of systems where the effects of concurrency are intertwined with probabilistic behavior and stochastic durations.

Probabilistic model checking supports a systematic practice through which the same model can be verified against multiple probabilistic properties, specified in some well-defined language, and open to automated regression verification when the model evolves. Empirical evidence indicates that most probabilistic requirements occurring in industrial practice can be specified through a limited set of specification patterns [1]. In particular, most *transient* properties can be reduced to the form of the probabilistic *interval until operator*, which specifies a bound on the probability that the model will be in a goal state at some time in the interval $[\alpha, \beta]$ after having visited only legal states. A large body of techniques and tools have been developed and demonstrated on industrial case studies following either a statistical [2], [3], [4], [5] or numerical [6], [7], [8], [9] approach. In particular, numerical probabilistic model checking aims at computing results with high accuracy and confidence through exhaustive state-space analysis [10], often at the expense of a restriction on the class of models amenable to verification.

Most numerical approaches addressed the verification of real-time properties over models of concurrent activities with *exponentially distributed* (EXP) durations. In this case, due to the memoryless property of the exponential distribution, the underlying stochastic process of the model satisfies the *Markov property* at each time instant, i.e., the current state provides sufficient information to predict future evolution, regardless of elapsed sojourn times. Efficient algorithms for the analysis of continuous-time Markov chains (CTMCs) can thus be applied to the verification of requirements specified in continuous stochastic logic (CSL), also allowing nesting of temporal operators [11], [12], [13], [9], [14]. Properties specified as deterministic timed automata can also be verified through the analysis of piecewise-deterministic Markov processes resulting from the synchronous composition of the model with the specification automaton [15], [16].

However, for valid modeling of several classes of systems, some durations must be represented as *generally distributed* (GEN) random variables, i.e., variables that break the limit of memoryless EXPs. Notable examples include aging processes, which structurally depend on some age variables, or real-time systems, in which correctness depends on firm time bounds resulting from Worst-Case Execution Times, minimum intervals between releases, deterministic duration of synchronous periods, timeouts and watchdogs. In this case, the future evolution of the model depends on random *timers* encoding the remaining time of GEN durations. The underlying stochastic process is no more a CTMC and rather

- *M. Paolieri and E. Vicario are with the Department of Information Engineering, Università di Firenze, Via di Santa Marta 3, 50139 Firenze, Italy. E-mail:* {marco.paolieri, enrico.vicario}@unifi.it.
- *A. Horváth is with the Department of Computer Science, Università di Torino, Corso Svizzera 185, 10149 Torino, Italy. E-mail:* horvath@di.unito.it.

Manuscript submitted to the IEEE Transactions on Software Engineering.

belongs to some wider class, which is determined by the persistence of GEN timers after discrete events [17].

If GEN timers cannot persist to any discrete event, the underlying stochastic process is semi-Markov (SMP): the system has "memory" of the sojourn time elapsed in the current state (i.e., its future evolution depends on this continuous variable, in addition to the discrete state), but the Markov property is satisfied immediately after each discrete event. In the terminology of Markov renewal theory [18], the corresponding time instants are called *regeneration points* and, by a *renewal argument* [18], execution paths reaching some goal state within time β can be decomposed into a prefix that reaches the first regeneration at some time $t \leq \beta$ and an independent suffix that reaches the goal state within the remaining time $\beta - t$. In [19], verification of a *probabilistic interval until* with $\alpha = 0$ is reduced for SMPs to a first-passage analysis problem by making illegal and goal states absorbing (i.e., disabling any timer). Following a different approach, [20] verifies non-nested until operators on stochastic automata with underlying SMPs by unfolding the set of model behaviors into a tree where nodes impose distinct pairwise orderings on timers; a fragment of the tree is analyzed to compute the probability of reaching each node and to decide a formula with $\alpha \neq 0$, but regenerations are not exploited and complexity is exponential with respect the number of events feasible within β. As a common limitation, the SMP assumption that GEN timers are disabled or reset after each discrete event is still a severe limit for model validity, as it rules out important design mechanisms such as timeouts that span across multiple activities, or asynchronous components that start their timers independently.

When GEN timers can persist after the occurrence of discrete events, memory of elapsed sojourn times can be carried across multiple states, but regeneration points can still be encountered, delimiting *regeneration epochs*. If this happens infinitely often with probability 1 (w.p.1), the process is *Markov regenerative* (MRP) [17], [18]. A large literature has addressed numerical solution for models in the subclass of MRPs that satisfy the so-called *enabling restriction*, i.e., models where at most a single GEN timer is enabled in each state. In this case, GEN timers cannot overlap their intervals of activity, and bounded reachability can be evaluated through the analysis of the CTMCs subordinated to the activity interval of each GEN timer [21], [22], [23]. Nonetheless, the enabling restriction imposes a severe limitation as well, as it rules out models with multiple concurrent GEN timers (e.g., a timeout over a GEN timer).

When regeneration points are not guaranteed to be reached infinitely often w.p.1, the underlying process belongs to the class of generalized semi-Markov processes (GSMPs) [17], [24], for which very few results of numerical evaluation were developed. Evaluation of (non-nested) probabilistic interval until operators with $\alpha = 0$ was proposed in [25] under the assumption of a bound on the number of discrete events feasible within time β; the solution is based on the enumeration of a tree of probability density functions (PDFs) of active timers over *regions* [26]. In [27], the assumption of a bound on the number of events within time β is removed and fairly general conditions that guarantee termination in exact or approximate evaluation are provided, through the enumeration of *stochastic state classes* [28], [29], [30] representing the joint PDF of timers over Difference Bounds Matrix zones [31]. While a calculus based on zones largely reduces the branching factor of enumerated trees, the complexity of [25] and [27] grows exponentially with the length of event sequences within time β, and no computational advantage is provided when the underlying stochastic process falls in the subclass of MRPs.

In this paper, we propose a technique for the verification of non-nested interval until operators in MRPs that always encounter a regeneration point or a conclusive state (i.e., an illegal state, or a goal state reached after α, or any state reached after β) w.p.1 within a bounded number of discrete events. This contribution extends the class of stochastic models amenable to numerical solution by including models that fall in the class of MRPs but do not satisfy the enabling restriction, without bounds on the number of events executed within time β, and with a major reduction of complexity when the stochastic process has a repetitive structure and multiple regenerations can be traversed before reaching the time limit β.

In Section 2, we introduce Stochastic Time Petri Nets (STPNs) for the specification of stochastic models. A probability measure on STPN execution paths is defined by establishing a formal relation between cylinder sets (i.e., sets of execution paths that follow a common qualitative ordering of events under given timing restrictions) and stochastic state classes. We provide an algorithm for the detection of regeneration points in the enumeration of stochastic state classes; in so doing, we generalize the state of the art in the literature on stochastic Petri nets [17] by encompassing the case of regenerations where all GEN timers have been enabled for a deterministic time (Section 3). Coming to the core result of the work, we then show that the verification of an interval until operator can be reduced to a set of Volterra integral equations of the second kind that extend generalized Markov renewal equations [18] into a bivariate form based on three kernels, and we provide algorithms to compute the kernels through the enumeration of stochastic state classes reachable before the first regeneration (Section 4). We finally illustrate how the approach extends the class of models amenable to numerical solution through the study of a probabilistic formulation of Fischer's mutual exclusion protocol (Section 5), and draw our conclusions on the verification of regenerative stochastic systems (Section 6).

2 Problem Definition

2.1 Stochastic Time Petri Nets

We specify concurrent systems with stochastic durations using stochastic time Petri nets (STPNs) [30], [28]. An STPN is defined by a set of *transitions*, representing activities with stochastic duration, and a set of *places*; a *marking* assigns a nonnegative number of *tokens* to each place. Places can serve as *input* or *output* places of a transition: when the marking assigns at least one token to each input place, the transition is enabled; after its firing, one token is removed from each input place and one token is added to each output place. STPNs have a natural graphical representation illustrated in Fig. 1a: transitions are drawn as bars, places as circles, and tokens as dots inside each place. Directed arcs connect input places to transitions, and transitions to output places.

The color of a transition represents the PDF type of its stochastic duration: white for exponential distributions, gray for deterministic durations, and black for other distributions (as a special case, a thin black bar is used for transitions with zero duration, see Fig. 8). Each transition samples a *time to fire* when it becomes enabled; as in discrete event systems, the transition with minimum time to fire is the next event and its firing enables, disables, or restarts other events by removing tokens from input places and adding tokens to output ones. Similarly to stochastic reward nets [32] or stochastic activity networks [33], the enabling of a transition can be limited using *enabling functions*, arbitrary constraints on token counts that are annotated next to the transition, after the symbol ?; *update functions* of the form *place* ← *expression* can specify additional updates of the token count of a place after the firing of the transition (see Fig. 8). In the following, we adopt STPNs to specify stochastic systems, although our techniques can be extended to other formalisms after defining a calculus for the computation of times to fire PDFs given a sequence of events (e.g., stochastic timed automata [34]). As a requirement, the underlying stochastic process must encounter regenerations w.p.1 after a bounded number of fired events, and the state space must be finite (e.g., for STPNs, the number of tokens accumulated in each place must be bounded).

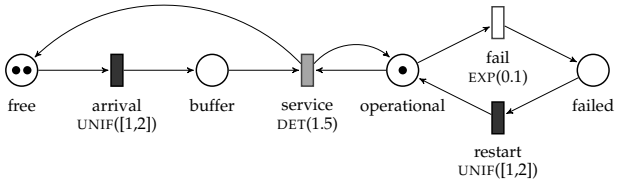
2.1.1 Syntax

Definition 1. An STPN is a tuple

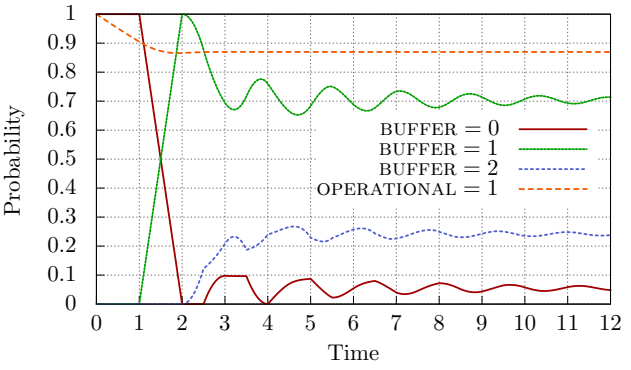
$$\langle P, T, A^-, A^+, B, U, EFT, LFT, F, W \rangle$$

where: P and T are disjoint sets of places and transitions; $A^- \subseteq P \times T$ and $A^+ \subseteq T \times P$ are the precondition and post-condition relations, respectively; B and U associate each transition $t \in T$ with an enabling function $B(t)\colon \mathbb{N}^P \to \{\text{TRUE}, \text{FALSE}\}$ and with an update function $U(t)\colon \mathbb{N}^P \to \mathbb{N}^P$. In addition, for each transition $t \in T$, the STPN specifies: an earliest firing time $EFT(t) \in \mathbb{Q}_{\geqslant 0}$, a latest firing time $LFT(t) \in \mathbb{Q}_{\geqslant 0} \cup \{\infty\}$ such that $EFT(t) \leq LFT(t)$, a cumulative distribution function (CDF) F_t such that $x < EFT(t) \Rightarrow F_t(x) = 0$ and $x > LFT(t) \Rightarrow F_t(x) = 1$, and a weight $W(t) \in \mathbb{R}_{>0}$.

A place p is said to be an *input* or *output* place for a transition t if $(p,t) \in A^-$ or $(t,p) \in A^+$, respectively. Following the usual terminology of stochastic Petri nets, a transition t is called *immediate* (IMM) if $EFT(t) = LFT(t) = 0$ and *timed* otherwise; a timed transition is called *exponential* (EXP) if $F_t(x) = 1 - e^{-\lambda x}$ for some rate $\lambda \in \mathbb{R}_{>0}$, or *general* (GEN) if its time to fire is distributed according to a non-exponential distribution; as a special case, a GEN transition t is *deterministic* (DET) if $EFT(t) = LFT(t) > 0$. For each transition t with $EFT(t) < LFT(t)$, we assume that F_t can be expressed as the integral function of a PDF f_t, i.e., $F_t(x) = \int_0^x f_t(y)\,dy$. The same notation is also adopted for IMM and DET transitions, which are associated with Dirac impulse functions $f_t(y) = \delta(y - \overline{y})$ with $\overline{y} = EFT(t) = LFT(t)$. In particular, we consider the class of *piecewise expolynomial* PDFs obtained by piecewise composition of products of exponentials and polynomials, on bounded or unbounded supports (also known in the literature as *exponomials* [14]).



(a) STPN model of the queue.



(b) Transient probabilities.

Fig. 1: G/D/1/2/2 queue with server breakdowns.

2.1.2 Semantics

Given an STPN $\langle P, T, A^-, A^+, B, U, EFT, LFT, F, W \rangle$, a marking $m \in \mathbb{N}^P$ assigns a natural number of tokens to each place of the net. A transition t is *enabled* by m if m assigns at least one token to each of its input places and the enabling function $B(t)(m)$ evaluates to TRUE; the set of transitions enabled by m is denoted as $E(m)$.

Definition 2. The state of an STPN is a pair $\langle m, \vec{\tau} \rangle$ where $m \in \mathbb{N}^P$ is a marking and $\vec{\tau} \in \mathbb{R}_{\geqslant 0}^{E(m)}$ assigns a *time to fire* to each enabled transition.

Given an initial state $s_0 = \langle m_0, \vec{\tau}_0 \rangle$, an execution of the STPN is represented by a (finite or infinite) *path*

$$\omega = s_0 \xrightarrow{\gamma_1} s_1 \xrightarrow{\gamma_2} s_2 \xrightarrow{\gamma_3} \cdots$$

where $\gamma_i \in T$ is the ith transition fired along the execution and $s_i = \langle m_i, \vec{\tau}_i \rangle$ is the state reached after the firing of γ_i. In each state s_i:

- The next transition γ_{i+1} is selected from the set of enabled transitions with minimum time to fire according to a discrete distribution given by weights: if $E_{\min} = \arg\min_{t \in E(m_i)} \vec{\tau}_i(t)$, then $t \in E_{\min}$ is selected with probability $p_t = W(t)/(\sum_{u \in E_{\min}} W(u))$.
- After the firing of γ_{i+1}, the new marking m_{i+1} is derived by (1) removing a token from each input place of γ_{i+1}, (2) adding a token to each output place of γ_{i+1}, and (3) applying the update function $U(\gamma_{i+1})$ to the resulting marking. A transition t enabled by m_{i+1} is said to be *persistent* if it is distinct from γ_{i+1}, and it is enabled also by m_i and by the intermediate markings after steps (1) and (2); otherwise, t is said to be *newly enabled*.
- For each newly enabled transition t, the time to fire $\vec{\tau}_{i+1}(t)$ is sampled according to the distribution F_t; for each persistent transition t, the time to fire in s_{i+1} is reduced by the sojourn time in the previous marking, i.e., $\vec{\tau}_{i+1}(t) = \vec{\tau}_i(t) - \vec{\tau}_i(\gamma_{i+1})$.

Example 1. Fig. 1a introduces a small running example inspired by [23] and representing a G/D/1/2/2 queue (1 server with DET service time, a population of 2 customers with GEN arrival times, a capacity of 2). Tokens in places *free* and *buffer* specify the number of customers in the idle state or inside the queue, respectively. Customers arrive in series after times uniformly distributed over $[1, 2]$ (transition *arrival*), while service has a deterministic duration of 1.5 (transition *service*) and requires the server to be *operational*; the time to failure of the server is exponentially distributed with rate 0.1 (transition *fail*), and repairs are completed in a time uniformly distributed over $[1, 2]$ (transition *restart*). Fig. 1b depicts transient probabilities for the number of customers in the buffer and for the state of the server.

2.2 Probability space and cylinder sets

In the formulation of the probabilistic model checking problem, we will need to refer to the probability measure of selected sets of paths of an STPN model. We thus formalize the concept by defining the probability space $(\Omega_{m_0}, \mathcal{F}_{m_0}, Pr_{m_0,f_{\vec{\tau}_0}})$ induced by the semantics of STPNs for a given initial marking m_0 and initial times to fire PDF $f_{\vec{\tau}_0}$. The outcomes Ω_{m_0} of the probability space are all paths $\omega = s_0 \xrightarrow{\gamma_1} s_1 \xrightarrow{\gamma_2} \cdots$ with initial marking m_0. Note that Ω_{m_0} may include paths that are not feasible under the initial distribution $f_{\vec{\tau}_0}$ due to temporal constraints of the model; for these paths, the probability measure $Pr_{m_0,f_{\vec{\tau}_0}}$ will be zero.

To identify a σ-algebra $\mathcal{F}_{m_0}$ of events over Ω_{m_0}, we define as *cylinder set* $C(m_0, \gamma_1, I_1, \gamma_2, I_2, \ldots, \gamma_n, I_n)$ the set of all paths with initial marking m_0 that fire the sequence of transitions $\gamma_1, \gamma_2, \ldots, \gamma_n$ at absolute times contained in the intervals $I_1, I_2, \ldots, I_n$, respectively:

$$C(m_0, \gamma_1, I_1, \gamma_2, I_2, \ldots, \gamma_n, I_n) := \{\, \omega \in \Omega_{m_0} \mid |\omega| \geq n \text{ and } \forall\, 0 < k \leq n.(\omega[k] = \gamma_k \text{ and } T(k,\omega) \in I_k) \,\}$$

where $|\omega|$ is the number of firings in ω, $\omega[k]$ is the kth transition fired along ω for all $0 < k \leq |\omega|$, and $T(k, \omega)$ is the absolute time of the kth firing in ω, for all k:

$$T(k,\omega) := \begin{cases} \sum_{i=0}^{k-1} \min_{t \in E(m_i)} \vec{\tau}_i(t) & \text{if } k \leq |\omega|, \\ +\infty & \text{if } k > |\omega|. \end{cases}$$

Note that, in contrast with the usual definition of cylinder sets for CTMCs [11], [12], constraints refer to absolute firing times rather than sojourn times. This formulation enables a simpler treatment of the dependence among subsequent sojourn times in models with underlying stochastic processes more general than SMPs.

The set of events $\mathcal{F}_{m_0}$ is defined as the smallest σ-algebra on Ω_{m_0} that contains all the cylinder sets $C(m_0, \gamma_1, I_1, \gamma_2, I_2, \ldots, \gamma_n, I_n)$ for $n \in \mathbb{N}$, $\gamma_1, \gamma_2, \ldots, \gamma_n$ ranging over all sequences of n transitions in T, and $I_1, I_2, \ldots, I_n$ ranging over all sequences of n non-empty intervals with rational endpoints (possibly right-unbounded).

Proposition 1. *$\mathcal{F}_{m_0}$ is countable, the intersection of cylinder sets is a cylinder set, and the complement of a cylinder set is a finite union of disjoint cylinder sets.*

Proof. The elements of $\mathcal{F}_{m_0}$ are uniquely identified by finite strings alternating transitions and firing intervals with rational endpoints: pairs of rational numbers and finite strings from a finite alphabet are countable sets, and thus $\mathcal{F}_{m_0}$ is also countable. The intersection of the cylinder sets $C(m_0, \gamma_1, I_1, \gamma_2, I_2, \ldots, \gamma_n, I_n)$ and $C(m_0, \gamma'_1, I'_1, \gamma'_2, I'_2, \ldots, \gamma'_m, I'_m)$ with $n \leq m$ is non-empty only if $\gamma_i = \gamma'_i$ for $i = 1, \ldots, n$ and it corresponds to the cylinder set $C(m_0, \gamma_1, I_1 \cap I'_1, \gamma_2, I_2 \cap I'_2, \ldots, \gamma_n, I_n \cap I'_n, \gamma'_{n+1}, I_{n+1}, \ldots, \gamma'_m, I'_m)$. Finally, the complement of a cylinder set $C(m_0, \gamma_1, I_1, \gamma_2, I_2, \ldots, \gamma_n, I_n)$ corresponds to the finite union of all the cylinder sets $C(m_0, \gamma'_1, I'_1, \gamma'_2, I'_2, \ldots, \gamma'_n, I'_n)$ such that either (1) $\exists i \leq n.(\gamma_i \neq \gamma'_i)$ and $I'_i = [0, \infty)$, or (2) $\forall i \leq n.(\gamma_i = \gamma'_i)$ and $\exists i \leq n.(I'_i = [0, \inf I_i] \vee I'_i = [\sup I_i, \infty))$. □

The probability measure $Pr_{m_0,f_{\vec{\tau}_0}}$ can be expressed in terms of *transient stochastic state classes* [35] and, if GEN transitions are associated with piecewise expolynomial PDFs, it can be computed numerically using the Sirio package of the ORIS Tool [36].

Definition 3. A *transient stochastic state class* is a tuple $\Sigma = \langle m, D, f_{\langle \tau_{age}, \vec{\tau} \rangle} \rangle$ where: $m \in \mathbb{N}^P$ is a marking; $f_{\langle \tau_{age}, \vec{\tau} \rangle}$ is the PDF (immediately after a firing) of the random vector $\langle \tau_{age}, \vec{\tau} \rangle$ including the absolute time τ_{age} and the times to fire $\vec{\tau} = (\tau_1, \ldots, \tau_n)$ of transitions $E(m) = \{t_1, \ldots, t_n\}$ enabled by m; $D \subseteq \mathbb{R}^{n+1}$ is the support of $f_{\langle \tau_{age}, \vec{\tau} \rangle}$.

The initial (unconditioned) transient stochastic state class Σ_0 assigns non-null probability to states with marking m_0 and absolute time $\tau_{age} = 0$ according to the times to fire PDF $f_{\vec{\tau}_0}$, i.e., $\Sigma_0 = \langle m_0, D_0, f_{\langle \tau_{age}, \vec{\tau}_0 \rangle} \rangle$, where $D_0 = [0, 0] \times [\text{support of } f_{\vec{\tau}_0}]$ and $f_{\langle \tau_{age}, \vec{\tau}_0 \rangle}(x_{age}, x_1, \ldots, x_n) = \delta(x_{age}) \cdot f_{\vec{\tau}_0}(x_1, \ldots, x_n)$ for $E(m_0) = \{t_1, \ldots, t_n\}$. Given a class Σ, the state PDF conditioned on the execution of a transition γ at an absolute time in the interval I is given by the *successor class* of Σ through γ and I.

Definition 4. We say that $\Sigma' = \langle m', D', f'_{\langle \tau_{age}, \vec{\tau} \rangle} \rangle$ is the *successor* of $\Sigma = \langle m, D, f_{\langle \tau_{age}, \vec{\tau} \rangle} \rangle$ through $\gamma \in T$ at some time in I and with succession probability μ, and we write $\Sigma \xRightarrow{\gamma, I, \mu} \Sigma'$, if, given that the marking of the STPN is m and $\langle \tau_{age}, \vec{\tau} \rangle$ is a random vector distributed over D according to $f_{\langle \tau_{age}, \vec{\tau} \rangle}$, then: (1) γ has non-null probability μ of firing in Σ at some time in I, and (2) if γ fires in Σ at some time in I, its firing yields the marking m' and, conditioned on this event, the times to fire and τ_{age} after the firing are distributed over D' according to $f'_{\langle \tau_{age}, \vec{\tau} \rangle}$.

The relation $\xRightarrow{\gamma, I, \mu}$ can be enumerated through a calculus for the computation of the probability of outgoing events, and for the symbolic derivation of the support and closed-form PDF of $\langle \tau_{age}, \vec{\tau} \rangle$ in successor classes. Given a class Σ, an enabled transition γ and a firing interval I, this calculus computes the succession probability μ and the (unique) successor class Σ' such that $\Sigma \xRightarrow{\gamma, I, \mu} \Sigma'$. Each class encodes the PDF of the current state given the past history, providing a full characterization of the future evolution. The treatment in this paper does not assume the knowledge of this calculus; for convenience, we provide a summary in the additional material, and refer to [29], [30], [35] for a comprehensive description.

The probability measure of cylinder sets can be evaluated through the repeated computation of successor classes.

Proposition 2. *Given an initial marking m_0 and PDF $f_{\vec{\tau}_0}$ of the times to fire, the probability measure of a cylinder set $C(m_0, \gamma_1, I_1, \gamma_2, I_2, \ldots, \gamma_n, I_n)$ according to $Pr_{m_0, f_{\vec{\tau}_0}}$ is equal to the product $\prod_{i=1}^{n} \mu_i$ of succession probabilities for the sequence of transient stochastic state classes*

$$\Sigma_0 \xrightarrow{\gamma_1, I_1, \mu_1} \Sigma_1 \xrightarrow{\gamma_2, I_2, \mu_2} \cdots \xrightarrow{\gamma_n, I_n, \mu_n} \Sigma_n,$$

or equal to 0 if the sequence is not defined (i.e., $\exists i \leq n . \mu_i = 0$).

Proof. The event $C(m_0, \gamma_1, I_1, \gamma_2, I_2, \ldots, \gamma_n, I_n)$ for $n \in \mathbb{N}$ can be expressed as $E_0 \cap E_1 \cap \cdots \cap E_n$, where $E_0 = \Omega_{m_0}$ and, for each $i > 0$,

$$E_i = \{\, \omega \in \Omega_{m_0} \mid |\omega| \geq i, \omega[i] = \gamma_i \text{ and } T(i, \omega) \in I_i \,\}$$

is the constrained set of paths imposing an absolute time only on the ith transition. By induction on the definition of succession of stochastic state classes, for all $i \leq n$, the class Σ_i in

$$\Sigma_0 \xrightarrow{\gamma_1, I_1, \mu_1} \Sigma_1 \xrightarrow{\gamma_2, I_2, \mu_2} \cdots \xrightarrow{\gamma_n, I_n, \mu_n} \Sigma_n$$

represents the joint PDF of the absolute time and current state given the events $E_0, E_1, \ldots, E_i$, and μ_i is the probability $Pr_{m_0, f_{\vec{\tau}_0}}\{E_i \mid E_0, E_1, \ldots, E_{i-1}\}$ that paths that performed $\gamma_1, \ldots, \gamma_{i-1}$ in $I_1, \ldots, I_{i-1}$ will also perform γ_i at some time in I_i. Then $Pr_{m_0, f_{\vec{\tau}_0}}\{E_0 \cap E_1 \cap \cdots \cap E_n\} = Pr_{m_0, f_{\vec{\tau}_0}}\{E_0\} \cdot Pr_{m_0, f_{\vec{\tau}_0}}\{E_1 \mid E_0\} \cdots Pr_{m_0, f_{\vec{\tau}_0}}\{E_n \mid E_0, E_1, \ldots, E_{n-1}\} = \prod_{i=1}^{n} \mu_i$ if $\mu_i > 0$ for all $i \leq n$; if some event E_i has null probability given $E_0, E_1, \ldots, E_{i-1}$ (i.e., Σ_{i-1} has no successor through γ_i at some time in I_i and thus $\mu_i = 0$), the measure of the cylinder set is zero. □

2.3 Probabilistic Temporal Logic

We specify quantitative properties of STPNs with a probabilistic temporal logic based on an *interval until operator* with predicates over the markings of the net. The logic can express bounds on the probability that the marking of the STPN satisfies a goal predicate φ_2 at some time in the interval $[\alpha, \beta]$ without violating a safety predicate φ_1. The syntax of the logic is

$$\psi ::= \text{TRUE} \mid \neg\psi \mid \psi \wedge \psi \mid P_{\sim p}[\, \varphi \, \mathcal{U}^{[\alpha,\beta]} \varphi \,]$$
$$\varphi ::= \text{TRUE} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \text{AP}$$

where $\sim \in \{<, >\}$, $p \in [0, 1]$ is a probability value, $\alpha, \beta \in \mathbb{Q}_{\geqslant 0}$, and atomic predicates on markings are defined as $\text{AP} ::= g \bowtie x$ where $\bowtie \in \{<, \leq, =, \neq, \geq, >\}$, $x \in \mathbb{R}$ and $g \colon \mathbb{N}^P \to \mathbb{R}$ is a real-valued function (e.g., "*free* > 1" for the net of Fig. 1a).

As in [20], the logic allows the Boolean composition of interval until operators [12], each evaluated from a random initial state $s_0 = \langle m_0, \vec{\tau}_0 \rangle$ in which m_0 is a marking and $\vec{\tau}_0$ is a vector of times to fire of enabled transitions sampled according to $f_{\vec{\tau}_0}$. Without loss of generality, we assume that all enabled transitions $E(m_0) = \{t_1, t_2, \ldots, t_n\}$ are newly enabled in the initial state, and thus $\vec{\tau}_0$ is distributed according to $f_{\vec{\tau}_0}(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} f_{t_i}(x_i)$. Note that φ-formulae (i.e., the arguments of an until operator) only depend on the marking m of visited states, while ψ-formulae depend on the initial marking m_0 and also on the initial distribution $f_{\vec{\tau}_0}$.

Definition 5. Given a stochastic time Petri net $\langle P, T, A^-, A^+, B, U, EFT, LFT, F, W \rangle$ with initial marking m_0 and times to fire $\vec{\tau}_0$ initially distributed according to $f_{\vec{\tau}_0}$, the relations $\langle m_0, f_{\vec{\tau}_0} \rangle \models \psi$ and $m \models \varphi$ for each $m \in \mathbb{N}^P$ are defined inductively as follows:

$$\begin{aligned}
&\langle m_0, f_{\vec{\tau}_0} \rangle \models \text{TRUE} \text{ and } m \models \text{TRUE} \text{ are always satisfied} \\
&m \models \text{AP} \iff \text{AP is satisfied by } m \\
&m \models \neg\varphi \iff m \not\models \varphi \\
&m \models \varphi_1 \wedge \varphi_2 \iff m \models \varphi_1 \wedge m \models \varphi_2 \\
&\langle m_0, f_{\vec{\tau}_0} \rangle \models \neg\psi \iff \langle m_0, f_{\vec{\tau}_0} \rangle \not\models \psi \\
&\langle m_0, f_{\vec{\tau}_0} \rangle \models \psi_1 \wedge \psi_2 \iff \langle m_0, f_{\vec{\tau}_0} \rangle \models \psi_1 \wedge \langle m_0, f_{\vec{\tau}_0} \rangle \models \psi_2 \\
&\langle m_0, f_{\vec{\tau}_0} \rangle \models P_{\sim p}[\, \varphi_1 \mathcal{U}^{[\alpha,\beta]} \varphi_2 \,] \iff \\
&\qquad Pr_{m_0, f_{\vec{\tau}_0}}\{\omega \in \Omega_{m_0} \mid \omega \models \varphi_1 \mathcal{U}^{[\alpha,\beta]} \varphi_2\} \sim p
\end{aligned}$$

where, for any path $\omega = s_0 \xrightarrow{\gamma_1} s_1 \xrightarrow{\gamma_2} \cdots$ with $s_i = \langle m_i, \vec{\tau}_i \rangle$, $m_i \in \mathbb{N}^P$ and $\vec{\tau}_i \in \mathbb{R}_{\geqslant 0}^{E(m_i)}$ for all i,

$$\begin{aligned}
\omega \models \varphi_1 \mathcal{U}^{[\alpha,\beta]} \varphi_2 \iff \exists n \leq |\omega| \text{ such that } & m_n \models \varphi_2 \wedge \\
\big(\forall k < n . (m_k \models \varphi_1)\big) \wedge \big(T(n, \omega) \in [\alpha, \beta] \vee & \\
(T(n, \omega) < \alpha \wedge T(n+1, \omega) \geq \alpha \wedge m_n \models \varphi_1)\big) . &
\end{aligned} \tag{1}$$

According to Eq. (1), a path satisfying the interval until operator $\varphi_1 \mathcal{U}^{[\alpha,\beta]} \varphi_2$ must visit only φ_1-states and then either reach a φ_2-state in $[\alpha, \beta]$, or reach a $(\varphi_1 \wedge \varphi_2)$-state before time α and let time advance past α. It is worth noting that this formulation based on the index n of each state in a path ensures that also intermediate states with zero sojourn time (which may occur with nonzero probability when the model includes IMM or DET transitions) satisfy φ_1, which could not be specified in the continuous-time formulation of [12].

Example 2. In the G/D/1/2/2 queue of Fig. 1a, the property $P_{<0.4}[(\mathit{buffer} < 2) \mathcal{U}^{[0,7]} (\mathit{failed} = 1)] \wedge P_{<0.2}[(\mathit{buffer} < 2) \mathcal{U}^{[2.5,7]} (\mathit{failed} = 1)]$ is satisfied when the probability of the server being down without ever reaching its full capacity is lower than 0.4 in the interval $[0, 7]$ and lower than 0.2 in the interval $[2.5, 7]$.

The following proposition shows that, for every pair of marking predicates φ_1, φ_2, and $\alpha, \beta \in \mathbb{Q}_{\geqslant 0}$, the set of paths satisfying the interval until operator is an event of $\mathcal{F}_{m_0}$. Concretely, this means that the value of $Pr_{m_0, f_{\vec{\tau}_0}}\{\omega \in \Omega_{m_0} \mid \omega \models \varphi_1 \mathcal{U}^{[\alpha,\beta]} \varphi_2\}$ is well-defined and the semantics of Definition 5 can be computed with stochastic state classes.

Proposition 3. *For each $\varphi_1, \varphi_2 \in \mathbb{N}^P$ and $\alpha, \beta \in \mathbb{Q}_{\geqslant 0}$, the set $\{\, \omega \in \Omega_{m_0} \mid \omega \models \varphi_1 \mathcal{U}^{[\alpha,\beta]} \varphi_2 \,\}$ of paths satisfying the corresponding interval until operator is a countable union of cylinder sets.*

Proof. The cylinder sets that end on a φ_2-marking reached only through φ_1-markings are countable. Each cylinder set is in fact uniquely identified by the sequence of transitions $\gamma_1, \gamma_2, \ldots, \gamma_n$ fired from m_0, which are strings on a finite alphabet. For each cylinder set $C(m_0, \gamma_1, I_1, \gamma_2, I_2, \ldots, \gamma_n, I_n)$ that ends on a φ_2-marking only through φ_1-markings, we consider (1) the cylinder set imposing only an absolute time $I_n = [\alpha, \beta]$ for the nth transition, and (2) if

the marking reached after γ_n satisfies also φ_1, the cylinder sets $C(m_0, \gamma_1, I_1, \gamma_2, I_2, \ldots, \gamma_n, I_n, \gamma_{n+1}, I_{n+1})$ for each $\gamma_{n+1} \in T$ that impose a bound $I_n = [0, \alpha)$ for the firing of γ_n and a bound $I_{n+1} = [\alpha, \infty)$ for the firing of γ_{n+1}. The countable union of these cylinder sets is an event of $\mathcal{F}_{m_0}$ including all and only the successful paths. □

3 Composing Paths Across Regenerations

Paths satisfying an interval until operator can be represented as a countable union of disjoint cylinder sets; when the union is finite or any bounded error is allowed, the formal relation between the measure of cylinder sets and stochastic state classes presented in Section 2.2 provides a concrete solution for the evaluation of $Pr_{m_0, f_{\vec{\tau}_0}}\{\omega \in \Omega_{m_0} \mid \omega \models \varphi_1 \mathcal{U}^{[\alpha,\beta]} \varphi_2\}$. In [27], a solution algorithm is formulated based on the enumeration of stochastic state classes for all finite sequences of fired transitions $\gamma_1, \gamma_2, \ldots, \gamma_n$ and maximal firing time intervals $I_1, I_2, \ldots, I_n$ that satisfy $\varphi_1 \mathcal{U}^{[\alpha,\beta]} \varphi_2$. Similarly to [25], the enumeration of closed-form PDFs enables the accurate evaluation of low probability events, and the on-the-fly analysis of cylinder sets can be stopped early if the interval until operator is decided. As a major limit, the number of discrete events, and thus the number of closed-form PDFs to compute, can grow exponentially with the bound β. To reduce this complexity, we identify discrete events after which the stochastic process exhibits the Markov property; the corresponding (random) time instants, called *regeneration points*, can be leveraged to decompose the evolution of the process through a *Markov renewal argument*: either a given property is satisfied before hitting a regeneration point, or, if a regeneration point is reached, the probability of satisfying the property in the remaining time is conditionally independent of the previous history given the current state. In Section 4, the computation of PDFs will be limited to the first regeneration along each sequence of discrete events, combining measures associated with distinct initial states into a system of integral equations. In this section, we generalize the concept of regeneration of state-of-the-art techniques for stochastic Petri nets [17], [21], [37] by detecting states in which the time elapsed since the enabling of each GEN transition is deterministic; moreover, we highlight the properties of regeneration points in the context of stochastic state classes, and provide an algorithm for their on-the-fly detection during the enumeration of discrete events.

3.1 Extended regenerations

In the evolution of a stochastic process, a *regeneration point* occurs whenever the state of the process subsumes any information on its previous history.

Definition 6. Let $\{ M(t),\ t \geq 0 \}$ be a stochastic process defined on a probability space $(\Omega, \mathcal{F}, P)$ with finite state space S. We say that $\{ M(t),\ t \geq 0 \}$ encounters a regeneration point at time $t \geq 0$ if it satisfies the Markov property in t:

$$P\{ M(t+t') = j \mid M(t) = i, (\forall u \in A_t).M(u) = i_u \} = P\{ M(t') = j \mid M(0) = i \}$$

for all $j \in S$, $t' \geq 0$, $A_t \subseteq [0, t)$, and $i_u \in S$ for $u \in A_t$.

While CTMCs encounter a regeneration point at every time instant, in STPNs regeneration points correspond to discrete events after which GEN timers do not "carry memory" of the previous evolution [17], similarly to EXP timers (which are always memoryless). This is the case of GEN timers that are either reset, or have been enabled for a deterministic time: given the current marking and the deterministic enabling times of GEN timers, the probability distribution of future states is conditionally independent of the previous history. A stochastic state class where the times to fire satisfy this condition is called *regenerative*, and the PDF of the vector $\langle \tau_{age}, \vec{\tau} \rangle$ is always in product form.

Definition 7. A stochastic state class Σ is *regenerative* if, for each enabled GEN transition t_i, the time elapsed from its enabling until the firing that led to Σ is equal to some deterministic value $d_i \in \mathbb{R}_{\geqslant 0}$, which we call the *enabling time* of t_i in Σ.

Remark 1. In a regenerative class $\Sigma = \langle m, D, f_{\langle \tau_{age}, \vec{\tau} \rangle} \rangle$ where the sets of enabled GEN, EXP, and IMM transitions are $\{t_1, \ldots, t_n\}$, $\{t_{n+1}, \ldots, t_m\}$, and $\{t_{m+1}, \ldots, t_l\}$, respectively, if $\vec{d} = (d_1, \ldots, d_n) \in \mathbb{R}^n_{\geqslant 0}$ is the vector of enabling times of GEN transitions, the support D and the probability density function $f_{\langle \tau_{age}, \vec{\tau} \rangle}$ of $\langle \tau_{age}, \vec{\tau} \rangle$ in Σ are equal to

$$D = D_{age} \times \prod_{i=1}^{n} [\max\{0, EFT(t_i) - d_i\}, LFT(t_i) - d_i] \times \prod_{i=n+1}^{m} [0, +\infty) \times \prod_{i=m+1}^{l} [0, 0]$$

and

$$f_{\langle \tau_{age}, \vec{\tau} \rangle}(x_{age}, \vec{x}) = f_{age}(x_{age}) \cdot \prod_{i=1}^{n} \frac{f_{t_i}(x_i + d_i)}{\int_{\max\{d_i, EFT(t_i)\}}^{LFT(t_i)} f_{t_i}(u)\, du} \cdot \prod_{i=n+1}^{m} \lambda_{t_i} e^{-\lambda_{t_i} x_i} \cdot \prod_{i=m+1}^{l} \delta(x_i)$$

respectively, for some PDF f_{age} of τ_{age} with support D_{age}.

Remark 2. As a notable case, a class in which all GEN transitions are newly enabled is regenerative with $\vec{d} = \vec{0}$; this case corresponds to the usual concept of regeneration in the literature on stochastic Petri nets [17], [21]. In particular, this is the regeneration of the initial class Σ_0.

Remark 3. A regenerative class is uniquely identified by its marking m, by the vector $\vec{d}$ of enabling times, and by the PDF f_{age} and support D_{age} of the absolute time τ_{age} of the discrete event associated with the stochastic state class. The marking m identifies the set of enabled transitions, and the support and distribution of EXP and IMM transitions, while the enabling times $\vec{d}$ identify the support and distribution of GEN timers.

Remark 4. In the following, we denote by $(\Omega_m, \mathcal{F}_m, Pr_{m,\vec{d}})$ the probability space for the paths of an STPN with initial marking m when times to fire are initially distributed according to a PDF $f_{\vec{\tau}_0}$ with the product form described in Remark 1 for enabling times $\vec{d}$ and τ_{age} equal to zero (i.e., $f_{age}(x_{age}) = \delta(x_{age})$ and $D_{age} = [0, 0]$).

3.2 Detection of regeneration points in the enumeration of stochastic state classes

At regeneration points, the time elapsed since the enabling of each GEN transition must be deterministic. In order to detect regeneration points in the enumeration of successor classes, we analyze synchronizations between the enabling of DET/IMM and GEN transitions over sequences of firings: immediately after the firing of a DET or IMM transition u, a GEN transition t has been enabled for a deterministic time iff t was enabled together with u, or with a deterministic delay (or advance) with respect to the enabling of u.

We define the map $\text{ENABLING}(u,t)$ to record, for each DET/IMM transition u, the deterministic time from the enabling of each GEN transition t until the scheduled firing of u, or NIL if this time is not deterministic. The algorithm in Fig. 2 constructs $\text{ENABLING}(u,t)$ for an initial regeneration $(m_0, \vec{d}_0)$ which assigns deterministic enabling time $\vec{d}_0(t)$ to each GEN transition t (lines 1–3). After the firing of the ith transition in the sequence $\gamma_1, \ldots, \gamma_n$, the enabling time $\vec{d}(t)$ of each GEN transition t is recomputed from $\text{ENABLING}(\gamma_i, t)$ (lines 7–13): if $\vec{d}$ assigns a deterministic enabling time to each enabled GEN transition, a regeneration point is detected, and the regeneration $(m, \vec{d})$ is returned (line 16), where m is the marking reached after the firing of γ_i. Otherwise, the map $\text{ENABLING}(u,t)$ is updated to maintain its invariant: information on each disabled transition is removed (lines 18–21) and, for each DET/IMM transition u and GEN transition t with deterministic enabling times $\vec{d}(u)$ and $\vec{d}(t)$, respectively, the enabling time of t at the firing of u is set to $\vec{d}(t) + [LFT(u) - \vec{d}(u)]$ (lines 22–24).

While the algorithm presented in Fig. 2 detects the first regeneration on a finite sequence of transition firings, lines 5–24 can be executed after the enumeration of each stochastic state class to identify regenerative classes and regenerations on-the-fly.

DETECT-FIRST-REGENERATION$((m_0, \vec{d}_0), \gamma_1, \ldots, \gamma_n)$

```
1  for each DET/IMM transition u enabled by m_0
2      for each GEN transition t enabled by m_0
3          ENABLING(u,t) ← d_0(t) + [LFT(u) − d_0(u)]
4  for i = 1 to n
5      regeneration ← TRUE
6      d(t) ← NIL for each GEN transition t
7      for each GEN transition t enabled after the firing of γ_i
8          if t is newly enabled
9              d(t) ← 0
10         elseif γ_i is DET/IMM ∧ ENABLING(γ_i, t) ≠ NIL
11             d(t) ← ENABLING(γ_i, t)
12         else
13             regeneration ← FALSE
14     if regeneration = TRUE
15         m ← marking reached after the firing of γ_i
16         return (m, d)
17     else
18         for each disabled transition x
19             ENABLING(u,x) ← NIL for each DET/IMM u
20             if x is DET/IMM
21                 ENABLING(x,t) ← NIL for each GEN t
22         for each DET/IMM transition u s.t. d(u) ≠ NIL
23             for each GEN transition t s.t. d(t) ≠ NIL
24                 ENABLING(u,t) ← d(t) + [LFT(u) − d(u)]
25 return NIL
```

Fig. 2: Algorithm detecting the first regeneration over the sequence of transitions $\gamma_1, \ldots, \gamma_n$ fired from the initial regeneration $(m_0, \vec{d}_0)$.

3.3 Properties of regeneration points

The transient behavior of an STPN is fully characterized by an infinite tree encoding the succession relation among stochastic state classes for all sequences of fired transitions. In the following, we omit the firing time interval I in the notation $\Sigma \xrightarrow{\gamma,I,\mu} \Sigma'$ whenever $I = [0, +\infty)$ (i.e., no constraint is imposed on the absolute firing time).

Definition 8. The *transient stochastic tree* from an initial class Σ_0 is a tuple $\langle N, E, n_0, \Sigma \rangle$ where: the set N is a countable set of nodes; $n_0 \in N$ is the root node; the function Σ associates each node $n \in N$ with a stochastic state class $\Sigma(n)$; the labeled edges $E \subseteq N \times T \times (0,1] \times N$ represent the (unconstrained) successions of stochastic state classes associated with transition firings, i.e., $(n, \gamma, \mu, n') \in E$ if and only if $\Sigma(n) \xRightarrow{\gamma,\mu} \Sigma(n')$.

When $\Sigma(n_0)$ corresponds to the initial class Σ_0 (Section 2.2), the transient tree enumerates all the sequences of transition firings with non-null probability, and associates each node with the resulting PDF over states and absolute reaching times. A node n associated with a regenerative class $\Sigma(n) = \langle m, D, f_{\langle \tau_{age}, \vec{\tau} \rangle} \rangle$ is said to be *regenerative*, and it satisfies two major properties presented in the following. The first property guarantees that two regenerative nodes reached at different times, but associated with the same marking and deterministic enabling times, enable the same firing sequences with the same probabilities.

Lemma 1. *Let n_i and n_j be two regenerative nodes associated with stochastic state classes $\Sigma(n_i)$ and $\Sigma(n_j)$ with the same marking m and the same deterministic enabling times $\vec{d}$. Then, the succession sequences feasible from n_i and from n_j are the same, they have the same probability, and they end up in nodes associated with stochastic state classes that have the same marking and the same random vector $\vec{\tau}$ of times to fire for enabled transitions.*

Proof. The proof runs by induction on the length of succession sequences originating from the nodes n_i and n_j, and leverages the fact that the two classes have the same marginal distribution of times to fire, so that they will allow the same set of feasible behaviors with the same probabilities: according to Remark 3, $\Sigma(n_i)$ and $\Sigma(n_j)$ have the same support and distribution for the vector of times to fire of enabled transitions; the former condition implies that they accept the same sets of feasible behaviors (sequences of firable transitions), and that equal succession sequences result in the same final markings and times to fire supports (due to the underlying non-deterministic model); the latter condition implies that probabilities of these firing sequences are also the same, and that they end up in classes with the same PDF of times to fire. □

Lemma 1 gives relevance to the pair $(m, \vec{d})$, which captures a condition that can be reached at different times

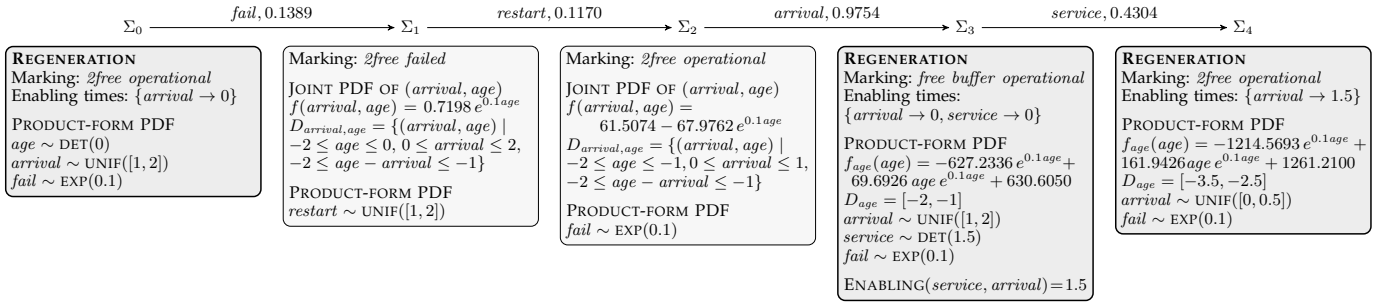


Fig. 3: Stochastic state classes for the sequence of events *fail*, *restart*, *arrival*, *service* in the queue of Fig. 1a.

in different regenerative classes, but always produces the same subsequent behaviors. In the sequel, we refer to the pair $(m, \vec{d})$ as *regeneration condition*.

The next lemma completes the picture by focusing on the advancement of the age before and after a regenerative node, and it fully exploits the properties of regenerative stochastic state classes to show that the times elapsed before and after the regeneration are independent.

Lemma 2. *Let n_j be a regenerative node, n_k be the descendant of n_j reached through the transitions $\rho = \gamma_0, \ldots, \gamma_n$, and f^j_{age} and f^k_{age} be the marginal PDF of the τ_{age} variables in $\Sigma(n_j)$ and $\Sigma(n_k)$, respectively. Then,*

$$f^k_{age}(x_{age}) = \int_{-\infty}^{+\infty} f^j_{age}(u)\, \hat{f}^k_{age}(x_{age} - u)\, du \qquad (2)$$

where $\hat{f}^k_{age}$ is the marginal PDF of τ_{age} for the node $\hat{n}_k$ reached through ρ in the transient tree rooted in a node $\hat{n}_j$ with the same marking and random vector as n_j, but with τ_{age} distributed as $\hat{f}^j_{age}(x_{age}) = \delta(x_{age})$.

Proof. The time spent in the execution of ρ from n_j or from $\hat{n}_j$ is the same as it only depends on the marginal distribution of times to fire in n_j and $\hat{n}_j$ (Lemma 1); moreover, the distribution of this time is given by $\hat{f}^k_{age}$, since the age is equal to zero in $\hat{n}_j$, i.e., $\hat{f}^j_{age}(x_{age}) = \delta(x_{age})$. Since n_j is regenerative, f^j_{age} is in product form with respect to the marginal PDF of times to fire (Remark 1), and thus the evolution from n_j is independent of the time at which the node is reached; the age in n_k is then the sum of the independent random variables associated with the age in n_j and the duration of ρ, and it is distributed as the convolution $\int_{-\infty}^{+\infty} f^j_{age}(u)\, \hat{f}^k_{age}(x_{age} - u)\, du$. □

Note that in Lemma 1 the assumption that n_i and n_j are regenerative is used only to guarantee that they have the same marginal PDF for the vector of times to fire. In Lemma 2, the assumption of regeneration is used also to guarantee that the PDFs of τ_{age} and $\vec{\tau}$ are in product form.

Example 3. In order to illustrate the concept of regeneration points, we report in Fig. 3 the stochastic state classes corresponding to the cylinder set with initial marking *2free operational* and sequence of discrete events *fail*, *restart*, *arrival*, *service* (without constraints on the absolute firing times) for the STPN of Fig. 1a. The probability measure of the continuous set of paths contained in the cylinder set is given by the product of succession probabilities (which we report up to the fourth significant figure). In regenerative stochastic state classes Σ_0, Σ_3, Σ_4, the absolute time of the last event and all the times to fire of enabled transitions are independent random variables with product-form PDF; notably, the GEN transition *arrival* in Σ_4 is not newly enabled, but its PDF is uniquely identified by the deterministic enabling time 1.5. Note that, as discussed in [35], τ_{age} encodes the *negation* of the absolute firing time, in order to operate on variables decreasing with the same rate.

Remark 5 (Nesting of until operators). In a Markov regenerative process, the discrete component of the state does not carry sufficient information to verify nested sub-formulas independently. We illustrate the problem with an example inspired by [10]. Consider an STPN with only one place p and one transition t with time to fire uniform on $[0, 4]$. Let p be a precondition for t, and let the initial marking assign one token to p: at the firing of t, the token is removed and no further transition is enabled. The underlying marking process is semi-Markov with two states: $p = 1$ and $p = 0$. Consider now the nested formula $\varphi_{ext} = P_{>1/3}[\varphi_{in}\, \mathcal{U}^{[0,2]}(p = 0)]$ with $\varphi_{in} = P_{<1/3}[(p = 1)\, \mathcal{U}^{[0,1]}(p = 0)]$. Intuitively, for φ_{ext} to hold, φ_{in} must be satisfied continuously until the firing of t. However, due to the dependency on past evolution introduced by the GEN sojourn time, the satisfaction of the inner formula φ_{in} changes over time: given a sojourn of x time units, the probability of $(p = 1)\, \mathcal{U}^{[0,1]}(p = 0)$ is $\int_x^{x+1} \mathbb{1}_{[x,4]}/(4-x)\, dy$, which is monotonically nondecreasing and remains under $1/3$ only when $x \in [0, 1)$. Therefore, the only paths that satisfy φ_{in} continuously are those that sojourn in $p = 1$ less than $\bar{x} = 1$ time units, and the probability of satisfying the nested operator $\varphi_{in}\, \mathcal{U}^{[0,2]}(p = 0)$ is $\int_0^{\bar{x}} 1/4\, dy = 1/4$. By this argument, we conclude that the external formula φ_{ext} is not satisfied; the important remark is that this conclusion requires that initial states be distinguished not only by their marking, but also by the sojourn duration.

4 Regenerative Solution of an Interval Until Operator

4.1 Renewal equations for the until operator

Given the predicates φ_1 and φ_2, a real interval $[\alpha, \beta]$ with $\alpha, \beta \in \mathbb{Q}_{\geqslant 0}$, and a regeneration condition $i = (m, \vec{d})$, we define

$$\Omega_i(\alpha, \beta) := \{\, \omega \in \Omega_m \mid \omega \models \varphi_1\, \mathcal{U}^{[\alpha,\beta]} \varphi_2 \,\}$$

to be the set of paths that start from the marking of the regeneration condition i and satisfy the until operator, and

we denote by $p_i(\alpha,\beta) := Pr_i\{\Omega_i(\alpha,\beta)\}$ its probability measure defined as in Remark 4 (Section 4). For each path $\omega = s_0 \xrightarrow{\gamma_1} s_1 \xrightarrow{\gamma_2} \cdots$ in $\Omega_i(\alpha,\beta)$, with $s_n = \langle m_n, \vec{\tau}_n\rangle$, $m_n \in N^P$ and $\vec{\tau}_n \in \mathbb{R}_{\geqslant 0}^{E(m_n)}$, we indicate as $\text{REG}(\omega)$ the index of the first regeneration along the path:

$$\text{REG}(\omega) := \min\{n \in \mathbb{N} \mid s_{n-1} \xrightarrow{\gamma_n} s_n \text{ is a regeneration}\}$$

and we indicate as $\text{OK}(\omega)$ the index of the first state leading to the satisfaction of the until operator:

$$\begin{aligned}\text{OK}(\omega) := \min\{n \in \mathbb{N} \mid n \leq |\omega| \wedge (m_n \models \varphi_2) \wedge \\ (\forall k < n.(m_k \models \varphi_1)) \wedge (T(n,\omega) \in [\alpha,\beta] \vee \\ (T(n,\omega) < \alpha \wedge T(n+1,\omega) \geq \alpha \wedge m_n \models \varphi_1))\}.\end{aligned}$$

Moreover, we indicate with $t_{\text{REG}}(\omega) := T(\text{REG}(\omega),\omega)$ the time of the first regeneration in ω, and with $(m_{\text{REG}(\omega)}, \vec{d}_{\text{REG}(\omega)})$ the corresponding regeneration condition. The probability $p_i(\alpha,\beta)$ can then be decomposed so as to separately account for paths in $\Omega_i(\alpha,\beta)$ that satisfy the until operator under different timings of the first regeneration. To this end, we distinguish paths that satisfy the until operator before reaching a regeneration from those that encounter the first regeneration before α, or between α and β, and then satisfy the until operator:

$$\begin{aligned}\Omega_i^L(\alpha,\beta) &:= \{\omega \in \Omega_i(\alpha,\beta) \mid \text{OK}(\omega) < \text{REG}(\omega)\}, \\ \Omega_i^G(\alpha,\beta) &:= \{\omega \in \Omega_i(\alpha,\beta) \mid \\ &\qquad \text{OK}(\omega) \geq \text{REG}(\omega) \wedge t_{\text{REG}}(\omega) < \alpha\}, \\ \Omega_i^H(\alpha,\beta) &:= \{\omega \in \Omega_i(\alpha,\beta) \mid \\ &\qquad \text{OK}(\omega) \geq \text{REG}(\omega) \wedge t_{\text{REG}}(\omega) \in [\alpha,\beta]\}.\end{aligned}$$

Proposition 4. *For any regeneration condition i and $\alpha,\beta \in \mathbb{Q}_{\geqslant 0}$, it holds $p_i(\alpha,\beta) = Pr_i\{\Omega_i^L(\alpha,\beta)\} + Pr_i\{\Omega_i^G(\alpha,\beta)\} + Pr_i\{\Omega_i^H(\alpha,\beta)\}$.*

Proof. The sets $\Omega_i^L(\alpha,\beta)$, $\Omega_i^G(\alpha,\beta)$, $\Omega_i^H(\alpha,\beta)$ are a partition of the set of paths $\Omega_i(\alpha,\beta)$: on the one hand, they are clearly disjoint; on the other hand, to prove that their union is $\Omega_i(\alpha,\beta)$ it is sufficient to consider that $\forall\omega \in \Omega_i(\alpha,\beta)$, $t_{\text{REG}}(\omega) > \beta$ implies $\text{OK}(\omega) < \text{REG}(\omega)$. □

The probability measure of the three sets $\Omega_i^L(\alpha,\beta)$, $\Omega_i^G(\alpha,\beta)$ and $\Omega_i^H(\alpha,\beta)$ can be expressed in terms of three *kernels* characterizing the behavior of the stochastic process within the first epoch of regeneration.

The measure of $\Omega_i^L(\alpha,\beta)$ is directly defined as the *local kernel* $L_i^{\varphi_1,\varphi_2}(\alpha,\beta) := Pr_i\{\Omega_i^L(\alpha,\beta)\}$, which evaluates the probability measure of paths that satisfy the until operator before reaching a regeneration. In contrast, the measures of $\Omega_i^G(\alpha,\beta)$ and $\Omega_i^H(\alpha,\beta)$ are not limited to a regeneration epoch and require the next Propositions 5 and 6.

Proposition 5. *The measure $Pr_i\{\Omega_i^G(\alpha,\beta)\}$ is equal to*

$$\sum_k \int_{x\in[0,\alpha)} dG_{ik}^{\varphi_1}(x)\, p_k(\alpha - x, \beta - x) \tag{3}$$

where $k = (m,\vec{d})$ ranges over all reachable regeneration conditions and the global kernel *$G_{ik}^{\varphi_1}(x)$ is defined as*

$$\begin{aligned}G_{ik}^{\varphi_1}(x) := Pr_i\{\omega \in \Omega_i \mid t_{\text{REG}}(\omega) \leq x \wedge \\ (m_{\text{REG}(\omega)}, \vec{d}_{\text{REG}(\omega)}) = k \wedge (\forall j < \text{REG}(\omega)).(m_j \models \varphi_1)\}.\end{aligned} \tag{4}$$

Proof. For each $\omega \in \Omega_i^G(\alpha,\beta)$, it must be $\text{OK}(\omega) \geq \text{REG}(\omega)$ and $t_{\text{REG}}(\omega) < \alpha$ (i.e., ω encounters a goal state after reaching a regeneration point before time α). According to Lemmas 1 and 2, the process evolution after the regeneration point depends only on the regeneration condition $(m_{\text{REG}(\omega)}, \vec{d}_{\text{REG}(\omega)})$, and the time remaining for the satisfaction of the until operator is reduced by $t_{\text{REG}}(\omega)$; since the only condition required by Eq. (1) for states $(m_j,\vec{\tau}_j)$ with $j < \text{OK}(\omega)$ is $m_j \models \varphi_1$, and $\text{OK}(\omega) \geq \text{REG}(\omega)$ for $\omega \in \Omega_i^G(\alpha,\beta)$, the measure $Pr_i\{\Omega_i^G(\alpha,\beta)\}$ is equal to

$$\int_{X(\alpha)} p_{(m_{\text{REG}(\omega)},\vec{d}_{\text{REG}(\omega)})}(\alpha - t_{\text{REG}}(\omega), \beta - t_{\text{REG}}(\omega))\, dPr_i(\omega) \tag{5}$$

where $X(\alpha) := \{\omega \in \Omega_i \mid t_{\text{REG}}(\omega) < \alpha \wedge (\forall j < \text{REG}(\omega)).(m_j \models \varphi_1)\}$. In Eq. (5), the measure of each path reaching its first regeneration point before α without violating φ_1 is multiplied by the probability that the until operator will be satisfied from that regeneration in the remaining time. By conditioning on all reachable regeneration conditions $(m_{\text{REG}(\omega)}, \vec{d}_{\text{REG}(\omega)}) = k$ and times $t_{\text{REG}}(\omega) = x$ of the first regeneration before α, we obtain Eq. (3), where the global kernel represents the probability of reaching, within time x, a regeneration with regeneration condition k while always satisfying φ_1 in previous states. □

Proposition 6. *The measure $Pr_i\{\Omega_i^H(\alpha,\beta)\}$ is equal to*

$$\sum_k \int_{x\in[\alpha,\beta]} dH_{ik}^{\varphi_1,\varphi_2}(\alpha,x)\, p_k(0,\beta - x) \tag{6}$$

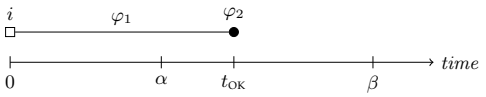
where $k = (m,\vec{d})$ ranges over all reachable regeneration conditions and the conditional global kernel *$H_{ik}^{\varphi_1,\varphi_2}(\alpha,x)$ is defined as*

$$\begin{aligned}H_{ik}^{\varphi_1,\varphi_2}(\alpha,x) := Pr_i\{\omega \in \Omega_i \mid t_{\text{REG}}(\omega) \in [\alpha,x] \wedge \\ (m_{\text{REG}(\omega)}, \vec{d}_{\text{REG}(\omega)}) = k \wedge (\forall j < \text{REG}(\omega)).(m_j \models \varphi_1 \wedge \\ (m_j \models \varphi_2) \Rightarrow T(j+1,\omega) < \alpha)\}.\end{aligned} \tag{7}$$

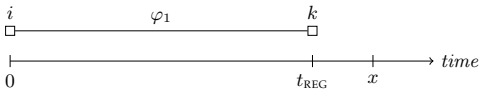
Proof. For each $\omega \in \Omega_i^H(\alpha,\beta)$, it must be that $\text{OK}(\omega) \geq \text{REG}(\omega)$ and $t_{\text{REG}}(\omega) \in [\alpha,\beta]$ (i.e., ω encounters a goal state after reaching a regeneration point at some time in $[\alpha,\beta]$). The proof is analogous to the case of Proposition 5: in this case, states $(m_j,\vec{\tau}_j)$ with $j < \text{REG}(\omega)$ must satisfy φ_1, but not φ_2 if the sojourn lasts until after α: otherwise, the formula would be satisfied at time α by letting time advance in a φ_2-state, and we would have $\text{OK}(\omega) < \text{REG}(\omega)$. □

Propositions 5 and 6 provide an important result, as they apply renewal arguments to the satisfaction of the until operator and distinguish the properties that must be satisfied by paths before a regeneration point in $[0,\alpha)$ or in $[\alpha,\beta]$. We can now present our main result, which follows directly from Propositions 4 to 6 and shows that the measure $p_{(m_0,\vec{0})}(\alpha,\beta)$ of paths satisfying the until operator from the initial regeneration $(m_0,\vec{0})$ can be computed from the measures $p_i(\alpha,\beta)$ for all possible i, α, β.

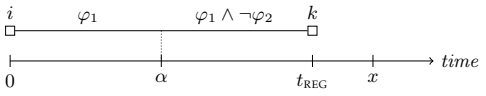
Theorem 1. *The measures $p_i(\alpha,\beta)$ for all $i = (m,\vec{d})$, each corresponding to the probability that the model satisfies the interval until operator $\varphi_1 \mathcal{U}^{[\alpha,\beta]} \varphi_2$ from the initial marking m with PDF*



(a) Paths contributing to $L_i^{\varphi_1,\varphi_2}(\alpha,\beta)$.



(b) Paths contributing to $G_{ik}^{\varphi_1}(x)$.



(c) Paths contributing to $H_{ik}^{\varphi_1,\varphi_2}(\alpha,x)$.

Fig. 4: Constraints on paths contributing to the kernels.

of GEN timers given by the deterministic enabling times $\vec{d}$, are given by the system of integral equations

$$\begin{aligned} p_i(\alpha,\beta) &= L_i^{\varphi_1,\varphi_2}(\alpha,\beta) \\ &+ \sum_k \int_{x\in[0,\alpha)} dG_{ik}^{\varphi_1}(x)\; p_k(\alpha-x,\beta-x) \\ &+ \sum_k \int_{x\in[\alpha,\beta]} dH_{ik}^{\varphi_1,\varphi_2}(\alpha,x)\; p_k(0,\beta-x) \end{aligned} \tag{8}$$

where i and k range over all reachable regeneration conditions.

The theorem represents a bivariate generalization of Markov renewal equations [18], [38] leveraging three kernels that result from a renewal argument specific to the interval until operator: the model can satisfy φ_2 between α and β either (1) without regenerations, (2) reaching the first regeneration before α, or (3) reaching the first regeneration in $[\alpha,\beta]$. As illustrated in Fig. 4, φ_1 must always be satisfied; additionally, also $\neg\varphi_2$ must be satisfied between α and the first regeneration in paths that satisfy the until operator only after a regeneration in $[\alpha,\beta]$.

The bivariate unknowns $p_i(\alpha,\beta)$ take into account both a minimum and maximum time for the satisfaction of φ_2; after a regeneration at time x with regeneration condition k, the success probability is given by the solution from k with reduced time constraints: $p_k(\alpha-x,\beta-x)$ if $x<\alpha$ and $p_k(0,\beta-x)$ if $x\geq\alpha$. In the next section, we will show that the numerical solution of the integral equations for $p_i(\alpha,\beta)$ requires a number of unknowns $p_k(x,y)$ that grows linearly with respect to β, similarly to the required values of $L_i^{\varphi_1,\varphi_2}$ and $G_{ik}^{\varphi_1}$; in contrast, the number of required values of $H_{ik}^{\varphi_1,\varphi_2}$ grows linearly with the product $\alpha(\beta-\alpha)$, as illustrated in Fig. 5.

4.2 Numerical integration and kernels evaluation

The kernels can be evaluated through the enumeration of stochastic state classes limited to the first regeneration along sequences of discrete events; Eq. (8) can then be solved numerically in the time domain through techniques such as Newton–Cotes formulas or Runge–Kutta methods [39], with various trade-offs between accuracy and complexity. Given a step h, discretizing the temporal domain $[0,\beta]$ into points

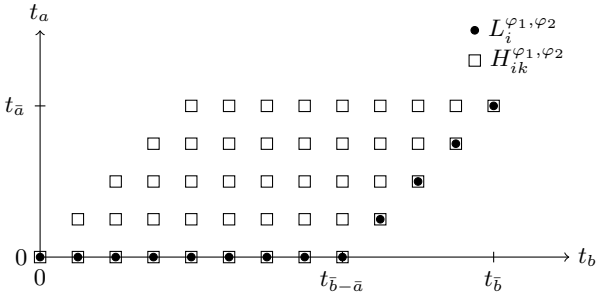


Fig. 5: Required values of $L_i^{\varphi_1,\varphi_2}(t_a,t_b)$ and $H_{ik}^{\varphi_1,\varphi_2}(t_a,t_b)$.

$t_n=nh$, with $\alpha=\bar{a}h$ and $\beta=\bar{b}h$, Newton–Cotes formulas define the linear system

$$\begin{aligned} \vec{p}(t_a,t_b) &= \vec{L}^{\varphi_1,\varphi_2}(t_a,t_b) \\ &+ \sum_{m=0}^{a} w_m\, d\mathbf{G}^{\varphi_1}(t_m)\,\vec{p}(t_{a-m},t_{b-m}) \\ &+ \sum_{m=a}^{b} w_m\, d\mathbf{H}^{\varphi_1,\varphi_2}(t_a,t_m)\,\vec{p}(0,t_{b-m}) \end{aligned} \tag{9}$$

in the unknowns $\vec{p}(0,t_b)$ for $b=0,\ldots,\bar{b}-\bar{a}$, and $\vec{p}(t_a,t_{a+\bar{b}-\bar{a}})$ for $a=1,\ldots,\bar{a}$, where, for first-degree formulas (*trapezoidal rule*), $w_m=h/2$ for $m=0$, $m=a$, or $m=b$, and $w_m=h$ otherwise. For regular MRPs $d\mathbf{G}(0)=0$ and $d\mathbf{H}(0,0)=0$, and Eq. (9) can be solved by forward substitution; in particular,

$$\vec{p}(0,t_b)=\vec{L}^{\varphi_1,\varphi_2}(0,t_b)+\sum_{m=1}^{b} w_m\, d\mathbf{H}^{\varphi_1,\varphi_2}(0,t_m)\,\vec{p}(0,t_{b-m})$$

for $b=0,\ldots,\bar{b}-\bar{a}$, and

$$\begin{aligned} \vec{p}(t_a,t_b) &= \vec{L}^{\varphi_1,\varphi_2}(t_a,t_b) \\ &+ \sum_{m=1}^{a} w_m\, d\mathbf{G}^{\varphi_1}(t_m)\,\vec{p}(t_{a-m},t_{b-m}) \\ &+ \sum_{m=a}^{b} w_m\, d\mathbf{H}^{\varphi_1,\varphi_2}(t_a,t_m)\,\vec{p}(0,t_{b-m}) \end{aligned} \tag{10}$$

for $a=1,\ldots,\bar{a}$ and $b=a+\bar{b}-\bar{a}$. Evaluating the unknowns $\vec{p}(t_a,t_b)$ in this order, the solution $\vec{p}(t_{\bar{a}},t_{\bar{b}})$ can be computed as a direct sum that requires:

- local kernel values $\vec{L}^{\varphi_1,\varphi_2}(0,t_b)$ for $b=0,\ldots,\bar{b}-\bar{a}$ and $\vec{L}^{\varphi_1,\varphi_2}(t_a,t_{a+\bar{b}-\bar{a}})$ for $a=1,\ldots,\bar{a}$;
- global kernel values $d\mathbf{G}^{\varphi_1}(t_m)$ for $m=1,\ldots,\bar{a}$;
- conditional global kernel values $d\mathbf{H}^{\varphi_1,\varphi_2}(t_a,t_m)$ for $a=0,\ldots,\bar{a}$ and $m=a,\ldots,a+\bar{b}-\bar{a}$.

Values of $L_i^{\varphi_1,\varphi_2}(t_a,t_b)$, $dG_{ik}^{\varphi_1}(t_m)$, and $dH_{ik}^{\varphi_1,\varphi_2}(t_a,t_m)$ can be derived from the transient tree enumerated from regeneration condition i, halting on (1) regenerative nodes, (2) nodes not satisfying φ_1, (3) nodes with minimum reaching time τ_{age} greater than β. In the enumeration, the successors of a class Σ, indicated as SUCCESSORS(Σ), are derived according to the calculus described in the Appendix in the supplemental material, and in [29], [30], [35]. Each class Σ_n derived through successions $\Sigma_{i-1} \overset{\gamma_i,\mu_i}{\Longrightarrow} \Sigma_i$ for $i=1,\ldots,n$ is associated with the probability measure $\eta(\Sigma_n)=\prod_{i=1}^{n}\mu_i$ of the cylinder set of paths that perform the

EVALUATE-$L_i^{\varphi_1,\varphi_2}(\alpha,\beta)$

1 $\Sigma_0 =$ initial state class with regeneration condition i
2 $p \leftarrow 0$
3 $\Gamma \leftarrow \{\, \Sigma_0 \,\}$
4 **while** $\Gamma \neq \emptyset$
5 select and remove a class $\Sigma = \langle m, D, f \rangle$ from Γ
6 **if** $m \models \neg\varphi_1 \wedge \neg\varphi_2$ or Σ is regenerative
7 discard Σ
8 **elseif** $m \models \neg\varphi_1 \wedge \varphi_2$
9 $p \leftarrow p + \eta(\Sigma_{in\in[\alpha,\beta]})$
10 **elseif** $m \models \varphi_1 \wedge \neg\varphi_2$
11 $\Gamma \leftarrow \Gamma \cup \text{SUCCESSORS}(\Sigma_{out\in[0,\beta]})$
12 **elseif** $m \models \varphi_1 \wedge \varphi_2$
13 $p \leftarrow p + \eta(\Sigma_{in\in[\alpha,\beta]})$
14 $+ \eta(\Sigma_{in\in[0,\alpha),out\in[\alpha,+\infty)})$
15 $\Gamma \leftarrow \Gamma \cup \text{SUCCESSORS}(\Sigma_{out\in[0,\alpha)})$
16 **return** p

Fig. 6: Algorithm evaluating $L_i^{\varphi_1,\varphi_2}(\alpha,\beta)$.

sequence of discrete events $\gamma_1, \ldots, \gamma_n$. Additional constraints on paths can be imposed by restricting the set of values of times to fire; in particular, given a stochastic state class $\Sigma = \langle m, D, f \rangle$ and the intervals I_1 and I_2, we indicate as $\Sigma_{in\in I_1, out\in I_2} = \langle m, D_{in\in I_1, out\in I_2}, f_{in\in I_1, out\in I_2} \rangle$ where

$$D_{in\in I_1,\, out\in I_2} := \{\, \langle \tau_{age}, \vec{\tau} \rangle \in D \mid -\tau_{age} \in I_1 \text{ and } (\min_i \tau_i) - \tau_{age} \in I_2 \,\}$$

$$\eta(\Sigma_{in\in I_1,\, out\in I_2}) := \eta(\Sigma) \int_{D_{in\in I_1,\, out\in I_2}} f_{\langle \tau_{age}, \vec{\tau} \rangle}(x_{age}, \vec{x})\, dx_{age}\, d\vec{x}$$

$$f_{in\in I_1,\, out\in I_2}(x_{age}, \vec{x}) := f(x_{age}, \vec{x}) \frac{\eta(\Sigma)}{\eta(\Sigma_{in\in I_1,\, out\in I_2})}$$

the class Σ conditioned on the event imposing that the last firing happened at some time in I_1 and the next firing will happen at some time in I_2 (note that τ_{age} encodes the negation of the absolute reaching time). Correspondingly, $\eta(\Sigma_{in\in I_1,\, out\in I_2})$ represents the measure of the cylinder set of paths where the firings that enter and leave Σ occur in the intervals I_1 and I_2, respectively. In the following, the superfluous restrictions $in \in [0, +\infty)$ and $out \in [0, +\infty)$ will be omitted in the notation.

Local kernel values $L_i^{\varphi_1,\varphi_2}(t_a, t_b)$. The algorithm in Fig. 6 evaluates $L_i^{\varphi_1,\varphi_2}(\alpha,\beta)$ by enumerating the transient tree from regeneration condition i. Specifically, Γ is the frontier set containing classes to be processed and p accumulates the value of $L_i^{\varphi_1,\varphi_2}(\alpha,\beta)$. For each non-regenerative class Σ selected from Γ, three cases are possible, depending on the satisfaction of φ_1 and φ_2:

- A state in a class $\neg\varphi_1 \wedge \varphi_2$ (line 8), contributes to the probability p iff it is reached in $[\alpha,\beta]$; according to this, p is incremented by the measure of the subset of Σ restricted with the constraint $in \in [\alpha,\beta]$.
- A state in a class $\varphi_1 \wedge \neg\varphi_2$ (line 10) does not contribute to p, but its successors can, provided that they are reached within β; according to this, the successors of Σ that are reached within β are added to Γ.
- A state in a class $\varphi_1 \wedge \varphi_2$ (line 12) can contribute to either p or the frontier Γ: p is incremented by the measure of the states in Σ that are reached within $[\alpha,\beta]$, or reached before α and left after α; the successors of Σ are added to Γ iff they are reached before α.

EVALUATE-$\vec{H}_i^{\varphi_1,\varphi_2}(\alpha,x)$

1 $\Sigma_0 =$ initial state class with regeneration condition i
2 $p_k \leftarrow 0$ for each regeneration condition k
3 $\Gamma \leftarrow \{\, \Sigma_0 \,\}$
4 **while** $\Gamma \neq \emptyset$
5 select and remove a class $\Sigma = \langle m, D, f \rangle$ from Γ
6 **if** Σ is regenerative with regeneration condition k
7 $p_k \leftarrow p_k + \eta(\Sigma_{in\in[0,x]})$
8 **elseif** $m \models \varphi_1 \wedge \neg\varphi_2$
9 $\Gamma \leftarrow \Gamma \cup \text{SUCCESSORS}(\Sigma_{out\in[0,x]})$
10 **elseif** $m \models \varphi_1 \wedge \varphi_2$
11 $\Gamma \leftarrow \Gamma \cup \text{SUCCESSORS}(\Sigma_{out\in[0,\alpha)})$
12 **return** $\vec{p}$

Fig. 7: Algorithm evaluating $\vec{H}_{ik}^{\varphi_1,\varphi_2}(\alpha,x)$.

Global kernel values $dG_{ik}^{\varphi_1}(t_m)$. The values $dG_{ik}^{\varphi_1}(t_m)$ for $m = 1, \ldots, \bar{a}$ can be derived from the transient tree enumerated from regeneration condition i, stopping on any regenerative class, or on any $(\neg\varphi_1)$-class, or after the time limit α. In particular, the value $dG_{ik}^{\varphi_1}(t_m)$ can be obtained by summing up, over each regenerative class n with regeneration k, the PDF value of the absolute reaching time multiplied by $\eta(n)$, i.e., $dG_{ik}^{\varphi_1}(t_m) = \sum_n \eta(n)\, f_{age}^n(-t_m)$.

Conditional global kernel values $dH_{ik}^{\varphi_1,\varphi_2}(t_a, t_m)$. The values $dH_{ik}^{\varphi_1,\varphi_2}(t_a, t_m)$ can be approximated as $(H_{ik}^{\varphi_1,\varphi_2}(t_a, t_m) - H_{ik}^{\varphi_1,\varphi_2}(t_a, t_{m-1}))/h$, where the values $H_{ik}^{\varphi_1,\varphi_2}(\alpha,x)$ are derived from the transient tree enumerated from regenerative condition i stopping on regenerations, on $(\neg\varphi_1)$-classes, and on classes reached after β. The evaluation also discards states in φ_2-classes that are left after α, since $H_{ik}^{\varphi_1,\varphi_2}(\alpha,x)$ provides the measure of the set of paths that end on regeneration condition k after visiting only φ_1-states and without visiting any φ_2-state after α. The algorithm in Fig. 7 evaluates $H_{ik}^{\varphi_1,\varphi_2}(\alpha,x)$ for all k by enumerating the transient tree from regeneration condition i; similarly to Fig. 6, Γ is the frontier set and p_k accumulates the value of $H_{ik}^{\varphi_1,\varphi_2}(\alpha,x)$. For each state class Σ selected from Γ:

- A state in a regenerative class with regeneration condition k (line 6), contributes to the probability p_k of $H_{ik}^{\varphi_1,\varphi_2}(\alpha,x)$ iff it is reached before time x; according to this, p_k is incremented by the measure of the subset of Σ restricted with the constraint $in \in [0,x]$.
- A state in a class $\varphi_1 \wedge \neg\varphi_2$ (line 8) does not contribute to any p_k, but its successors can, provided that they are reached within x; according to this, the successors of Σ that are reached within x are added to Γ.
- The successors of a state in a class $\varphi_1 \wedge \varphi_2$ (line 10) can contribute to $H_{ik}^{\varphi_1,\varphi_2}(\alpha,x)$ if the state is left before time α; according to this, the successors of Σ that are reached before α are added to Γ.

Overall, for each regeneration condition i, the transient tree enumeration is performed:

- $\frac{\beta}{h} + 1$ times to compute $L_i^{\varphi_1,\varphi_2}(t_a, t_b)$ for $a = 0$ and $b = 0, \ldots, \bar{b} - \bar{a}$, and for $a = 1, \ldots, \bar{a}$ and $b = a + \bar{b} - \bar{a}$;

- once for the evaluation of the global kernel values $dG_{ik}^{\varphi_1}(t_m)$ for $m = 1, \ldots, \bar{a}$ and all k;
- $(\frac{\alpha}{h} + 1)(\frac{\beta-\alpha}{h} + 2)$ times for the evaluation of the conditional global kernel values $H_{ik}^{\varphi_1,\varphi_2}(t_a, t_m)$ for $a = 0, 1, \ldots, \bar{a}$, $m = a - 1, \ldots, a + \bar{b} - \bar{a}$, and all k.

If R is the number of reachable regeneration conditions, the number of transient tree enumerations is thus $O(\frac{\alpha}{h}\frac{\beta-\alpha}{h}R)$. The advantage with respect to [25], [27], where model checking is performed with a single transient tree enumeration, lies in the reduced depth of these transient trees: the computation of successors for the leaves of the tree now halts not only on $\neg\varphi_1$ classes, but also on regenerative ones. Notably, both the worst case space and time required for the computation of a successor class grow exponentially with the depth of the predecessor in the tree [29]; when the time bound β is large and regenerations are reached in a limited number of discrete events, the repeated enumeration of shallow trees becomes extremely beneficial.

4.3 Eliminating the lower bound α

The availability of deterministic transitions in STPNs can be leveraged to remove the lower bound α for the satisfaction of φ_2, reducing the evaluation of an interval until operator to a first-passage problem. We present this alternative approach by discussing the effect of $\alpha = 0$ on Eq. (8), and then the negative consequences of extending the model with an additional transition with deterministic duration equal to α.

The complexity of Eq. (8) is largely reduced if the until operator does not restrict the minimum time for the acceptance of the goal condition φ_2, i.e., if $\alpha = 0$. In this case, both Eq. (8) and its kernels are simplified: the local kernel $L_i^{\varphi_1,\varphi_2}(\alpha, \beta)$ becomes the probability that, starting from the regenerative state i, a φ_2-state is encountered before the first regeneration and not later than β. Moreover, the second term of Eq. (8) gives a null contribution. Finally, the conditional global kernel $H^{\varphi_1,\varphi_2}(\alpha, x)$ becomes the probability that a regeneration k is reached before x after visiting only states that satisfy $\varphi_1 \wedge \neg\varphi_2$.

The two kernels $L_i^{\varphi_1,\varphi_2}(0, \beta)$ and $H^{\varphi_1,\varphi_2}(0, x)$ can be derived from the transient trees rooted in regenerative classes reached within the first regeneration epoch, not later than β and through executions that visit only classes satisfying $\varphi_1 \wedge \neg\varphi_2$: the local kernel $L_i^{\varphi_1,\varphi_2}(0, \beta)$ is derived from the transient tree rooted in a class with regeneration condition i and limited to the first regeneration, or to time β, or to the first conclusive state that satisfies φ_2 or $\neg\varphi_1$; finally, $H_{ik}^{\varphi_1,\varphi_2}(0, x)$ is derived through the analysis of behaviors that reach the first regeneration within time β, visiting only states that satisfy $\varphi_1 \wedge \neg\varphi_2$.

This construction applies the strategy of [12] to the context of non-Markovian processes. In fact, restrictions made in the enumeration of transient trees correspond to manipulations performed on the underlying stochastic process to turn any state that satisfies φ_2 or $\neg\varphi_1$ into an absorbing state (i.e., disabling any transition).

The case $[0, \beta]$ can be lifted to solve the case $[\alpha, \beta]$ by exploiting the ability of STPNs to represent DET transitions. Following the same principle of techniques that reduce probabilistic model checking to the analysis of a synchronous composition of the model with a specification automaton [15], [16], the STPN model can be extended with a DET transition t with density $f_t(x) = \delta(x - \alpha)$ and φ_2 can be restricted to $\varphi_2' = \varphi_2 \wedge \{t \text{ has fired}\}$. In so doing, regenerations before α of the original model are not exploited, since the additional DET transition t is not "memoryless". Only after the firing of t at time α, regenerations will be fully exploited in the analysis. This approach is thus well-suited only for cases with a small α with respect to the duration of regeneration epochs.

Example 4. The property of Example 2 (Section 2.3) is not satisfied. In fact, the measure of paths satisfying $\varphi_1 \mathcal{U}^{[\alpha,\beta]} \varphi_2$ from the initial marking *2free operational* with $\varphi_1 = (buffer < 2)$ and $\varphi_2 = (failed = 1)$ is $0.3313 < 0.4$ for $\alpha = 0$ and $\beta = 7$, and $0.2359 > 0.2$ for $\alpha = 2.5$ and $\beta = 7$. In the latter case, when limited to φ_1-markings, the model can reach only 3 distinct regenerations before time $\beta = 7$ and the corresponding transient trees include a total of 44 classes. In contrast, if a transition with deterministic value $\alpha = 2.5$ is added to the model, a total of 130 classes need to be enumerated. In particular, the transient tree enumerated from the initial regeneration includes 115 classes: this larger number is a consequence of the deterministic timer added to the initial state, which results in a larger number of transition firings required to reach the first regeneration.

5 Fischer's Mutual Exclusion Protocol

5.1 Model definition

We illustrate the proposed technique on a stochastic model of n concurrent processes $P_1, P_2, \ldots, P_n$ accessing a critical section with Fischer's protocol [40]. The protocol ensures mutual exclusion using atomic read and write operations on a shared communication variable id taking the values $0, 1, \ldots, n$. When $id = 0$, each process P_i can attempt the access to the critical section. To this end, it performs the (time-consuming) write operation $id \leftarrow i$, waits for a time not lower than W, and then reads id again: if $id = i$, it can access the critical section and write $id \leftarrow 0$ on exit; whereas, if $id \neq i$, it has to wait until $id = 0$ to attempt again.

Fischer's protocol is a typical benchmark for real-time model checking, as it neatly illustrates the interaction between concurrency and firm timing: mutual exclusion is guaranteed provided that the waiting time W is greater than the maximum time required by the write operation of any process. This condition inherently requires a model with multiple concurrent timers with upper and lower bounds. While the protocol has been verified in the qualitative perspective using real-time model checkers such as Kronos [41] and Uppaal [42], randomized versions have been analyzed in closed-form [43] or through simulation [44], [45] only with timed activities modeled through exponential or gamma distributions. In this case, due to unbounded PDF supports, mutual exclusion can be violated with probability greater than zero.

We analyze quantitative properties in a stochastic model of the protocol enforcing the requirement of mutual exclusion with certainty through the use of concurrent GEN timers with bounded supports. Fig. 8 illustrates an STPN model with three processes P_1, P_2, P_3 (the same scheme can be extended

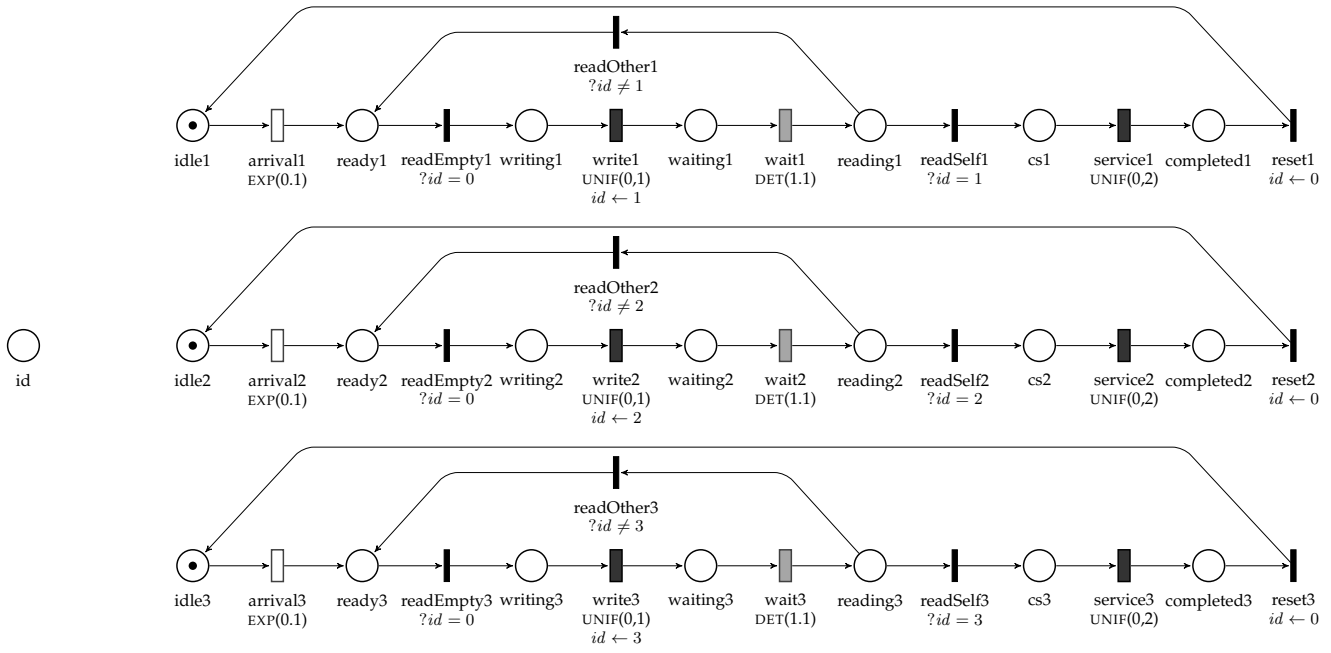


Fig. 8: STPN model of three processes accessing a critical section with Fischer's mutual exclusion protocol.

to any number of processes). The shared variable is encoded by the marking of place id (initially empty). Each process P_i eventually leaves its idle state through transition $arrival_i$ (EXP with rate 0.1), and enters the contention by reaching place $writing_i$ as soon as $id = 0$ (IMM transition $readEmpty_i$ with enabling function $?id = 0$); it then sets the shared variable to its own identifier (as specified by the update function $id \leftarrow i$) at the end of a write operation (transition $write_i$, with duration uniformly distributed over $[0, 1]$), and sojourns in a waiting state (place $waiting_i$) for a time greater than the maximum time that any process can spend writing to id (transition $wait_i$, DET equal to 1.1). When the wait completes, process P_i reads id again to ensure that its write was the last one (place $reading_i$): if $id \neq i$, the control goes back to the initial state of contention $ready_i$ (IMM transition $readOther_i$); whereas, if the shared variable is still equal to the process identifier (i.e., $id = i$), P_i enters the critical section cs_i (IMM transition $readSelf_i$), performs its service (transition $service_i$, uniform over $[0, 2]$), and then resets the shared variable (IMM transition $reset_i$), returning idle.

5.2 Quantitative evaluation

We consider a deadline requirement prescribing that the latency for the access of P_1 to the critical section be (1) not higher than β (which we call base deadline) with probability greater than p, and (2) not higher than $\beta_r > \beta$ (which we call relaxed deadline) probability greater than $p_r > p$. This property can be encoded as the Boolean conjunction of two *probabilistic existence* properties [1]:

$$P_{>p}[\,\text{TRUE}\ \mathcal{U}^{[0,\beta]}(cs_1 = 1)\,] \wedge P_{>p_r}[\,\text{TRUE}\ \mathcal{U}^{[0,\beta_r]}(cs_1 = 1)\,]. \quad (11)$$

Fig. 9 reports the measure $Pr_{(m,\vec{0})}\{\omega \in \Omega_m \mid \omega \models \text{TRUE}\ \mathcal{U}^{[0,\beta]}(cs_1 = 1)\}$ as a function of β, for $m \in \{m_A, m_B, m_C\}$ where: $m_A = ready_1\, idle_2\, idle_3$ (which occurs when P_1 becomes ready while the other processes are idle), $m_B = 3id\, ready_1\, idle_2\, waiting_3$ (which occurs when P_1 becomes ready and P_3 has just set the shared variable, closing the access to the contention), $m_C = 3id\, ready_1\, writing_2\, waiting_3$ (which occurs when P_1 becomes ready while both P_2 and P_3 are in the contention, with P_2 writing to id and P_3 waiting to check id after a write operation). As intuitive, the latency of P_1 increases when the initial condition changes from m_A to m_B, and then from m_B to m_C. Properties in the form of Eq. (11) are decided by comparing the probability computed for a given value of β with the threshold p. For example, with initial condition m_A, for $\beta = 2$, $p = 0.90$, $\beta_r = 6$ and $p_r = 0.95$, we conclude from Fig. 9 that $P_{>p}[\,\text{TRUE}\ \mathcal{U}^{[0,\beta]}(cs_1 = 1)\,] = \text{FALSE}$ and $P_{>p_r}[\,\text{TRUE}\ \mathcal{U}^{[0,\beta_r]}(cs_1 = 1)\,] = \text{TRUE}$. The relaxed deadline is thus met with the required probability, but the base

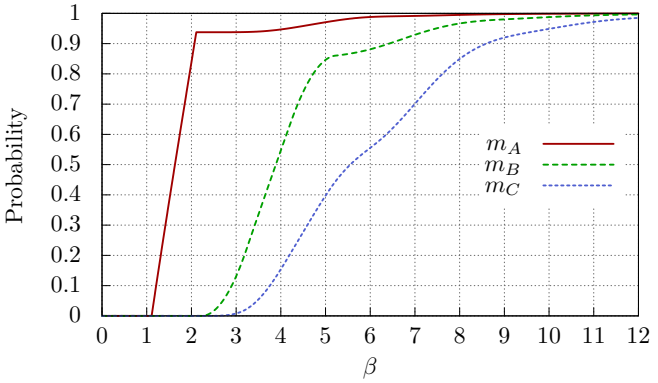


Fig. 9: The probability of paths satisfying $\text{TRUE}\ \mathcal{U}^{[0,\beta]}(cs_1 = 1)$ as a function of β, for markings $m_A = ready_1\, idle_2\, idle_3$, $m_B = 3id\, ready_1\, idle_2\, waiting_3$, $m_C = 3id\, ready_1\, writing_2\, waiting_3$.

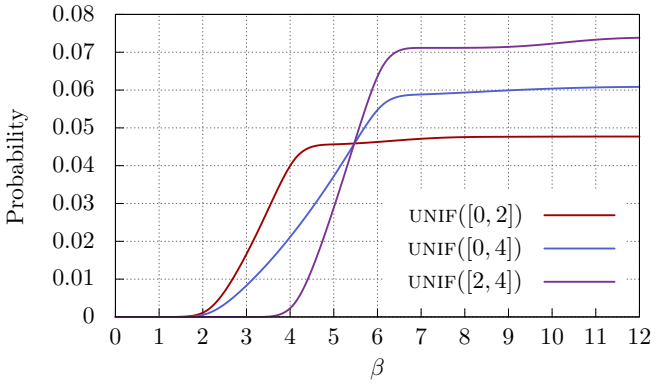


Fig. 10: The probability of paths satisfying $(cs_1 = 0)\ \mathcal{U}^{[0,\beta]}(\sum_{i=2}^{n} completed_i = 1 \wedge \sum_{i=2}^{n} ready_i \geq k)$ as a function of β, from marking $m_A = ready_1\, idle_2\, idle_3$, and different service times.

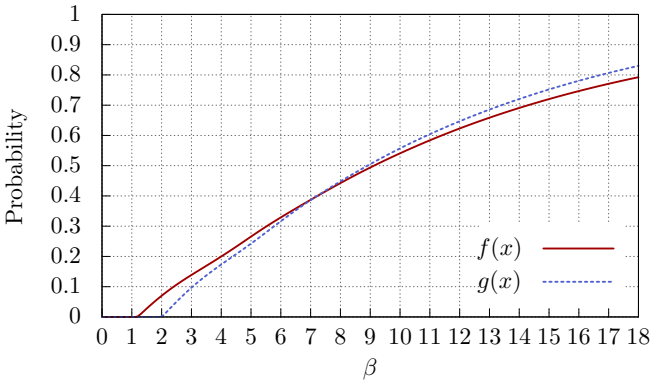


Fig. 11: The probability measure of paths satisfying TRUE $\mathcal{U}^{[0,\beta]}(cs_1 = 1)$ from $m_0 = idle_1\, idle_2\, idle_3$ when $write_1$ is distributed according to truncated Erlang PDF $f(x) = xe^{-20x}/400$ over $[0, 1]$ (mean value 0.1) or its symmetrical $g(x) = f(1 - x)$ (mean value 0.9).

deadline is not.

In order to understand the role of different design parameters in the overall distribution of latency, the probability measure of subsets of successful paths can be evaluated with additional until properties. For instance, the *probabilistic until* pattern [1]

$$P_{>p}[\,(\textstyle\sum_{i\neq 1} cs_i = 0)\ \mathcal{U}^{[0,\beta]}(cs_1 = 1)\,] \tag{12}$$

evaluated from the initial marking m_A formulates a requirement on the measure of probability of the set of behaviors where P_1 is the first process accessing the critical section. In a practical perspective, this property expresses a bound p on the probability that P_1 is not overtaken in the access to the critical section by some process that was initially idle. The corresponding probability measure is determined by the trade-off between the rapidity of P_1 in completing the write operation (and thus preventing the access of other processes to contention) and the number $n - 1$ and rate λ with which other processes enter the ready state. Results of the evaluation show that the probability of no-overtaking depends on the total load $(n-1)\,\lambda$, but it is relatively immune to the number of processes that produce it. For instance, if $(n - 1)\,\lambda$ is kept equal to 0.2 (to 0.1) while varying $n - 1$ from 2 to 8, the probability of no-overtaking remains equal to 0.93 (to 0.96) with a variation lower than 0.001.

When P_1 is overtaken, the service time of the overtaking process plays a twofold role: it determines the time that P_1 must wait before the next attempt, and it also determines the probability that more processes leave the idle state. The latter factor requires that service time be kept sufficiently low with respect to the total load of the system. For a quantitative assessment of this property, we can formulate a requirement on the maximum probability that at least k processes become ready during the first failed attempt of P_1:

$$P_{<p}[\,(cs_1 = 0)\ \mathcal{U}^{[0,\beta]}(\textstyle\sum_{i=2}^{n} completed_i = 1 \wedge \sum_{i=2}^{n} ready_i \geq k)\,]. \tag{13}$$

Fig. 10 reports the probability measure associated with this property (as function of β, from the initial state m_A) for three service time distributions when $n = 3$ and $k = 1$.

While the impact of service times is intuitive, the role of writing time distributions is twofold: due to the *last-write-wins* policy of Fischer's protocol, a lower writing time favors P_1 in keeping concurrent processes out of the contention, but in case of contention a greater writing time will favor P_1 in being the last process that completes its write to id, and thus the first to enter the critical section. To give a quantitative insight into this mechanism, we consider a setting in which the writing times of P_2 and P_3 are distributed uniformly over $[0, 1]$ (mean value 0.5), while the writing time of P_1 has either a truncated Erlang PDF $f(x) = xe^{-20x}/400$ over $[0, 1]$ (mean value 0.1) or its symmetrical $g(x) = (1 - x)e^{-20(1-x)}/400$ over $[0, 1]$ (mean value 0.9). Fig. 11 shows that, for $\beta < 7$, a faster writing time PDF $f(x)$ results in a higher probability that P_1 will reach the critical section from the initial marking $m_0 = ready_1\, ready_2\, ready_3$, while the slower PDF $g(x)$ is advantageous when $\beta > 7$. This result captures the following intuition: while the lower mean value of f favors process P_1 in the first attempt, the greater mean value of g makes P_1 more competitive in trials subsequent to an initial overtaking; until time 7, the gain in the first attempt prevails, but after time 7, the competitive advantage in subsequent trials becomes more relevant. In this perspective, it is worth noting that the unbiased distribution with mean value 0.5 is always worse than one of the two biased distributions f and g.

As a last example, we evaluate the probability that process P_1 is in the critical section within a given time window $[\alpha, \beta]$ after an execution in which P_3 has never accessed the critical section. This property might be of interest in a problem of real-time testing where the system can be observed only within an interval $[\alpha, \beta]$ and the test case requires P_1 in the critical section without prior accesses of P_3. The requirement can be formulated as the probabilistic interval until

$$P_{>p}[\,(cs_3 = 0)\ \mathcal{U}^{[\alpha,\beta]}(cs_1 = 1)\,] \tag{14}$$

and verified for given values of α, β and p so as to determine the time α at which it is best to start the observation, or the minimum duration of $\beta - \alpha$ to obtain a probability of conclusive execution greater than a given threshold p. Fig. 12 plots the probability measure of paths satisfying $(cs_3 =$

0) $\mathcal{U}^{[\alpha,\beta]}(cs_1 = 1)$ for different values of α and duration $\beta - \alpha$ of the observation window. For each window size $\beta - \alpha \in \{0.1, 0.5, 1.0, 1.5, 2.0\}$, the probability $p_{(m_0,\vec{0})}(\alpha, \beta) := Pr_{(m_0,\vec{0})}\{\omega \in \Omega_{m_0} \mid \omega \models (cs_3 = 0)\ \mathcal{U}^{[\alpha,\beta]}(cs_1 = 1)\}$ is plotted in Fig. 12 for $\alpha = 0, 0.1, \ldots, 4$, which might serve to select an optimum time α to start the observation.

5.3 Computational advantage of regenerative analysis

In the construction of Fig. 12, for each value δ of $\beta - \alpha$, the measures $p_{(m_0,\vec{0})}(\alpha, \alpha+\delta)$ for $\alpha = 0, 0.1, \ldots, 4$ are computed as the by-product of a single solution of Eq. (8) for the evaluation of $p_{(m_0,\vec{0})}(4, 4+\delta)$ with step 0.1. For this measure, we identify 17 distinct regeneration conditions reachable under $\varphi_1 = (cs_3 = 0)$, whose transient trees include 904 stochastic state classes. For $\alpha = 4$ and $\delta = 1$, adopting a step size $h = 0.1$, the enumeration of each tree is repeated $\frac{5}{0.1} + 1 = 51$ times for the evaluation of $L_i^{\varphi_1,\varphi_2}$, once for the evaluation of $dG_{ik}^{\varphi_1}(t_m)$, and $(\frac{4}{0.1} + 1)(\frac{1}{0.1} + 2) = 492$ times for the evaluation of $H_{ik}^{\varphi_1,\varphi_2}$, resulting in $491,776$ enumerated classes.

As a comparison, the reduction to transient analysis through a deterministic timer $\alpha = 4$ described in Section 4.3 requires more than 8 million classes to be enumerated; of these, more than 99% belong to the enumerations of the initial transient tree, in which a regeneration is reached only after the elapse of α. Note that the approaches of [27], [25] would necessarily incur in this exponential complexity by enumerating the times to fire PDFs after each discrete event along any feasible path until $\alpha + \delta$; in the approach of [25], this complexity would be further exacerbated by the finer grain of regions with respect to zones.

Fig. 13 highlights the reduction of complexity achieved by the proposed regenerative approach. A transient analysis eliminating the lower bound α through the inclusion of a deterministic transition forgoes all regeneration points occurring before α, forcing the enumeration of each discrete event along any feasible path within α and producing the complexity shown by the dotted line. Whereas, in the regenerative approach based on the bivariate Markov renewal equations of Eq. (8), the total number of enumerated classes follows the solid line.

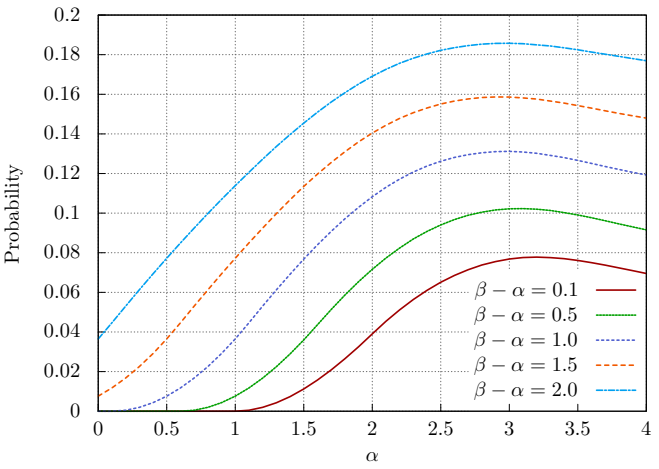


Fig. 12: The probability measure of paths satisfying $(cs_3 = 0)\ \mathcal{U}^{[\alpha,\beta]}(cs_1 = 1)$ from marking $m_0 = idle_1\, idle_2\, idle_3$.

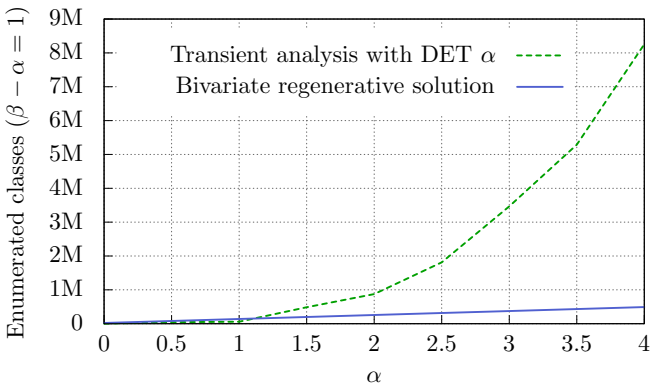


Fig. 13: Enumerated classes in the computation of the measure of paths satisfying $(cs_3 = 0)\ \mathcal{U}^{[\alpha,\beta]}(cs_1 = 1)$ from marking $m_0 = idle_1\, idle_2\, idle_3$ when $\beta - \alpha = 1$.

6 Conclusions

The verification of an interval until operator $\varphi_1\, \mathcal{U}^{[\alpha,\beta]}\, \varphi_2$ in regenerative systems presents major challenges, both theoretical and practical, that cannot leverage existing approaches for CTMCs [12] nor established results of Markov renewal theory [18], [38].

Stochastic models with concurrent GEN timers "accumulate memory" over time: the state at time α does not summarize, in general, the past evolution of the system, and the process cannot be verified independently before and after α, in contrast to CTMCs, in which every time instant, and thus α, is a regeneration point. On the other hand, the reduction to a first-passage analysis problem requires the introduction of a deterministic timer in order to account for the minimum time α for the satisfaction of φ_2. Unfortunately, this approach crucially affects regenerative transient analysis [35]: it is now the deterministic timer that carries memory until its elapse, in order to characterize the state distribution of the system at time α. Regeneration points before α are thus inevitably lost, forcing the enumeration of all sequences of discrete events before α.

We provided a solution based on the bivariate extension of Markov renewal equations, explicitly accounting for a satisfaction interval $[\alpha, \beta]$. The result is based on the formal definition of the probability space of STPN paths, which established the theoretical relation between cylinder sets of paths and stochastic state classes [35]. Enumeration of stochastic state classes was in turn the basis for algorithms computing kernels of bivariate Markov renewal equations.

The computation of the kernels requires to repeat the enumeration of stochastic state classes from each regeneration condition for a number of times linear in $\alpha(\beta - \alpha)$, but each enumeration is limited to the first regeneration epoch and regeneration points are exploited also before α. In so doing, since the number of feasible events grows exponentially with the time bound, repeated enumeration of a limited number of shallow trees can produce considerable benefits when the stochastic process has a repetitive structure and multiple regenerations can be traversed before reaching the time bound β. Moreover, the enumeration is always restricted to

paths satisfying the safety condition φ_1, and the underlying stochastic process needs to encounter regeneration points w.p.1 in a bounded number of events only on paths that always satisfy φ_1.

The benefits of the approach were demonstrated by a preliminary implementation in the analysis of a probabilistic model of Fischer's mutual exclusion protocol, a typical benchmark for real-time model checking. Notably, quantitative properties were analyzed in a stochastic model that guarantees the correctness of the protocol due to generally distributed timers with bounded and deterministic supports. The construction of these results highlighted important problems of the probabilistic model checking of transient properties in regenerative systems, which can serve as the basis for further analysis techniques.

REFERENCES

[1] L. Grunske, "Specification patterns for probabilistic quality properties," in *ICSE'08*. ACM, May 2008, pp. 31–40.

[2] H. L. Younes and R. G. Simmons, "Probabilistic Verification of Discrete Event Systems Using Acceptance Sampling," in *Computer Aided Verification*, ser. LNCS, vol. 2404. Springer, 2002, pp. 223–235.

[3] H. L. Younes, M. Kwiatkowska, G. Norman, and D. Parker, "Numerical vs. statistical probabilistic model checking," *Int. J. on Softw. Tools for Techn. Transfer*, vol. 8, no. 3, pp. 216–228, 2006.

[4] G. Behrmann, A. David, K. G. Larsen, P. Pettersson, and W. Yi, "Developing uppaal over 15 years," *Softw. Pract. Exper.*, vol. 41, no. 2, pp. 133–142, Feb. 2011.

[5] K. Sen, M. Viswanathan, and G. Agha, "On statistical model checking of stochastic systems," in *Computer Aided Verification*, ser. LNCS, vol. 3576. Springer, 2005, pp. 266–280.

[6] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: verification of probabilistic real-time systems," in *Computer Aided Verification*, ser. LNCS, vol. 6806. Springer, 2011, pp. 585–591.

[7] J.-P. Katoen, I. S. Zapreev, E. M. Hahn, H. Hermanns, and D. N. Jansen, "The ins and outs of the probabilistic model checker MRMC," *Perform. Eval.*, vol. 68, no. 2, pp. 90–104, Feb. 2011.

[8] A. Clark, S. Gilmore, J. Hillston, and M. Tribastone, "Stochastic process algebras," in *Formal Methods for Performance Evaluation*, ser. LNCS. Springer, 2007, vol. 4486, pp. 132–179.

[9] G. Ciardo, R. L. Jones, III, A. S. Miner, and R. I. Siminiceanu, "Logic and stochastic modeling with SMART," *Perform. Eval.*, vol. 63, no. 6, pp. 578–608, Jun. 2006.

[10] H. L. Younes and R. G. Simmons, "Statistical probabilistic model checking with a focus on time-bounded properties," *Information and Computation*, vol. 204, no. 9, pp. 1368–1409, 2006.

[11] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton, "Model-checking continuous-time Markov chains," *ACM Trans. Comput. Logic*, vol. 1, no. 1, pp. 162–170, Jul. 2000.

[12] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen, "Model-checking algorithms for continuous-time Markov chains," *IEEE Trans. Softw. Eng.*, vol. 29, no. 6, pp. 524–541, Jun. 2003.

[13] M. Kwiatkowska, G. Norman, and D. Parker, "Probabilistic symbolic model checking with PRISM: A hybrid approach," *Int. J. Softw. Tools Technol. Transf.*, vol. 6, no. 2, pp. 128–142, Aug. 2004.

[14] K. S. Trivedi and R. Sahner, "SHARPE at the Age of Twenty Two," *SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 4, pp. 52–57, Mar. 2009.

[15] S. Donatelli, S. Haddad, and J. Sproston, "Model checking timed and stochastic properties with CSL^{TA}," *IEEE Trans. Softw. Eng.*, vol. 35, no. 2, pp. 224–240, Mar. 2009.

[16] T. Chen, T. Han, J.-P. Katoen, and A. Mereacre, "Quantitative model checking of continuous-time Markov chains against timed automata specifications," in *LICS'09*, Aug. 2009, pp. 309–318.

[17] G. Ciardo, R. German, and C. Lindemann, "A characterization of the stochastic process underlying a stochastic Petri net," *IEEE Trans. Softw. Eng.*, vol. 20, no. 7, pp. 506–515, Jul. 1994.

[18] E. Çinlar, "Markov renewal theory: A survey," *Management Science*, vol. 21, no. 7, pp. 727–752, 1975.

[19] G. Infante-López, H. Hermanns, and J.-P. Katoen, "Beyond memoryless distributions: Model checking semi-Markov chains," in *PAPM-PROBMIV'01*, ser. LNCS, vol. 2165. Springer, 2001, pp. 57–70.

[20] J. Bryans, H. Bowman, and J. Derrick, "Model checking stochastic automata," *ACM Trans. Comput. Logic*, vol. 4, no. 4, pp. 452–492, Oct. 2003.

[21] H. Choi, V. G. Kulkarni, and K. S. Trivedi, "Markov regenerative stochastic Petri nets," *Perform. Eval.*, vol. 20, no. 1-3, pp. 337–357, May 1994.

[22] R. German and C. Lindemann, "Analysis of stochastic Petri nets by the method of supplementary variables," *Perform. Eval.*, vol. 20, no. 1-3, pp. 317–335, May 1994.

[23] J. M. Martinez and B. R. Haverkort, "CSL Model Checking of Deterministic and Stochastic Petri Nets," in *MMB'06 13th GI/ITG Conference*. IEEE CS, 2006, pp. 1–18.

[24] A. Bobbio and M. Telek, "Markov regenerative SPN with non-overlapping activity cycles," in *IPDS'95*. IEEE, 1995, pp. 124–133.

[25] R. Alur and M. Bernadsky, "Bounded model checking for GSMP models of stochastic real-time systems," in *Hybrid systems: computation and control*, ser. LNCS, vol. 3927. Springer, 2006, pp. 19–33.

[26] R. Alur and D. L. Dill, "A theory of timed automata," *Theor. Comput. Sci.*, vol. 126, no. 2, pp. 183–235, Apr. 1994.

[27] A. Horváth, M. Paolieri, L. Ridi, and E. Vicario, "Probabilistic model checking of non-Markovian models with concurrent generally distributed timers," in *QEST'11*. IEEE CS, 2011, pp. 131–140.

[28] G. Bucci, R. Piovosi, L. Sassoli, and E. Vicario, "Introducing probability within state class analysis of dense time dependent systems," in *QEST'05*. IEEE CS, Sep. 2005, pp. 13–22.

[29] L. Carnevali, L. Grassi, and E. Vicario, "State-density functions over DBM domains in the analysis of non-Markovian models," *IEEE Trans. Softw. Eng.*, vol. 35, no. 2, pp. 178–194, Mar. 2009.

[30] E. Vicario, L. Sassoli, and L. Carnevali, "Using stochastic state classes in quantitative evaluation of dense-time reactive systems," *IEEE Trans. Softw. Eng.*, vol. 35, no. 5, pp. 703–719, Sep./Oct. 2009.

[31] D. L. Dill, "Timing assumptions and verification of finite-state concurrent systems," in *AVMFSS'89*, ser. LNCS, vol. 407. Springer, 1990, pp. 197–212.

[32] K. S. Trivedi, *Probability and statistics with reliability, queuing, and computer science applications*. New York: John Wiley and Sons, 2001.

[33] W. H. Sanders and J. F. Meyer, "Stochastic Activity Networks: Formal Definitions and Concepts," in *Lectures on Formal Methods and Performance Analysis*, ser. LNCS. Springer, 2001, vol. 2090, pp. 315–343.

[34] P. Ballarini, N. Bertrand, A. Horváth, M. Paolieri, and E. Vicario, "Transient Analysis of Networks of Stochastic Timed Automata using Stochastic State Classes," in *QEST'13*, ser. LNCS, vol. 8054. Springer, 2013, pp. 355–371.

[35] A. Horváth, M. Paolieri, L. Ridi, and E. Vicario, "Transient analysis of non-Markovian models using stochastic state classes," *Perform. Eval.*, vol. 69, no. 7-8, pp. 315–335, Jul. 2012.

[36] L. Carnevali, L. Ridi, and E. Vicario, "A framework for simulation and symbolic state space analysis of non-Markovian models," in *SAFECOMP'11*, ser. LNCS, vol. 6894. Springer, 2011, pp. 409–422.

[37] A. Puliafito, M. Scarpa, and K. S. Trivedi, "Petri nets with k simultaneously enabled generally distributed timed transitions," *Perform. Eval.*, vol. 32, no. 1, pp. 1–34, 1998.

[38] V. Kulkarni, *Modeling and analysis of stochastic systems*. Chapman & Hall, 1995.

[39] H. Brunner and P. van der Houwen, *The numerical solution of Volterra equations*. North-Holland Amsterdam, 1986, vol. 268.

[40] N. Lynch and N. Shavit, "Timing-based mutual exclusion," in *Real-Time Systems Symposium, 1992*. IEEE, 1992, pp. 2–11.

[41] C. Daws, A. Olivero, S. Tripakis, and S. Yovine, "The tool Kronos," in *Hybrid Systems III*, ser. LNCS. Springer, 1996, vol. 1066, pp. 208–219.

[42] J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson, and W. Yi, "UPPAAL – A tool suite for automatic verification of real-time systems," in *Hybrid Systems III*, ser. LNCS. Springer, 1996, vol. 1066, pp. 232–243.

[43] E. Gafni and M. Mitzenmacher, "Analysis of timing-based mutual exclusion with random times," *SIAM Journal on Computing*, vol. 31, no. 3, pp. 816–837, 2001.

[44] J.-P. Katoen, H. Bohnenkamp, R. Klaren, and H. Hermanns, "Embedded software analysis with MOTOR," in *Formal Methods for the Design of Real-Time Systems*. Springer, 2004, pp. 268–293.

[45] D. D. Deavours, G. Clark, T. Courtney, D. Daly, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. G. Webster, "The Mobius framework and its implementation," *IEEE Trans. Softw. Eng.*, vol. 28, no. 10, pp. 956–969, Oct. 2002.

APPENDIX A
CALCULUS OF STOCHASTIC STATE CLASSES

The succession relation $\xRightarrow{\gamma,I,\mu}$ among stochastic state classes can be enumerated through symbolic operations on the support and probability density function (PDF) of vectors of random variables. While a complete presentation is in [30], [29], we report the main steps required by the calculus.

A stochastic state class $\Sigma = \langle m, D, f_{\langle \tau_{age}, \vec{\tau} \rangle} \rangle$ includes the marking $m \in \mathbb{N}^P$, and the joint PDF $f_{\langle \tau_{age}, \vec{\tau} \rangle}$ and support $D \subseteq \mathbb{R}^{n+1}$ of the random vector $\langle \tau_{age}, \vec{\tau} \rangle$, where:

- the random variable τ_{age} encodes the negation of the time of the previous firing;
- the components of $\vec{\tau} = (\tau_1, \ldots, \tau_n)$ encode the times to fire of enabled transitions $E(m)$.

As detailed in Remark 1, the initial stochastic state class corresponding to a regeneration $(m, \vec{d})$ can be constructed for an STPN $\langle P, T, A^-, A^+, B, U, EFT, LFT, F, W \rangle$ from the marking m, the enabling times $\vec{d}$, and the sampling PDFs f_t of enabled transitions $t \in E(m)$, where f_t is the PDF of the distribution $F(t)$. Initially, the components of the vector $\langle \tau_{age}, \vec{\tau} \rangle$ are independent random variables and their joint PDF $f_{\langle \tau_{age}, \vec{\tau} \rangle}$ is in product form.

Given a stochastic state class $\Sigma = \langle m, D, f_{\langle \tau_{age}, \vec{\tau} \rangle} \rangle$, a transition $\gamma \in E(m)$ and a firing interval I, the probability μ that γ fires in Σ at some time in I can be computed as

$$\mu = \int_{D_a} \frac{W(t_1)}{\sum_{i:x_i=x_1} W(t_i)} f_{\langle \tau_{age}, \vec{\tau} \rangle}(x_{age}, \vec{x}) \, dx_{age} \, d\vec{x} \quad (15)$$

where (without loss of generality) enabled transitions corresponding to the times to fire $\tau_1, \ldots, \tau_n$ are indicated as $t_1, \ldots, t_n$ with $t_1 = \gamma$, and

$$D^a = \{\langle x_{age}, \vec{x} \rangle \in D \mid (\forall j > 1 . x_1 \leq x_j) \wedge (x_1 - x_{age}) \in I\}$$

is the set of values for $\langle \tau_{age}, \vec{\tau} \rangle$ such that the time to fire of t_1 is minimum, and the time of the next firing (obtained increasing $-\tau_{age}$ by the sojourn time τ_1) is contained in I. By integrating the PDF $f_{\langle \tau_{age}, \vec{\tau} \rangle}(x_{age}, \vec{x})$ over this set of values, we obtain the succession probability μ. The ratio of weights in Eq. (15) accounts for the case in which τ_1 is deterministic and other deterministic times to fire attain the minimum value: according to the semantics of STPNs, the probability that t_1 is selected is $W(t_1) / \sum_{i:x_i=x_1} W(t_i)$.

The unique successor class Σ' such that $\Sigma \xRightarrow{\gamma,I,\mu} \Sigma'$ can be derived as follows. The marking m' is computed by removing one token from each input place of t_1, adding one token to each output place of t_1, and applying the update function $U(t_1)$ to the resulting marking. The support D' and PDF $f_{\langle \tau'_{age}, \vec{\tau}' \rangle}$ of the vector of times to fire $\langle \tau'_{age}, \vec{\tau}' \rangle$ immediately after the firing of t_1 (and conditioned on this event) is computed in four steps.

Precedence conditioning. The assumption that t_1 fires at some time in I before any other transition yields a new random vector $\langle \tau^a_{age}, \vec{\tau}^a \rangle$ distributed over D^a according to $f_{\langle \tau^a_{age}, \vec{\tau}^a \rangle}(x_{age}, \vec{x}) = f_{\langle \tau_{age}, \vec{\tau} \rangle}(x_{age}, \vec{x}) / \mu$.

Time advancement and marginalization. When t_1 fires, τ_{age} and the times to fire $\tau^a_2, \ldots, \tau^a_n$ of enabled transitions are reduced by the random variable τ^a_1 associated with t_1. The time to fire of t_1 is eliminated from the joint PDF $f_{\langle \tau^a_{age}, \vec{\tau}^a \rangle}$ and from its support D^a by integrating the PDF over all values of τ^a_1. The PDF of the resulting random vector $\langle \tau^b_{age}, \vec{\tau}^b \rangle$ with $\vec{\tau}^b = (\tau^a_2 - \tau^a_1, \ldots, \tau^a_n - \tau^a_1)$ and $\tau^b_{age} = \tau^a_{age} - \tau^a_1$ is

$$f_{\langle \tau^b_{age}, \vec{\tau}^b \rangle}(x_{age}, x_2, \ldots, x_n) = \int_{L_1(x_{age}, \vec{x}^b)}^{U_1(x_{age}, \vec{x}^b)} f_{\langle \tau^a_{age}, \vec{\tau}^a \rangle}(x_{age} + x_1, x_1, x_2 + x_1, \ldots, x_n + x_1) dx_1$$

where $U_1(x_{age}, x_2, \ldots, x_n)$ and $L_1(x_{age}, x_2, \ldots, x_n)$ are the maximum and minimum value of x_1 such that $(x_{age}, x_1, \ldots, x_n) \in D^a$, respectively. As detailed in [29], U_1 and L_1 are piecewise functions; as a consequence, the PDF $f_{\langle \tau^b_{age}, \vec{\tau}^b \rangle}$ is a piecewise function defined on a partitioning of its support $D^b = \{(x_{age}, x_2, \ldots, x_n) \in \mathbb{R}^n \mid \exists x_1 \in \mathbb{R} : (x_{age}, x_1, \ldots, x_n) \in D^a\}$.

Disabling. Times to fire associated with transitions disabled during the firing (i.e., after removing tokens from input places of t_1, or after adding tokens to output places of t_1, or after applying the update function $B(t_1)$) are eliminated one at a time from the vector $\langle \tau^b_{age}, \vec{\tau}^b \rangle$ by integrating the PDF $f_{\langle \tau^b_{age}, \vec{\tau}^b \rangle}$ over all of their values in D^b. For example, the disabling of transition t_2 yields the random vector $\langle \tau^c_{age}, \vec{\tau}^c \rangle$ with $\vec{\tau}^c = (\tau^b_3, \ldots, \tau^b_n)$ that has PDF $f_{\langle \tau^c_{age}, \vec{\tau}^c \rangle}(x_{age}, \vec{x}) = \int_{L_2}^{U_2} f_{\langle \tau^b_{age}, \vec{\tau}^b \rangle}(x_{age}, x_2, x_3, \ldots, x_n) \, dx_2$ with support $D^c = \{(x_{age}, x_3, \ldots, x_n) \in \mathbb{R}^{n-1} \mid \exists x_2 \in \mathbb{R} : (x_{age}, x_2, x_3, \ldots, x_n) \in D^b\}$.

Newly enabling. Times to fire of transitions newly enabled after the firing (i.e., enabled by the new marking m', but not by m or by some intermediate marking) are added one at a time as independent random components to the vector $\langle \tau^c_{age}, \vec{\tau}^c \rangle$. For example, the newly enabling of transition t_{n+1} yields the random variable $\langle \tau'_{age}, \vec{\tau}' \rangle$ that has PDF $f_{\langle \tau'_{age}, \vec{\tau}' \rangle}(\tau_{age}, x_3, \ldots, x_n, x_{n+1}) = f_{\langle \tau^c_{age}, \vec{\tau}^c \rangle}(\tau_{age}, x_3, \ldots, x_n) f_{t_{n+1}}(x_{n+1})$ with support $D' = D^c \times [EFT(t_{n+1}), LFT(t_{n+1})]$.

Example 5. For the STPN of Fig. 1a, the successor of Σ_0 in Fig. 3 through the firing of transition *fail* (without constraints on the firing time) is computed as follows. The new marking *2free failed* is obtained from *2free operational* by removing one token from the input place *operational* and adding one token to the output place *failed*. The integral of the PDF $f_{\langle \tau_{age}, \vec{\tau} \rangle}(age, arrival, fail) = \delta(age) \, 0.1 \exp(-0.1 \, fail)$ over $D^a = \{\langle x_{age}, \vec{x} \rangle \in D \mid fail \leq arrival\} = \{\langle x_{age}, \vec{x} \rangle \in \mathbb{R}^3 \mid x_{age} = 0 \wedge (1 \leq arrival \leq 2) \wedge (fail \leq arrival)\}$ gives $\mu = 1 + 10 \, (e^{-0.2} - e^{-0.1}) \simeq 0.1389$ and $f_{\langle \tau^a_{age}, \vec{\tau}^a \rangle}(age, arrival, fail) = 0.7198 \exp(-0.1 \, fail)$. Time advancement and elimination of the fired timer *fail* give $f_{\langle \tau^b_{age}, \vec{\tau}^b \rangle}(age, arrival) = \int_0^2 \delta(age + fail) \, 0.7198 \exp(-0.1 \, fail) \, dfail = 0.7198 \exp(0.1 \, age)$ and

$$D^b = \{(age, arrival) \in \mathbb{R}^2 \mid (-2 \leq age \leq 0) \wedge (0 \leq arrival \leq 2) \wedge (-2 \leq age - arrival \leq -1)\}.$$

No other transition is disabled thus $f_{\langle \tau^c_{age}, \vec{\tau}^c \rangle}(age, arrival) = f_{\langle \tau^b_{age}, \vec{\tau}^b \rangle}(age, arrival)$ and $D^c = D^b$, while *restart* (uniformly distributed over $[1, 2]$) is newly enabled after the firing, resulting in $f_{\langle \tau'_{age}, \vec{\tau}' \rangle}(age, arrival, restart) = f_{\langle \tau^c_{age}, \vec{\tau}^c \rangle}(age, arrival)$ and $D' = D^c \times [1, 2]$ for the final vector of times to fire $\langle \tau'_{age}, \vec{\tau}' \rangle = \langle age, (arrival, restart) \rangle$.