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 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 model executions: 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 discrete events before the first regeneration, and the repetitive structure of the stochastic process is exploited also before the lower bound \(\alpha\), 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.


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 a 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 GEN variables, or real-time systems, in which correctness depends on firm time bounds resulting from worst-case execution times, minimum intervals between releases, deterministic durations 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...
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], each execution path reaching some goal state within time $\beta$ 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 in such states). 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 $\beta$. As a common limitation of [19] and [20], the SMP assumption that GEN timers are disabled or reset after each discrete event is still a significant hurdle 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 CTMCs subordinated to the activity intervals of GEN timers [21], [22], [23]. Nonetheless, the enabling restriction imposes a severe limitation, 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], [25], for which very few results of numerical evaluation were developed. Evaluation of (non-nested) probabilistic interval until operators with $\alpha = 0$ was proposed in [26] under the assumption of a bound on the number of discrete events feasible within time $\beta$: the solution is based on the enumeration of a tree of probability density functions (PDFs) of active timers with supports encoded as regions [27]. In [28], the assumption of a bound on the number of events within time $\beta$ is removed and fairly general conditions that guarantee termination in exact or approximate evaluation are provided, through the enumeration of stochastic state classes [29], [30], [31] representing the joint PDF of timers over Difference Bound Matrix zones [32], [33]. While a calculus based on zones largely reduces the branching factor of enumerated trees, the complexity of [26] and [28] grows exponentially with the length of event sequences within time $\beta$, and no computational advantage is gained 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 $\alpha$, or any state reached after $\beta$) 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 $\beta$, and with a major reduction of complexity when the stochastic process has a repetitive structure and multiple regenerations can be traversed before time $\beta$.

In Section 2, we introduce stochastic time Petri nets for the specification of stochastic models. A probability measure on model executions is defined by establishing a formal relation between cylinder sets (i.e., sets of execution paths that follow a common qualitative ordering of events and given timing restrictions) and stochastic state classes. We provide an algorithm for the detection of regeneration points during 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 activity durations using stochastic time Petri nets (STPNs) [29], [31]. An STPN is defined by a set of transitions, representing activities with stochastic duration, and a set of places; a marking assigns a natural 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 [34] or stochastic activity networks [35], 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 [36]). 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 \( B(t) : \mathbb{N}^P \to \{ \text{TRUE, FALSE} \} \) and with an update function \( U(t) : \mathbb{N}^P \to \mathbb{N}^P \).

In addition, for each transition \( t \in T \), the STPN specifies: an earliest firing time \( EFT(t) \in \mathbb{Q}_{\geq 0} \), a latest firing time \( LFT(t) \in \mathbb{Q}_{\geq 0} \cup \{ \infty \} \) 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 \( W(t) \in \mathbb{R}_{\geq 0} \).

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 \( \lambda \in \mathbb{R}_{\geq 0} \), 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 \( f_t(x) = \delta(y-x) \) with \( y = EFT(t) = LFT(t) \). For GEN transitions, 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]).
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 \( (\Omega_m, \mathcal{F}_m, P_{\text{f}}) \) induced by the semantics of STPN for a given initial marking \( m_0 \) and initial times to fire PDF \( f_\gamma \). The outcomes \( \Omega_m \) of the probability space are all paths \( \omega = s_0, s_1, s_2, \ldots \) with initial marking \( m_0 \). Note that \( \Omega_m \) may include paths that are not feasible under the initial distribution \( f_\gamma \) due to temporal constraints of the model; for these paths, the probability measure \( P_{\text{f}} \) will be zero.

To identify a \( \sigma \)-algebra \( \mathcal{F}_m \) of events over \( \Omega_m \), we define as cylinder set \( C(m_0, \gamma_0, I_1, \gamma_1, I_2, \ldots, \gamma_n, I_n) \) 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:

\[
C(m_0, \gamma_0, I_1, \gamma_1, I_2, \ldots, \gamma_n, I_n) := \{ \omega \in \Omega_m \mid |\omega| \geq n \text{ and } \forall 0 < k \leq n, (\omega[k] = \gamma_k \text{ and } T(k, \omega) \in I_k) \}
\]

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

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

We let \( T(0, \omega) := 0 \), so that \( T(k, \omega) \) also represents, for all \( 0 \leq k \leq |\omega| \), the time at which state \( s_k \) is reached in \( \omega \). 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 \( \mathcal{F}_m \) is defined as the smallest \( \sigma \)-algebra on \( \Omega_m \) that contains all the cylinder sets \( C(m_0, \gamma_0, I_1, \gamma_1, I_2, \ldots, \gamma_n, I_n) \) 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. \( \mathcal{F}_m \) 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 \( \mathcal{F}_m \) 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 \( \mathcal{F}_m \) 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 \( C(m_0, \gamma_1, I_1', \gamma_2, I_2', \ldots, \gamma_n, I_n') \) 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 \( C(m_0, \gamma_1, I_1 \cap I_1', \gamma_2, I_2 \cap I_2', \ldots, \gamma_n, I_n \cap I_n') \). Finally, the complement of \( C(m_0, \gamma_1, I_1, \gamma_2, I_2, \ldots, \gamma_n, I_n) \) corresponds to the finite union of all disjoint cylinder sets \( C(m_0, \gamma_1, I_1', \gamma_2, I_2', \ldots, \gamma_n, I_n') \) such that \( \exists i \leq n. (\gamma_i \neq \gamma'_i) \) and \( \forall i \leq n, I_i = [0, \infty) \), or \( \forall i \leq n, (\gamma_i = \gamma'_i) \) and, for each non-empty subset \( A \subseteq \{1, \ldots, n\} \), \( I_i = [0, \sup I_i, \infty) \) for all \( i \in A \).

The probability measure \( P_{\text{f}} \) can be expressed in terms of transient stochastic state classes [37] and, if GEN transitions are associated with piecewise exponential PDFs, it can be computed numerically using the Sisio package of the ORIS Tool [38].

Definition 3. A transient stochastic state class is a tuple \( \Sigma = (m, D, f_{\tau_{\text{age}}, \tau}) \) where \( m \in \mathbb{N}^D \) is a marking; \( f_{\tau_{\text{age}}, \tau} \) is the PDF immediately after a firing) of the random vector \( \langle \tau_{\text{age}}, \tau \rangle \) including the absolute time \( \tau_{\text{age}} \) (elapsed from the start of the process until the transition firing) and the times to fire \( \tau = (\tau_1, \ldots, \tau_n) \) of transitions \( E(m) = \{t_1, \ldots, t_n\} \) enabled by \( m; D \subseteq \mathbb{N}^\infty \) is the support of \( f_{\tau_{\text{age}}, \tau} \).

The initial (unconditioned) transient stochastic state class \( \Sigma_0 \) assigns non-zero probability to states with marking \( m_0 \) and absolute time \( \tau_{\text{age}} = 0 \) according to the times to fire PDF \( f_\gamma \); i.e., \( \Sigma_0 = \langle m_0, D_0, f_{\tau_{\text{age}}, \tau} \rangle \), where \( D_0 = [0, \infty) \times \text{[support of } f_\gamma \text{ and } f_{\tau_{\text{age}}, \tau} \text{]} \). Given a class \( \Sigma \), the state PDF conditioned on the execution of a transition \( \gamma \) at an absolute time in the interval \( I \) is given by the successor class of \( \Sigma \) through \( \gamma \) and \( I \).

Definition 4. We say that \( \Sigma' = (m', D', f_{\tau_{\text{age}}, \tau}) \) is the successor of \( \Sigma = (m, D, f_{\tau_{\text{age}}, \tau}) \) at some time in \( I \) and with succession probability \( \mu \) and we write \( \Sigma \xrightarrow{\gamma, I, \mu} \Sigma' \), if, given that the marking of the STPN is \( m \) in \( \tau_{\text{age}} \) and, a random vector distributed over \( D \) according to \( f_{\tau_{\text{age}}, \tau} \) then: (1) \( \gamma \) has non-zero probability \( \mu \) of firing in \( \Sigma \) at some time in \( I \), and (2) if \( \gamma \) fires in \( \Sigma \) at some time in \( I \), its firing yields the marking \( m' \) and, conditioned on this event, the times to fire and \( \tau_{\text{age}} \) after the firing are distributed over \( D' \) according to \( f_{\tau_{\text{age}}, \tau} \).

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 \( \tau_{\text{age}}, \tau \) in successor classes. Given a class \( \Sigma \), an enabled transition \( \gamma \) and a firing interval \( I \), this calculus computes the succession probability \( \mu \) and the (unique) successor class \( \Sigma' \) 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 future evolution. The treatment in this paper does not assume knowledge of the calculus;
for convenience, we provide a summary in the additional material, and refer to [30], [31], [37] for a detailed description.

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

**Proposition 2.** Given an initial marking $m_0$ and PDF $f_{\vec{r}_0}$ 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 $Pr_{m_0,f_{\vec{r}_0}}$ is equal to the product $\prod_{i=1}^n \mu_i$ of succession probabilities for the sequence of transient stochastic state classes

$$
\Sigma_0 \gamma_1 I_1 \mu_1 \to \Sigma_1 \gamma_2 I_2 \mu_2 \to \cdots \to \gamma_n I_n \mu_n \to \Sigma_n,
$$
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 $E_0 = \Omega_m$ and, for each $i > 0$,

$$
E_i = \{ \omega \in \Omega_m | | \omega | \geq i, \omega[i] = \gamma_i \text{ and } T(i, \omega) \in I_i \}
$$
is the constrained set of paths imposing an absolute time only on the $i$th transition. By induction on the definition of succession of stochastic state classes, for all $i \leq n$, the class $\Sigma_i$ in

$$
\Sigma_0 \gamma_1 I_1 \mu_1 \to \Sigma_1 \gamma_2 I_2 \mu_2 \to \cdots \to \gamma_n I_n \mu_n \to \Sigma_n
$$
represents the joint PDF of the absolute time and current state given the events $E_0, E_1, E_2, \ldots, E_i$, and $\mu_i$ is the probability $Pr_{m_0,f_{\vec{r}_0}}(E_i | E_0, E_1, E_2, \ldots, E_{i-1})$ that paths that performed $\gamma_1, \ldots, \gamma_{i-1}$ in $I_1, \ldots, I_{i-1}$ will also perform $\gamma_i$ at some time in $I_i$. Then $Pr_{m_0,f_{\vec{r}_0}}(E_0 \cap E_1 \cap \cdots \cap E_n) = Pr_{m_0,f_{\vec{r}_0}}(E_0) \cdot Pr_{m_0,f_{\vec{r}_0}}(E_1 | E_0, E_1, \ldots, E_{i-1}) = \prod_{i=1}^n \mu_i$ if $\mu_i > 0$ for all $i \leq n$; if some event $E_i$ has zero probability given $E_0, E_1, \ldots, E_{i-1}$ (i.e., $\Sigma_{i-1}$ has no successor through $\gamma_i$ at some time in $I_i$ and thus $\mu_i = 0$), the measure of the cylinder set is zero. \hfill \Box

### 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 $\varphi_2$ at some time in the interval $[\alpha, \beta]$ without violating a safety predicate $\varphi_1$ in previous states. The syntax of the logic is

$$
\psi ::= \text{TRUE} \mid \neg \varphi \mid \psi \land \psi \mid P_{\mu_0}[\varphi \Upsilon[\alpha, \beta] \varphi_2]
$$

$$
\varphi ::= \text{TRUE} \mid \neg \varphi \mid \varphi \land \psi \mid \text{AP}
$$

where $\alpha, \beta \in \mathbb{R}_{\geq 0}$ is a probability value, $\varphi, \psi \in \mathbb{P}_{\geq 0}$, and atomic predicates on markings are defined as $\text{AP} ::= g \circ x \in \mathbb{Q}_{>0}$, $(\leq, =, \neq, \geq, >), x \in \mathbb{R}$ for $g : \mathbb{N}^p \to \mathbb{R}$ is a real-valued function (e.g., “free > 1” is an atomic predicate 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 $s_0 = (m_0, \vec{r}_0)$ in which $m_0$ is a given marking and $\vec{r}_0$ is a vector of times to fire of enabled transitions sampled according to a given PDF $f_{\vec{r}_0}$. Without loss of generality, we assume that all enabled transitions $E(m_0) = \{f_1, f_2, \ldots, f_n\}$ are newly enabled in the initial marking, and thus $\vec{r}_0$ is distributed according to $f_{\vec{r}_0}(x_1, x_2, \ldots, x_n) = \prod_{i=1}^n f_{\vec{r}_0}(x_i)$.

Note that $\varphi$-formulae (i.e., the arguments of an until operator) only depend on the marking $m$ of visited states, while $\psi$-formulae (i.e., Boolean combinations of until operators) depend on the initial marking $m_0$ and also on the PDF $f_{\vec{r}_0}$.

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

$$
m \models \text{TRUE} \text{ and } (m_0, f_{\vec{r}_0}) \models \text{TRUE} \text{ are always satisfied}
$$

$$
m \models \neg \varphi \iff m \not\models \varphi \text{ is satisfied by } m
$$

$$
m \models \varphi_1 \land \varphi_2 \iff m \models \varphi_1 \land m \models \varphi_2 \equiv (m_0, f_{\vec{r}_0}) \models \neg \psi \iff (m_0, f_{\vec{r}_0}) \models \psi \iff \psi \equiv (m_0, f_{\vec{r}_0}) \models \psi_1 \land (m_0, f_{\vec{r}_0}) \models \psi_2
$$

According to Eq. (1), a path satisfying the interval until operator $\varphi_1 \Upsilon[\alpha, \beta] \varphi_2$ must visit only $\varphi_1$-states and then either reach a $\varphi_2$-state in $[\alpha, \beta]$, or reach a $(\varphi_1 \land \varphi_2)$-state before time $\alpha$ and let time advance past $\alpha$. 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 non-zero probability when the model includes IMM or DET transitions) satisfy $\varphi_1$, 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 $P_{< 0.4}(\text{buffer < 2}) \Upsilon[0, 7](\text{failed = 1})$ 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 $\varphi_1, \varphi_2$, and $\alpha, \beta \in \mathbb{Q}_{>0}$, the set of paths satisfying the interval until operator is an event of $f_{m_0}$. Concretely, this means that the value of $Pr_{m_0,f_{\vec{r}_0}}(\omega \in \Omega_m | \omega \models \varphi_1 \Upsilon[\alpha, \beta] \varphi_2)$ is well-defined and the semantics of Definition 5 can be computed with stochastic state classes.

**Proposition 3.** For each $\varphi_1, \varphi_2 \in \mathbb{P}^p$ and $\alpha, \beta \in \mathbb{Q}_{>0}$, the set

$$
\{ \omega \in \Omega_m | \omega \models \varphi_1 \Upsilon[\alpha, \beta] \varphi_2 \}
$$

of paths satisfying the interval until operator is a countable union of cylinder sets.

**Proof.** The cylinder sets that end on a $\varphi_2$-marking reached only through $\varphi_1$-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 $\varphi_2$-marking only through $\varphi_1$-markings, we consider (1) the cylinder set imposing only an absolute
time in \( I_n = [\alpha, \beta] \) for the \( n \)th transition firing, and
(2) if the marking reached after \( \gamma_n \) satisfies also \( \varphi_1 \),
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 \( \gamma_n \) and a bound \( I_{n+1} = [\alpha, \infty) \) for the firing of \( \gamma_{n+1} \).

The countable union of these cylinder sets is an event of \( \mathcal{F}_{m_0} \)
including all and only the successful paths. \( \square \)

### 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 some 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 \( \Pr_{m_0,f_{\gamma_n}} \{ \omega \in \Omega_{m_0} \mid \omega = \varphi_1 U^{[\alpha,\beta]} \varphi_2 \} \).
In [28], 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 \( \varphi_1 U^{[\alpha,\beta]} \varphi_2 \). Similarly
to [26], 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 \( \beta \). 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 discrete events
before the first regeneration, 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],
[39] 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 \( t \):

\[
\begin{align*}
P\{ M(t + t') = j \mid M(t) = i, (\forall u \in A_t).M(u) = i_u \} = \\
P\{ M(t') = j \mid M(0) = i \}
\end{align*}
\]

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 times
to fire satisfy this condition is called regenerative, and the PDF
of the vector \( \langle \tau_{\text{age}}, \bar{\tau} \rangle \) is always in product form.

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

**Remark 1.** In a regenerative state class \( \Sigma = \langle m, D, f_{\langle \tau_{\text{age}}, \bar{\tau} \rangle} \rangle \) 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 \( d = (d_1, \ldots, d_n) \in \mathbb{R}_{\geq 0}^n \) is the vector of enabling times
of GEN transitions, the support \( D \) and the probability density
function \( f_{\langle \tau_{\text{age}}, \bar{\tau} \rangle} \) of \( \langle \tau_{\text{age}}, \bar{\tau} \rangle \) in \( \Sigma \) are equal to

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

and

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

respectively, for some PDF \( f_{\text{age}} \) of \( \tau_{\text{age}} \) with support \( D_{\text{age}} \).

**Remark 2.** As a notable case, a class in which all GEN
transitions are newly enabled is regenerative with \( d = 0 \);
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 \( \Sigma_0 \).

**Remark 3.** A regenerative class is uniquely identified by its
marking \( m \), by the vector \( d \) of enabling times, and by the PDF \( f_{\text{age}} \) and support \( D_{\text{age}} \) of the absolute time \( \tau_{\text{age}} \) 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 \( d \) identify the support and distribution
of GEN timers.

**Remark 4.** In the following, given a pair \( i = (m, \bar{d}) \), where \( m \)
is a marking and \( \bar{d} \) is a vector of enabling times, we denote by
\( (\Omega_i, \mathcal{F}_i, P_{\tau_{\text{age}}}) \) the probability space for the STPN paths
with initial marking \( m \) when times to fire are initially distributed
according to the product-form PDF \( f_{\bar{d}} \) described in Remark 1
for the given enabling times \( \bar{d} \) and for \( \tau_{\text{age}} \) equal to zero (i.e.,
\( f_{\text{age}}(x_{\text{age}}) = \delta(x_{\text{age}}) \) and \( D_{\text{age}} = [0, 0] \)).
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 if and only if \( t \) 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 \( i \)th 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 5–13): 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 \( \gamma_i \). 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.

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 \( \Sigma \xrightarrow{\gamma, \mu, \tau} \Sigma' \) 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 \( \Sigma_0 \) 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 \( \Sigma \) associates each node \( n \in N \) with a stochastic state class \( \Sigma(n) \), with \( \Sigma(n_0) = \Sigma_0 \); 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 \( \Sigma(n) \xrightarrow{\gamma, \mu} \Sigma(n') \).

The transient tree enumerates all the sequences of transition firings with non-zero probability, and associates each node with the resulting PDF over states and absolute reaching times. A node \( n \) associated with a regenerative class \( \Sigma(n) = (m, D, f_{\gamma_{n_0}, \tau}) \) 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 \( \tau \) 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 the sequences 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 in distinct regenerative nodes, but always produces the same
subsequent behaviors. In the sequel, we refer to the pair $(m, \overline{d})$ as regeneration condition.

The next lemma completes the picture by focusing on the time advancement 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 a regeneration are independent.

**Lemma 2.** Let $n_j$ be a regenerative node, $n_k$ be the descendant of $n_j$ reached through the transitions $\rho = \gamma_1, \ldots, \gamma_n$, and $f^j_{\text{age}}$ and $f^k_{\text{age}}$ be the marginal PDF of the $\tau_{\text{age}}$ variables in $\Sigma(n_j)$ and $\Sigma(n_k)$, respectively. Then,

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

(2)

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

**Proof.** The time spent in the execution of $\rho$ from $n_j$ or from $n_j$ is the same as it only depends on the marginal distribution of times to fire in $n_j$ and $\overline{n_j}$ (Lemma 1); moreover, the distribution of this time is given by $f^k_{\text{age}}$, since the age is equal to zero in $\overline{n_j}$, i.e., $f^j_{\text{age}}(x_{\text{age}}) = \delta(x_{\text{age}})$. Since $n_j$ is regenerative, $f^j_{\text{age}}$ is in product form with respect to the marginal PDF of times to fire (Remark 1), and thus the evolution from $n_j$ to $\overline{n_j}$ 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 with the duration of $\rho$, and it is distributed as the convolution $\int_{-\infty}^{+\infty} f^j_{\text{age}}(u) f^k_{\text{age}}(x_{\text{age}} - u) \, du$.

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 $\overline{\tau}$. In Lemma 2, the assumption of regeneration is used also to guarantee that the PDFs of $\tau_{\text{age}}$ and $\overline{\tau}$ 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 $2\text{free operational}$ and sequence of discrete events $\text{fail, restart, arrival, service}$ (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 $\Sigma_0, \Sigma_3, \Sigma_4$, 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 $\text{arrival}$ in $\Sigma_4$ is not newly enabled, but its PDF is uniquely identified by the deterministic enabling time $1.5$. Note that, as discussed in [37], $\tau_{\text{age}}$ 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 [5]. 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 $\varphi_{\text{ext}} = P_{>1/3}(\varphi_{\text{in}} \cup \{p = 0\})$ with $\varphi_{\text{in}} = P_{<1/3}(p = 1)$. Intuitively, for $\varphi_{\text{ext}}$ to hold, $\varphi_{\text{in}}$ 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 $\varphi_{\text{in}}$ changes over time: given a sojourn of $x$ time units, the probability of $(p = 1) \cup \{p = 0\}$ is $f_x + 1 / (4 - x) dy$, which is monotonically nondecreasing and remains under $1/3$ only when $x \in [0, 1)$. Therefore, the only paths that satisfy $\varphi_{\text{in}}$ continuously are those that sojourn in $p = 1$ less than $x = 1$ time units, and the probability of satisfying the nested operator $\varphi_{\text{ext}} \cup \{p = 0\}$ is $f_x 1 / 4 dy = 1 / 4$. By this argument, we conclude that the external formula $\varphi_{\text{ext}}$ 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 interval until operator

Given the predicates $\varphi_1$ and $\varphi_2$, a real interval $[\alpha, \beta]$ with $\alpha, \beta \in Q_{\geq 0}$, and a regeneration condition $i = (m, \overline{d})$, we define

$$\Omega_i(\alpha, \beta) := \{ \omega \in \Omega_i \mid \omega = \varphi_1 U^{(\alpha, \beta)} \varphi_2 \}$$

as the set of paths that start from the marking of $i$ and satisfy the interval until operator, and we denote by $p_{\omega}(\alpha, \beta) := Pr_i(\Omega_i(\alpha, \beta))$ its probability measure, as defined
characterizing the behavior of the stochastic process \( \Omega \) reaching a regeneration from those that encounter the first regeneration condition. The probability \( \text{OK}(\omega) \) the index of the first state leading to the satisfaction of the until operator:

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

Moreover, we indicate with \( t_{\text{REG}}(\omega) := T(\text{REG}(\omega), \omega) \) the time of the first regeneration in \( \omega \), and with \( (m_{\text{REG}(\omega)}, \bar{d}_{\text{REG}(\omega)}) \) 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 \( \alpha \) or between \( \alpha \) and \( \beta \), and then satisfy the until operator:

\[
\Omega_i^L(\alpha, \beta) := \{ \omega \in \Omega_i(\alpha, \beta) \mid \text{OK}(\omega) < \text{REG}(\omega) \},
\]

\[
\Omega_i^G(\alpha, \beta) := \{ \omega \in \Omega_i(\alpha, \beta) \mid \text{OK}(\omega) \geq \text{REG}(\omega) \wedge t_{\text{REG}}(\omega) < \alpha \},
\]

\[
\Omega_i^H(\alpha, \beta) := \{ \omega \in \Omega_i(\alpha, \beta) \mid \text{OK}(\omega) \geq \text{REG}(\omega) \wedge t_{\text{REG}}(\omega) \in [\alpha, \beta) \}.
\]

Proposition 4. For any regeneration condition \( i \) and \( \alpha, \beta \in \mathbb{Q}_{\geq 0} \), 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 \( \Omega_i^L(\alpha, \beta), \Omega_i^G(\alpha, \beta), \Omega_i^H(\alpha, \beta) \) can be expressed in terms of three kernels characterizing the behavior of the stochastic process within the first regeneration epoch.

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 the first regeneration epoch and require the next Propositions 5 and 6.

Proposition 5. The measure \( Pr_i(\Omega_i^H(\alpha, \beta)) \) is equal to

\[
\sum_k \int_{x \in [0, \alpha)} dG_{ik}^{x}(x) \, p_k(\alpha - x, \beta - x)
\]

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

\[
G_{ik}^{x}(x) := Pr_i(\omega \in \Omega_i \mid t_{\text{REG}}(\omega) \leq x \wedge (m_{\text{REG}(\omega)}, \bar{d}_{\text{REG}(\omega)}) = k \wedge (\forall j < \text{REG}(\omega), \{m_j = \varphi_1\} \wedge \{m_j = \varphi_2\} \Rightarrow T(j + 1, \omega) < \alpha) \).
\]

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

\[
\int_{X(\alpha)} p_{(m_{\text{REG}(\omega)}, \bar{d}_{\text{REG}(\omega)})}(\alpha - t_{\text{REG}}(\omega), \beta - t_{\text{REG}}(\omega)) \, dPr_i(\omega)
\]

where \( X(\alpha) := \{ \omega \in \Omega_i \mid t_{\text{REG}}(\omega) < \alpha \wedge (\forall j < \text{REG}(\omega), \{m_j = \varphi_1\}) \}. \) In Eq. (5), the measure of each path reaching its first regeneration point before \( \alpha \) without violating \( \varphi_1 \) 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 \( (m_{\text{REG}(\omega)}, \bar{d}_{\text{REG}(\omega)}) = k \) and times \( t_{\text{REG}}(\omega) = x \) of the first regeneration before \( \alpha \), we obtain Eq. (3), where the global kernel \( G_{ik}^{x}(x) \) represents the probability of reaching, within time \( x \), a regeneration with regeneration condition \( k \) while always satisfying \( \varphi_1 \) in previous states.

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

\[
\sum_k \int_{x \in [\alpha, \beta]} dG_{ik}^{x}(x) \, p_k(0, \beta - x)
\]

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

\[
H_i^{\varphi_1, \varphi_2}(\alpha, x) := Pr_i(\omega \in \Omega_i \mid t_{\text{REG}}(\omega) \leq x \wedge (m_{\text{REG}(\omega)}, \bar{d}_{\text{REG}(\omega)}) = k \wedge (\forall j < \text{REG}(\omega), \{m_j = \varphi_1 \wedge \{m_j = \varphi_2\} \Rightarrow T(j + 1, \omega) < \alpha) \).
\]

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., \( \omega \) encounters a goal state after reaching a regeneration point at some time in \( [\alpha, \beta] \)). The proof is analogous to that of Proposition 5: in this case, states \( (m_j, \bar{t}_j) \) with \( j < \text{REG}(\omega) \) must satisfy \( \varphi_1 \), but not \( \varphi_2 \) unless the sojourn ends before \( \alpha \); otherwise, the interval until operator would be satisfied by letting time advance past \( \alpha \) in a \( \varphi_2 \)-state, and we would have \( \text{OK}(\omega) < \text{REG}(\omega) \). The last line of Eq. (7) imposes this additional constraint, and the conditional global kernel \( H_i^{\varphi_1, \varphi_2}(\alpha, x) \) represents the probability of reaching, within time \( x \), a regeneration with regeneration condition \( k \) while satisfying \( \varphi_1 \) in previous states and also \( \neg \varphi_2 \) after time \( \alpha \).

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 \( p_{(m_0, \bar{d})}(\alpha, \beta) \) of paths satisfying the until operator from the initial regeneration condition \( (m_0, \bar{0}) \) can be computed from the measures \( p_i(\alpha, \beta) \) for all possible \( i, \alpha, \beta \).
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 \( \varphi_1 \ U^{[\alpha, \beta]} \varphi_2 \) from the initial marking \( m \) with PDF of GEN timers given by the deterministic enabling times \( \vec{d} \), are given by the system of integral equations
\[
p_i(\alpha, \beta) = L_{i}^{\varphi_1, \varphi_2}(\alpha, \beta) + \sum_k \int_{x \in [0, \alpha]} dG_{ik}^{\varphi_1}(x) p_k(\alpha - x, \beta - x) + \sum_k \int_{x \in [\alpha, \beta]} dH_{ik}^{\varphi_1, \varphi_2}(\alpha, x) p_k(0, \beta - x) \tag{8}
\]
where \( i \) and \( k \) range over all reachable regeneration conditions.

The theorem represents a bivariate generalization of Markov renewal equations [18, 40] leveraging three kernels that result from a renewal argument specific to the interval until operator: the model can satisfy \( \varphi_2 \) between \( \alpha \) and \( \beta \) either (1) without regenerations, (2) reaching the first regeneration before \( \alpha \), or (3) reaching the first regeneration in \( [\alpha, \beta] \). As illustrated in Fig. 4, \( \varphi_1 \) must always be satisfied; additionally, also \( \neg \varphi_2 \) must be satisfied between \( \alpha \) 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 \( \varphi_2 \); 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 \( \beta \), 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 after discrete events limited to the first regeneration epoch; Eq. (8) can then be solved numerically in the time domain through techniques such as Newton–Cotes formulas or Runge–Kutta methods [41], with various trade-offs between accuracy and complexity. Given a step \( h \), discretizing the temporal domain \([0, \beta]\) into points \( t_n = nh \), with \( \alpha = \bar{a}h \) and \( \beta = \bar{b}h \), Newton–Cotes formulas define the linear system
\[
\bar{p}(t_a, t_b) = \bar{L}_{i}^{\varphi_1, \varphi_2}(t_a, t_b) + \sum_{m=0}^{a} w_m dG_{i}(t_m) \bar{p}(t_{a-m}, t_{b-m}) \tag{9}
\]
\[
+ \sum_{m=a}^{b} w_m dH_{i}(t_a, t_m) \bar{p}(0, t_{b-m})
\]
in the unknowns \( \bar{p}(0, t_b) \) for \( b = 0, \ldots, \bar{b} - \bar{a} \), and \( \bar{p}(t_a, t_{a+b-\bar{a}}) \) 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, \( dG_{i}(0) = 0 \) and \( dH_{i}(0,0) = 0 \), and Eq. (9) can be solved by forward substitution:
\[
\bar{p}(0, t_b) = \bar{L}_{i}^{\varphi_1, \varphi_2}(0, t_b) + \sum_{m=1}^{b} w_m dH_{i}(0, t_m) \bar{p}(0, t_{b-m})
\]
for \( b = 0, \ldots, \bar{b} - \bar{a} \), and
\[
\bar{p}(t_a, t_b) = \bar{L}_{i}^{\varphi_1, \varphi_2}(t_a, t_b) + \sum_{m=1}^{a} w_m dG_{i}(t_m) \bar{p}(t_{a-m}, t_{b-m}) + \sum_{m=a}^{b} w_m dH_{i}(t_a, t_m) \bar{p}(0, t_{b-m})
\]
for \( a = 1, \ldots, \bar{a} \) and \( b = a + \bar{b} - \bar{a} \). By evaluating the unknowns \( \bar{p}(t_a, t_b) \) in this order, the solution \( \bar{p}(t_a, t_b) \) can be computed as a direct sum that requires:
- local kernel values \( \bar{L}_{i}^{\varphi_1, \varphi_2}(0, t_b) \) for \( b = 0, \ldots, \bar{b} - \bar{a} \) and \( \bar{L}_{i}^{\varphi_1, \varphi_2}(t_a, t_{a+b-\bar{a}}) \) for \( a = 1, \ldots, \bar{a} \);
- global kernel values \( dG_{i}(t_m) \) for \( m = 1, \ldots, \bar{a} \);
- conditional global kernel values \( dH_{i}(t_a, t_m) \) for \( a = 0, \ldots, \bar{a} \) and \( m = a, \ldots, a + \bar{b} - \bar{a} \).

Values of \( L_{i}^{\varphi_1, \varphi_2}(t_a, t_b) \), \( G_{ik}^{\varphi_1}(t_m) \), and \( H_{ik}^{\varphi_1, \varphi_2}(t_a, t_m) \) can be derived from the transient tree enumerated from regeneration condition \( i \), halting on (1) regenerative nodes, (2) nodes not satisfying \( \varphi_1 \), (3) nodes with minimum reaching time greater than \( \beta \). In the enumeration, the successors of a class \( \Sigma \), indicated as \( \text{Successors}(\Sigma) \), are derived according to the calculus described in the supplemental material and in [30], [31], [37]. A class \( \Sigma_\alpha \) derived through successions \( \Sigma_{i-1} \rightarrow_{\gamma_{\alpha}^{i-1}} \Sigma_i \) for \( i = 1, \ldots, n \) is associated with the
**Evaluate:** $L_{i}^{\varphi_1 \lor \varphi_2}(\alpha, \beta)$

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 $\Sigma = (m, D, f)$ from $\Gamma$
   6. if $m \models \neg \varphi_1 \land \neg \varphi_2$ or $\Sigma$ is regenerative
      7. discard $\Sigma$
   8. **else if** $m \models \neg \varphi_1 \land \varphi_2$
      9. $p \leftarrow p + \eta(\Sigma_{in}, \alpha, \beta)$
   10. **else if** $m \models \varphi_1 \land \neg \varphi_2$
      11. $\Gamma \leftarrow \Gamma \cup \text{SUCCESSORS}(\Sigma_{out}, \alpha, \beta)$
   12. **else if** $m \models \varphi_1 \land \varphi_2$
      13. $p \leftarrow p + \eta(\Sigma_{in}, \alpha, \beta)$
      14. $\Gamma \leftarrow \Gamma \cup \text{SUCCESSORS}(\Sigma_{out}, \alpha, \beta)$
15. $\Gamma \leftarrow \Gamma \cup \text{SUCCESSORS}(\Sigma_{out}, \alpha, \beta)$
16. return $p$

**Evaluate:** $H_{i}^{\varphi_1 \lor \varphi_2}(\alpha, x)$

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 $\Sigma = (m, D, f)$ from $\Gamma$
   6. if $\Sigma$ is regenerative with regeneration condition $k$
      7. $p_k \leftarrow p_k + \eta(\Sigma_{in}, \alpha, \beta)$
   8. **else if** $m \models \varphi_1 \land \neg \varphi_2$
      9. $\Gamma \leftarrow \Gamma \cup \text{SUCCESSORS}(\Sigma_{out}, \alpha, \beta)$
   10. **else if** $m \models \varphi_1 \land \varphi_2$
      11. $\Gamma \leftarrow \Gamma \cup \text{SUCCESSORS}(\Sigma_{out}, \alpha, \beta)$
12. return $p_k$

**Fig. 6:** Algorithm evaluating $L_{i}^{\varphi_1 \lor \varphi_2}(\alpha, \beta)$.

**Global kernel values** $dG_{ik}^{\varphi_1}(t_m)$. The values $dG_{ik}^{\varphi_1}(t_m)$ for $m = 1, \ldots, n$ 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 $\alpha$. In particular, the value $dG_{ik}^{\varphi_1}(t_m)$ can be obtained by summing up, for each regenerative leaf $n$ with regeneration $k$, the PDF value of the absolute reaching time multiplied by $\eta(n)$, i.e., $dG_{ik}^{\varphi_1}(t_m) = \sum \eta(n) f_{\text{age}}(-t_m)$.

**Conditional global kernel values** $dH_{ik}^{\varphi_1 \lor \varphi_2}(t_a, t_m)$. The values $dH_{ik}^{\varphi_1 \lor \varphi_2}(t_a, t_m)$ can be approximated as $(H_{ik}^{\varphi_1 \lor \varphi_2}(t_a, t_m) - H_{ik}^{\varphi_1}(t_a, t_{m-1}) - H_{ik}^{\varphi_2}(t_a, t_{m-1}) - \bar{H}_{ik}^{\varphi_1 \lor \varphi_2}(t_a, t_{m-1})) / h$, where the values $H_{ik}^{\varphi_1 \lor \varphi_2}(\alpha, x)$ are derived from the transient tree enumerated from regeneration condition $i$, stopping on regenerations, or $\neg \varphi_1$-classes, and on classes reached after $x$. The evaluation discards also states in $\varphi_1 \land \varphi_2$-classes that are left after $\alpha$, since $H_{ik}^{\varphi_1 \lor \varphi_2}(\alpha, x)$ gives the measure of the set of paths that end on regeneration condition $k$ within time $x$, after visiting only $\varphi_1$-states, and without visiting any $\varphi_2$-state after $\alpha$. The algorithm in Fig. 7 evaluates $H_{ik}^{\varphi_1 \lor \varphi_2}(\alpha, x)$ for all $k$ by enumerating the transient tree from regeneration condition $i$; similarly to Fig. 6, $\Gamma$ is the frontier set and $p_k$ accumulates the value of $H_{ik}^{\varphi_1 \lor \varphi_2}(\alpha, x)$.

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

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

**Overall,** for each regeneration condition $i$, the transient tree enumeration is repeated:

- $\frac{\beta}{n} + 1$ times to compute $L_{i}^{\varphi_1 \lor \varphi_2}(t_a, t_b)$ for $a = 0$ and $b = 0, \ldots, b - \bar{a}$, and for $a = 1, \ldots, \bar{a}$ and $b = a + \bar{a}$.
• once for the evaluation of the global kernel values $dG_k^{t_i}(t_m)$ for $m = 1, \ldots, a$ and all $k$;
• $(\frac{1}{2} + 1)(\frac{\beta - \alpha}{T} + 2)$ times for the evaluation of the conditional global kernel values $H_{ik}^{t_1, t_2}(t_a, t_m)$ for $a = 0, 1, \ldots, a, m = a - 1, \ldots, a + b - a$, and all $k$.

If $R$ is the number of reachable regeneration conditions, the number of transient tree enumerations is thus $O(\frac{\beta - \alpha}{T}R)$. The advantage with respect to [26], [28], where model checking is performed with a single transient tree enumeration, lies in the reduced depth of these transient trees: the computation of a successor class grows exponentially with the depth of the predecessor in the tree [30]; when the time bound $\beta$ 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 $\alpha$

The availability of deterministic transitions in STPNs can be leveraged to remove the lower bound $\alpha$ for the satisfaction of $\varphi_2$, 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 $\alpha$.

The complexity of Eq. (8) is largely reduced if the until operator does not impose a minimum time for the acceptance of the goal condition $\varphi_2$, i.e., if $\alpha = 0$. In this case, both Eq. (8) and its kernels are simplified: the local kernel $L_{i1}^{t_1, t_2}(0, \beta)$ becomes the probability that, starting from the regenerative condition $i$, a $\varphi_2$-state is encountered before the first regeneration and within time $\beta$, traversing only $\varphi_1$-states. Moreover, the second term of Eq. (8) gives zero contribution, while the conditional global kernel $H_{i1}^{t_1, t_2}(0, x)$ becomes the probability that regeneration condition $k$ is reached within $x$ after visiting only states that satisfy $\varphi_1 \land \neg \varphi_2$.

The kernels $L_{i1}^{t_1, t_2}(0, \beta)$ and $H_{i1}^{t_1, t_2}(0, x)$ can be derived from transient trees enumerated from each regeneration condition $i$ and halting on regenerative nodes, nodes not satisfying $\varphi_1 \land \neg \varphi_2$, or nodes with minimum reaching time greater than $\beta$. The local kernel $L_{i1}^{t_1, t_2}(0, \beta)$ can be computed from the $\tau_{age}$ PDF of leaf nodes satisfying $\varphi_2$, while $H_{i1}^{t_1, t_2}(0, x)$ can be computed from the $\tau_{age}$ PDF of regenerative leaf nodes. 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 $\varphi_2$ 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 PDF $f_t(x) = \delta(x - \alpha)$ and $\varphi_2$ can be restricted to $\varphi'_2 = \varphi_2 \land \{t \text{ has fired}\}$. In so doing, regenerations before $\alpha$ of the original model are not exploited, since the additional DET transition $t$ is not “memoryless.” Only after the firing of $t$ at time $\alpha$, regenerations will be fully exploited in the analysis. This approach is thus well-suited only for cases with a small $\alpha$ 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 $\varphi_1 \land \neg \varphi_2$ from the initial marking $\{\text{free operational with } \varphi_1 = (\text{buffer} < 2) \text{ and } \varphi_2 = (\text{failed} = 1)\}$ is $0.3313 < 0.4$ for $\alpha = 0$ and $\beta = 7$, and $0.2339 > 0.2$ for $\alpha = 2.5$ and $\beta = 7$.

In the latter case, when limited to $\varphi_1$-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 [42]. 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 $i \leftarrow i$, 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 ← 0 on exit; whereas, if id ≠ 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 [43] and Uppaal [44], randomized versions have been analyzed in closed-form [45] or through simulation [46], [47] 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 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, (EXP with rate 0.1), and enters the contention by reaching...
place writing, as soon as id = 0 (IMM transition readEmpty with enabling function ?id = 0); it then sets the shared variable to its own identifier (as specified by the update function id ← i) at the end of a write operation (transition writei, with duration uniformly distributed over [0, 1],) and sojourns in a waiting state (place waiting) for a time greater than the maximum time that any process can spend writing to id (transition waiti, DET equal to 1.1). When the wait completes, process Pi reads id again to ensure that its write was the last one (place reading); if id ≠ i, the control goes back to the initial state of contention ready, (IMM transition readOther), whereas, if the shared variable is still equal to the process identifier (i.e., id = i), Pi enters the critical section cs, (IMM transition readSelf), performs its service (transition servicei, uniform over [0, 2]), and then resets the shared variable (IMM transition reset), returning idle.

5.2 Quantitative evaluation

We consider a deadline requirement prescribing that the latency for the access of Pi 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 βr > β (which we call relaxed deadline) with probability greater than pr > p. This property can be encoded as the Boolean conjunction of two probabilistic existence properties [1]:

\[ P_{>p}[\text{true } \mathcal{U}^{[0,\beta]}(cs_i = 1)] \land P_{>p_r}[\text{true } \mathcal{U}^{[0,\beta_r]}(cs_i = 1)]. \tag{11} \]

Fig. 9 reports the measure \(Pr_{(m,0)}\{\omega \in \Omega_m \mid \omega \models \mathcal{U}^{[0,\beta]}(cs_i = 1)\}\) as a function of β, for m ∈ \(\{m_A, m_B, m_C\}\) where: \(m_A = \text{ready}_1 \text{idle}_2 \text{idle}_3\) (which occurs when \(P_1\) becomes ready while the other processes are idle), \(m_B = 3\text{id ready}_1 \text{idle}_2 \text{waiting}_3\) (which occurs when \(P_1\) becomes ready and \(P_3\) has just set the shared variable, closing the access to the contention), \(m_C = 3\text{id ready}_1 \text{writing}_2 \text{waiting}_3\) (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 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 \(P_{>p}[\text{true } \mathcal{U}^{[0,\beta]}(cs_i = 1)]\) = FALSE and \(P_{>p_r}[\text{true } \mathcal{U}^{[0,\beta_r]}(cs_i = 1)]\) = TRUE. The relaxed deadline is thus met with the required probability, but the base 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
The probability of paths satisfying \((cs_1 = 0)\ U^{[0,\beta]}(\sum^n_{i=2} completed_i = 1 \land \sum^n_{i=2} ready_i \geq k)\) as a function of \(\beta\), from marking \(m_A = ready_1 idle_2 idle_3\), and different service times.

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 the contention) and the number \(n - 1\) and arrival rate \(\lambda\) of other processes entering 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 \(P_1\):

\[
P_{>p}[\{cs_1 = 0\} U^{[0,\beta]}(cs_1 = 1)] \tag{12}
\]

Fig. 10 reports the probability measure associated with this property (as function of \(\beta\), from the initial marking \(m_A\)) 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 subtle: 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 one 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 \(f(x) = x e^{-20x}/400\) 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 \(m_0 = idle_1 idle_2 idle_3\), 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 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

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

and verified for given values of \(\alpha, \beta\) and \(p\) so as to determine the time \(\alpha\) 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 \((cs_3 = 0)\ U^{[\alpha,\beta]}(cs_1 = 1)\) for different values of \(\alpha\) 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 \(P_{(m_0,\bar{0})}(\alpha, \beta) := P_{(m_0,\bar{0})}[\omega \in \Omega_{m_0} | \omega = (cs_3 = 0) U^{[\alpha,\beta]}(cs_1 = 1)]\) is plotted in Fig. 12 for \(\alpha = 0, 0.1, \ldots, 4\), which might serve to select an optimum time \(\alpha\) to start the observation.
would necessarily incur in this exponential complexity by the product of a single solution of Eq. (8) for the evaluation of \( p_{(m_0, β)}(4, 4 + δ) \) with step 0.1. For this measure, we identify 17 distinct regeneration conditions reachable under condition \( c_{S_3} = 0 \), whose transient trees include 904 stochastic state classes. For \( α = 4 \) and \( δ = 1 \), adopting a step size \( h = 0.1 \), the enumeration of each tree is repeated \( \frac{5}{2} + 1 = 51 \) times for the evaluation of \( L_{ik}^{α+1; β+1} \), once for the evaluation of \( dL_{ik}^{α} \((t_m) \), and \( \frac{3}{1} + 1)(\frac{1}{1} + 2) = 492 \) times for the evaluation of \( dH_{ik}^{α} \), resulting in 491776 enumerated classes.

As a comparison, the reduction to transient analysis through a deterministic timer \( α = 4 \) 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 [26], [28] would necessarily incur in this exponential complexity by enumerating the times to fire PDFs after each discrete event along any feasible path until \( α + δ \); in the approach of [26], this complexity would be further exacerbated by the finer grain of regions [27] with respect to zones [32], [33].

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 number of enumerated classes shown by the dotted line. In contrast, in the regenerative approach based on the bivariate Markov renewal equations of Eq. (8), the total number of enumerated classes follows the solid line.

5.3 Computational advantage of regenerative analysis

In the construction of Fig. 12, for each value \( δ \) of \( β - α \), the measures \( p_{(m_0, β)}(α, α+δ) \) for \( α = 0, 0.1, \ldots, 4 \) are computed as the by-product of a single solution of Eq. (8) for the evaluation of \( p_{(m_0, β)}(4, 4 + δ) \) with step 0.1. For this measure, we identify 17 distinct regeneration conditions reachable under condition \( c_{S_3} = 0 \), whose transient trees include 904 stochastic state classes. For \( α = 4 \) and \( δ = 1 \), adopting a step size \( h = 0.1 \), the enumeration of each tree is repeated \( \frac{5}{2} + 1 = 51 \) times for the evaluation of \( L_{ik}^{α+1; β+1} \), once for the evaluation of \( dL_{ik}^{α} \((t_m) \), and \( \frac{3}{1} + 1)(\frac{1}{1} + 2) = 492 \) times for the evaluation of \( dH_{ik}^{α} \), resulting in 491776 enumerated classes.

6 Conclusions

The verification of an interval until operator \( ϕ_1 U^{[α, β]} ϕ_2 \) 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], [40].

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 with \( α = 0 \) 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 [37]: 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 until \( α \).

We provided a solution based on the bivariate extension of Markov renewal equations, explicitly accounting for a satisfaction interval \([α, β]\). 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 [37]. 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 \( α(β - α) \), 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 regeneration 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 prob-
ablility 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 supports. The construction of these results highlighted important problems of probabilistic model checking of transient properties in regenerative systems, which can serve as the basis for further analysis techniques.

References


Marco Paolieri is a research scientist at the University of Florence, where he received his Ph.D. in Computer Science, Systems and Telecommunications (2015) and his B.S. and M.S. in Computer Engineering (2011). His research interests focus on stochastic modeling and quantitative evaluation of performance and reliability in concurrent and distributed systems.

Enrico Vicario (IEEE CS Member) is full professor of computer science and engineering at the University of Florence, since 2002. His scientific activity is focused on model-based methods for correctness verification and quantitative evaluation of concurrent systems with non-deterministic or stochastic temporal parameters. He also carries out a relevant activity of experimentation and technology transfer in software architectures and software development methods.

András Horváth was born in 1974 in Budapest where he received the M.Sc. degree in Computer Science from the University of Technology and Economics. From 1998 to 2002 he was a Ph.D. student supervised by Miklós Telek at the same university. From 2003 he is at the University of Turin (Italy). His research interests are in the area of stochastic processes including performance analysis of Markovian and non-Markovian models with application to communication networks and manufacturing systems.
appendix a

calculus of stochastic state classes

the succession relation $\tau_g \rightarrow \tau_{ag}$ 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], [31], [37], we report the main steps required by the calculus.

a stochastic state class $\Sigma = (m, D, f_{(\tau_{ag}, \tau)})$ includes the marking $m \in \mathbb{N}^p$, and the joint pdf $f_{(\tau_{ag}, \tau)}$ and support $D \subseteq \mathbb{R}^{n+1}$ of the random vector $\langle \tau_{ag}, \tau \rangle$, where:

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

as detailed in remark 1, the initial stochastic state class given by a regeneration condition $(m, \bar{d})$ 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 $\bar{d}$ and the sampling pdf $f_t$ of each enabled transition $t \in E(m)$, where $f_t$ is the pdf of the distribution $F(t)$. initially, the components of the vector $\langle \tau_{ag}, \tau \rangle$ are independent random variables and their joint pdf $f_{(\tau_{ag}, \tau)}$ is in product form.

given a stochastic state class $\Sigma = (m, D, f_{(\tau_{ag}, \tau)})$, a transition $\gamma \in E(m)$ and a firing interval $I$, the probability $\mu$ that $\gamma$ fires in $\Sigma$ at some time in $I$ can be computed as

$$\mu = \int_{D_a} \frac{W(t_1)}{\sum_{i: x_i = x_1} W(t)} f_{(\tau_{ag}, \tau)}(x_{age}, \bar{x}) \, dx_{age} \, d\bar{x} \tag{15}$$

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

$$D_a = \{ \langle x_{age}, \bar{x} \rangle \in D \mid (\forall j > 1, x_1 \leq x_j) \land (x_1 - x_{age}) \in I \}$$

is the set of values for $\langle \tau_{ag}, \tau \rangle$ such that the time to fire of $t_1$ is minimum, and the time of the next firing (obtained increasing $-\tau_{ag}$ by the sojourn time $\tau_1$) is contained in $I$. by integrating the pdf $f_{(\tau_{ag}, \tau)}(x_{age}, \bar{x})$ over this set of values, we obtain the succession probability $\mu$. the ratio of weights in eq. (15) accounts for the case in which $\gamma$ 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 $\Sigma'$ such that $\Sigma \xrightarrow{\gamma, \mu} \Sigma'$ 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 $f_{(\tau_{ag}, \tau)}$ of the vector of times to fire $\langle \tau_{ag}, \tau \rangle$ immediately after the firing of $t_1$ (and conditioned on this event) are 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 $\langle \tau_{ag}', \tau \rangle$ distributed over $D_a$ according to $f_{(\tau_{ag}', \tau)}(x_{age}, \bar{x}) = f_{(\tau_{ag}, \tau)}(x_{age}, \bar{x})/\mu$.

time advancement and marginalization. when $t_1$ fires, $\tau_{ag}$ and the times to fire $\tau_2^a, \ldots, \tau_n^a$ of enabled transitions are reduced by the random variable $\tau_a^a$ associated with $t_1$. the time to fire of $t_1$ is eliminated from the joint pdf $f_{(\tau_{ag}, \tau)}$ and from its support $D_a$ by integrating the joint pdf over all values of $\tau_1^a$. the pdf of the resulting random vector $\langle \tau_{ag}^a, \tau \rangle$ with $\tau_{ag} = \tau_{ag} - \tau_1^a$ and $\tau = (\tau_2 - \tau_1^a, \ldots, \tau_n - \tau_1^a)$ is

$$f_{(\tau_{ag}^a, \tau)}(x_{age}, x_2, \ldots, x_n) = \int_{L_1^n} \int_{U_1^n} f_{(\tau_{ag}, \tau)}(x_{age} + x_1, x_1, x_2 + x_1, \ldots, x_n + x_1) \, dx_1$$

where $U_1^n(x_{age}, x_2, \ldots, x_n)$ and $L_1^n(x_{age}, x_2, \ldots, x_n)$ are the maximum and minimum value of $x_1$ such that $(x_{age}, x_1, x_n) \in D_a$, respectively. as detailed in [30], $U_1^n$ and $L_1^n$ are piecewise functions; as a consequence, the pdf $f_{(\tau_{ag}^a, \tau)}$ is a piecewise function defined on a partitioning of its support $D_a = \{ (x_{age}, x_2, \ldots, x_n) \in \mathbb{R}^n \mid \exists x_1 \in \mathbb{R} : (x_{age}, x_1, x_n) \in D_a \}$.

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 updating the function $B(t_1)$) are eliminated one at a time from the vector $\langle \tau_{ag}, \tau \rangle$ by integrating the pdf $f_{(\tau_{ag}, \tau)}$ over all of their values in $D_a$. for example, the disabling of transition $t_2$ yields the random vector $\langle \tau_{ag}, \tau' \rangle$ with $\tau' = (\tau_3^b, \ldots, \tau_n^b)$ that has pdf $f_{(\tau_{ag}, \tau')} (x_{age}, \bar{x}) = \int_{L_2^n} \int_{U_2^n} f_{(\tau_{ag}, \tau)}(x_{age}, x_2, x_3, \ldots, x_n) \, dx_2$ with support $D' = \{ (x_{age}, x_3, \ldots, x_n) \in \mathbb{R}^{n-1} \mid \exists x_2 \in \mathbb{R} : (x_{age}, x_2, x_3, \ldots, x_n) \in D_a \}$.

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 $\langle \tau_{ag}, \tau' \rangle$. for example, the newly enabling of transition $t_{n+1}$ yields the random variable $\langle \tau_{ag}, \tau'' \rangle$ that has pdf $f_{(\tau_{ag}, \tau'')} (\tau_{ag}, x_3, \ldots, x_n, x_{n+1}) = f_{(\tau_{ag}, \tau')} (\tau_{ag}, x_3, \ldots, x_n) f_{(\tau_{ag}, \tau)} (x_{age}, x_1) \mu_{n+1}$ with support $D' = D_a \times \{ EFT(t_{n+1}), LFT(t_{n+1}) \}$.

example 5. for the stpn of fig. 1a, the successor of $\Sigma_0$ in fig. 3 through the firing of transition fail (without constraints on the firing time) is computed as follows. the new marking $2\text{free failed}$ is obtained from $2\text{free operational}$ by removing one token from the input place oper and adding one token to the output place failed. the integral of the pdf $f_{(\tau_{ag}, \tau)} (age, arrival, fail) = 0.1 \delta(\text{age}) \exp(-0.1 \text{fail})$ over $D_a = \{ \langle \text{age}, \text{arrival}, \text{fail} \rangle \in \mathbb{R}^3 \mid \text{age} = 0.1 \leq \text{arrival} \leq 2, \text{fail} \leq 0.1 \text{fail} \}$ gives $\mu = 1 + 10(0.02 - 0.01) \approx 1.0389$ and $f_{(\tau_{ag}, \tau)} (age, \text{arrival}, \text{fail}) = 0.7198 \delta(\text{age}) \exp(-0.1 \text{fail})$. time advancement and marginalization of timer fail give $f_{(\tau_{ag}, \tau)} (age, \text{arrival}) = \int_0^{0.7198} 0.7198 \delta(\text{age}+\text{fail}) \exp(-0.1 \text{fail}) \delta(\text{age}+\text{arrival})$. dfailed $= 0.7198 \exp(0.1 \text{age})$ and

$$D_b = \{ \langle \text{age}, \text{arrival} \rangle \in \mathbb{R}^2 \mid -2 \leq \text{age} \leq 0,$$

$$0 \leq \text{arrival} \leq 2, -2 \leq \text{age} - \text{arrival} \leq -1 \}.$$