UNIVERSITÀ
DEGLI STUDI
FIRENZE

PHD PROGRAM IN SMART COMPUTING
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE (DINFO)

New methods for quantitative evaluation of Markov regenerative models

Marco Biagi

Dissertation presented in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Smart Computing

PhD Program in Smart Computing
University of Florence, University of Pisa, University of Siena

New methods for quantitative evaluation of Markov regenerative models

Marco Biagi

Advisor:

Prof. Enrico Vicario

Head of the PhD Program:

Prof. Paolo Frasconi

Evaluation Committee:
Prof. Reinhard German, *University Erlangen-Nuremberg, Germany*
Prof. Peter Buchholz, *University Dortmund, Germany*

XXXI ciclo — October 2019

Acknowledgments

Firstly, I would like to thank my advisor, Enrico Vicario, for his precious advices during the whole PhD period. His guidance has been fundamental in the development of this thesis. I would like also to thank the members of my thesis supervisory committee, András Horváth and Mieke Massink, and the reviewers of the thesis, Reinhard German and Peter Buchholz, for their insightful comments and help. Moreover, I would like extend my thanks to Regione Toscana for their grant which allowed me to carry out these research.

I wish to express my sincere thanks to Laura Carnevali, Marco Paolieri and Francesco Santoni, I could not have imagined having better colleagues for my PhD study.

Finally, I want to thank my family and Francesca, that have been always there for me.

Abstract

The thesis describes the research that has been carried out in order to reduce the limitations of current analysis techniques for non-Markovian models, which led to a three-fold contribution.

The first contribution is a technique that allows to integrate different analysis techniques for the evaluation of kernels of a *MRGP*. Specifically, the state space of the underlying timed model is analyzed to identify epochs between regenerations and apply distinct methods for their analysis depending on the locally satisfied conditions. For epochs not amenable to existing methods, an adaptive approximation of kernel entries based on partial exploration of the state space is proposed, leveraging heuristics that permit to reduce the error on transient probabilities. This approach extends the class of models that can be analyzed, reduces errors committed by approximate analysis and allows one to automatize the selection of the analysis technique.

The second contribution is a technique for the computations of the equilibrium probability density functions (PDFs) for the continuous component of the state in *MRGP*. Equilibrium PDFs are derived as closed-form analytical expressions by applying the *Key Renewal Theorem* to stochastic state classes computed between regenerations. This techniques provides a basis to analyze system properties from the equilibrium such as survivability.

The last contribution, is an extension of the analysis of hierarchical semi-Markov processes with parallel regions, a technique that evaluates steady-state probabilities of models with multiple concurrent non-Markovian timers in a compositional way without the need of full state space generation. Specifically, the technique has been extended by removing some of its limitations and increasing its modeling power. By applying the time advancement mechanism known from stochastic state classes, exits in parallel regions with different time origins can be taken into account. Furthermore, exits can be put on state borders such that the model evolution depends on the exited region and a concept for history states is also presented. This significantly increases modeling power, such that the gap between semi-Markov processes with restricted modeling power and non-Markovian models without modeling restrictions but also with less efficient analysis is filled.

Contents

Chapter 1

Introduction

Over the last four decades, our life has become increasingly dependent on automated systems whose complexity is constantly growing. We all daily rely on various transportation systems, medical devices, communication systems and monitoring systems, which need to be highly reliable and available in order to not cause loss of life or money. Additionally, the new trend of *industry 4.0* (Lee et al., 2015; Lasi et al., 2014) has led to the emergence of even bigger cooperative systems whose correctness and efficiency heavily affects the competitiveness of involved companies. According to this the main challenge for modern engineering is to provide formalisms, techniques, and tools that will enable the design of correct and efficient systems regardless of their complexity.

These systems must fulfill quality, performance and *dependability* requirements. Dependability (Laprie, 1992) commonly identifies a set of quantitative measures that allow to asses how much we can depend on a specific system. Specifically, in the case of a safety-critical system, an important measure of dependability is the *reliability* (Trivedi and Bobbio, 2017), which is defined as the probability that no critical failure of the system occurs up to time t. In case of a highly availability system, another measure of interest is the *availability*, defined as the probability to find the system available at time t. This measure can for instance be used to asses the monetary loss due to an unavailability interval of time or the probability that QoS requirements will be met. Finally, it is worth to mention the *survivability* measure (Heegaard and Trivedi, 2009; Liu and Trivedi, 2006), which evaluates the performance of the system immediately after the occurrence of a critical failure. In particular, it allows one to study what happens to the system under the hypothesis that something catastrophic occurs and to measure how long it requires to return to its steady state.

The evaluation of such measures can be done directly on the system through testing or with *model-based* approaches. Model-based approaches have on one hand the advantage to allow one to study a system when it is in an early stage of design and also to study low probability events that are otherwise too rare to be observed on the real system within a reasonable time or too catastrophic to want to observe them (e.g. a tsunami on a nuclear power plant); on another hand, building a realistic model of a complex system is not trivial due to limitations in their analysis techniques that often requires less realistic simplifications of models which may lead to incorrect or unreliable results.

System models are represented through *formalisms*, which are languages that allow describing systems dynamics through an abstract representation. Many types of formalism exist, each one allowing one to model different aspects of systems. This thesis focuses on the study of systems characterized by discrete logic state and continuous time behavior, in particular to those systems having non-Markovian concurrent timers. Many continuous time models assume to have only exponentially distributed timers which are memoryless, that is, their future behavior does not depend on the time already elapsed since their enabling (Kulkarni, 2016). This assumption is typically referred as the *Markov assumption*, that from the analysis point of view dramatically helps since system states can be represented considering only the discrete logical state of the system, completely neglecting past history since it doesn't affect future. But most of real systems are by nature non-Markovian since timers accumulate memory of previous events and thus the remaining time before it runs out depends on the elapsed time. For instance, aging of systems is non-Markovian: given a component, the probability of a failure initially decreases over time due to "infant mortality", and then increases due to progressive degradation. If the model contains only a single non-Markovian timer concurrently enabled, it can be regarded as a *Semi-Markov process* (SMP) and its analysis is trivial (Kulkarni, 2016). But once again, real systems are often characterized by having several concurrently enabled timers and thus the assumption to have a single timer enabled in each state limits the number of real systems that can be modeled. Therefore we are interested in the study of models with concurrent non-Markovian timers.

The analysis of such systems can be performed through simulation or numerical techniques. Simulation can be applied regardless of the structure of the model under analysis, but when dealing with rare events it fails to give a sufficiently reliable estimation (Rubino and Tuffin, 2009). Rare event simulation techniques are a possible solution to that problem, in particular *importance splitting* (I-SPLIT) is easily applicable to the simulation of non-Markovian systems, but it needs the manual construction of a so called importance function which requires specific knowledge of the structure of the state space. Recently researchers tried to automatize the construction of the importance function (Budde et al., 2015), but although it seems a promising solution, currently these techniques do not always provide good results.

Many numerical techniques were proposed for the numerical analysis of non-Markovian models (German, 2000; Horváth et al., 2012; Choi et al., 1994; German et al., 1995; Amparore et al., 2014; German and Lindemann, 1994; Telek and Horváth, 2001; Lindemann and Thümmler, 1999), which can be applied depending on the type of distributions and the type of concurrency present in the model. In this thesis we focus on analysis techniques for models where the underlying process is a *Markov Regenerative Process* (MRGP), that are processes which always reach a *regeneration point* with probability 1. A regeneration point is a time instant where the Markov property is satisfied, thus the future behavior doesn't depend on what happened before such time instant.

All the analysis techniques for the analysis of MRGP have limitations. Specifically, most works address the subclass where at most a single non-Markovian timer is enabled in each state (*enabling restriction*), so that it can be analyzed by studying the *Continuous Time Markov Chain* (CTMC) subordinated to the activity interval of the non-Markovian timer (Choi et al., 1994; German et al., 1995; Amparore et al., 2014). The method of supplementary variables (German and Lindemann, 1994; Telek and Horváth, 2001) might in principle encompass the case of multiple concurrently enabled GEN timers, but practical feasibility restrains applicability under the enabling restriction. Sampling at equidistant time points (Lindemann and Thümmler, 1999) permits evaluation of models where all timers have either deterministic (DET) or exponentially distributed (EXP) durations. The method of stochastic state classes (Horváth et al., 2012) enables exact analysis of models with multiple concurrent non-Markovian timers, when a regeneration is reached within a bounded number of discrete events (*bounded regeneration restriction*), that is not always the case in real systems. Moreover, an approximated version of the technique allows performing the analysis beyond the bounded regeneration restriction. Furthermore, another technique was developed for a specific formalism derived from UML state machines (Group, 2018), where the particular hierarchical structure of the model can be exploited to efficiently analyze it (Homm and German, 2016), but expressiveness of the resulting formalism is very limited greatly reducing its applicability.

1.1 Contributions

In this thesis we describe the contributions made in order to increase the applicability of the before mentioned techniques, which led to a three-fold contribution.

The first contribution is a technique that allows to integrate different analysis techniques for the evaluation of kernels of a *MRGP*. Specifically, the state space of the underlying timed model is analyzed to identify epochs between regenerations and apply distinct methods for their analysis depending on the locally satisfied conditions. For epochs not amenable to existing methods, an adaptive approximation of kernel entries based on partial exploration of the state space is proposed, leveraging heuristics that permit to reduce the error on transient probabilities. This approach extends the class of models that can be analyzed, reduces errors committed by approximate analysis and allows one to automatize the selection of the analysis technique.

The second contribution is a technique for the computations of the equilibrium probability density functions (PDFs) for the continuous component of the state in a *MRGP*. Equilibrium PDFs are derived as closed-form analytical expressions by applying the *Key Renewal Theorem* to stochastic state classes computed between regenerations. This techniques provides a basis to analyze system properties from the equilibrium such as survivability (Heegaard and Trivedi, 2009; Liu and Trivedi, 2006).

The last contribution, is an extension of the analysis of hierarchical semi-Markov processes with parallel regions (Homm and German, 2016), a technique that evaluates steady-state probabilities of models with multiple concurrent non-Markovian timers in a compositional way without the need of full state space generation. Specifically, we extend the technique by removing some of its limitations and increasing its modeling power. By applying the time advancement mechanism known from stochastic state classes, exits in parallel regions with different time origins can be taken into account. Furthermore, exits can be put on state borders such that the model evolution depends on the exited region and a concept for history states is also presented. This significantly increases modeling power, such that the gap between semi-Markov processes with restricted modeling power and non-Markovian models without modeling restrictions but also with less efficient analysis is filled.

The first two contributions adopt the formalism of *Stochastic Time Petri Nets* (STPN), that will be described in Chapter 3, along with some techniques for their analysis. The STPN formalism is built on top of the *Time Petri Nets* (TPN) formalism (Berthomieu and Diaz, 1991), adding the concept of probabilities. According to this, in Chapter 2, we start introducing the TPN formalism and also the *non-deterministic analysis*, which will be used in the approach described in Chapter 4. Chapters 2 and 3 will provide the basis to understand the approaches described in following Chapters 4, 5 and 6, which form the main contributions of this work. Finally we draw our conclusions in Chapter 7 and describe possible future directions of study.

Chapter 2

Time Petri Nets: formalism and analysis

The *Time Petri Net* (TPN) formalism (Berthomieu and Diaz, 1991), is derived from *Petri Net*(PN) decorating transitions with durations, expressed as timers having execution times taking values non-deterministically within dense intervals. According to this, models represented through this formalism doesn't represent a probabilistic measure of its acceptable behaviors.

In this chapter, we will first describe the TPN formalism and then the *non- deterministic analysis* of TPN that was developed in (Vicario, 2001). Such technique, will be used as a basis of the approach described in Chapter 4.

2.1 The Time Petri Nets formalism

The TPN formalism is an extension of the Petri Nets (PN) formalism that extends it with a notion of time (Berthomieu and Diaz, 1991). In the following, its syntax and semantics will be illustrated.

2.1.1 Syntax

The syntax of a TPN is defined as follows:

Definition 1. *A TPN is a tuple $\langle P, T, A^-, A^+, A^i, m_0, EFT, LFT, U\rangle$ where P is the set of places, T is the set of transitions, $A^- \subseteq P \times T$ is the set of preconditions, $A^+ \subseteq T \times P$ is the set of postconditions, $A^i \subseteq P \times T$ is the set of inhibition conditions, $m_0 \in \mathbb{N}^P$ is the initial marking, $EFT : T \to \mathbb{Q} \geqslant 0$ and $LFT : T \to \mathbb{Q} \geqslant 0 \cup \{+\infty\}$ associate each transition with a firing interval $[EFT(t), LFT(t)]$, $U(t) : \mathbb{N}^P \to \mathbb{N}^P$ is an update function which given a transition t, associates each marking with a new marking.*

Figure 2.1 shows an example of TPN. The TPN has 5 places $P = \{p_1, p_2, p_3, p_4,$

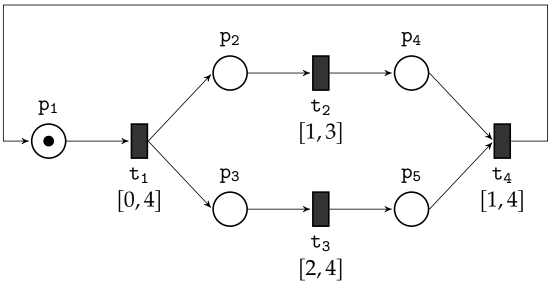


Figure 2.1: Example of TPN

$p_5\}$, represented by circles, and 4 transitions $T = \{t_1, t_2, t_3, t_4\}$, represented by thick black bars. A marking identifies the number of tokens present in each place of the TPN and it represents the logical state of the system. Initially, one token is present in the place p_1, thus the initial marking is $m_0 = [1, 0, 0, 0, 0]$. For simplicity, in the following, we will refer to this marking as $m_0 = [1p_0]$ or as $m_0 = [p_0]$, avoiding to specify the number of tokens of places where there is exactly one token. Transitions t_1, t_2, t_3 and t_4 are associated with firing intervals of $[0, 4]$, $[1, 3]$, $[2, 4]$ and $[1, 4]$, respectively. Preconditions and postconditions are represented by arrows. For instance transition t_1 has a precondition $\langle p_1, t_1\rangle$ and has two postconditions $\langle t_1, p_2\rangle$ and $\langle t_1, p_3\rangle$. Inhibitor arcs and update functions are not present in the example, but they are usually represented as a line with a small circle at its end, and annotated as text next to transitions, respectively. A place p is called *input place* for a transition t if $\langle p, t\rangle \in A^-$, *output place* for the transition if $\langle t, p\rangle \in A^+$, *inhibitor place* for the transition if $\langle p, t\rangle \in A^i$.

2.1.2 Semantics

The behavior of a TPN is defined by the *state* and by the *transition rule*. The state $s = \langle m, \bar{\tau}\rangle$ is composed by a marking m and by a vector of times to fire $\bar{\tau}$. The vector of times to fire associates each enabled transition, in the current marking, with a time to fire $\bar{\tau} = [\tau_1, ..., \tau_n]$, where $\tau_j \in \mathbb{R} > 0$ is the time to fire of the j-th enabled transition and n is the number of enabled transitions. For simplicity in the following, we will also write $\bar{\tau}(t_0)$ to refer to the time to fire associated to the transition t_0.

The state of the model evolves according to a transition rule composed by three clauses: *firability*, *progress* and *firing*:

- *firability*: a transition t_0 is *enabled* in a specific marking m, if each of its input places contains at least one token and no place connected through an inhibitor arc contains any token. We call $T_E(m)$ the set of enabled transitions in the marking m.

- *progress*: a transition t_0 is *firable* if it is enabled and its time to fire $\bar{\tau}(t_0)$ in the current state $s = \langle m, \bar{\tau} \rangle$ is not greater than the time to fire $\bar{\tau}(t_j)$ of any other enabled transition:
$$\bar{\tau}(t_0) \leqslant \bar{\tau}(t_j) \quad \forall t_j \in T_E(m) \tag{2.1}$$
We call $T_F(s)$ the set of firable transitions in the state s. A transition $t_i \in T_F(s)$ among the firable transition will fire as next event at time $\bar{\tau}(t_i)$.

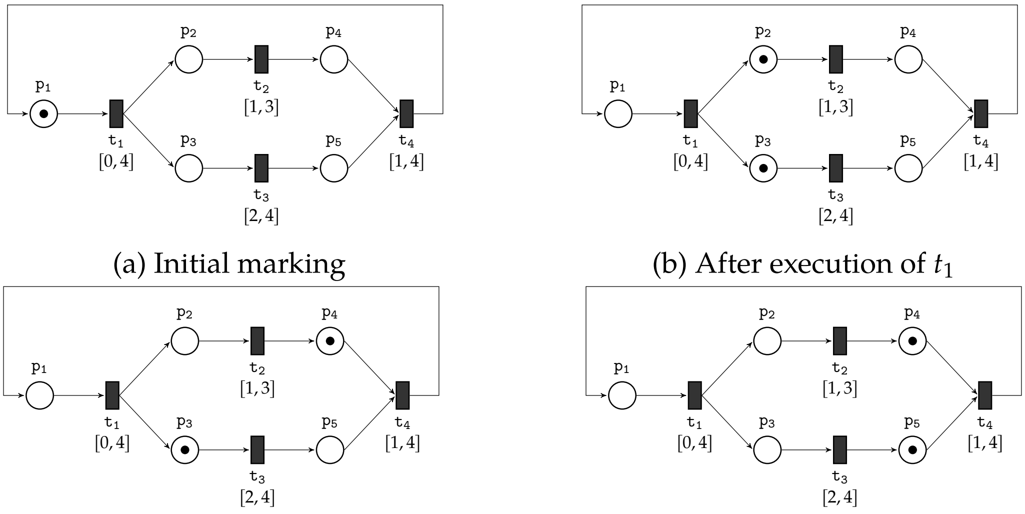
- *firing*: when t_0 fires, the state $s = \langle m, \bar{\tau} \rangle$ is replaced by a new state $s' = \langle m', \bar{\tau}' \rangle$. We write this change as $s \xrightarrow{t_0} s'$.

  The marking m' is computed by removing a token from each of the input places of t_0, leading to a temporary marking called m^{tmp}, and by adding a token to each of the output places of t_0. Formally, given that the transition t_0 fires, the number of tokens in a place p in the new marking can be derived according to equations 2.2 and 2.3.
$$m^{tmp}(p) = \begin{cases} m(p) - 1 & \text{if } \langle p, t_0 \rangle \in A^- \\ m(p) & \text{otherwise} \end{cases} \tag{2.2}$$
$$m'(p) = \begin{cases} m^{tmp}(p) + 1 & \text{if } \langle t_0, p \rangle \in A^+ \\ m^{tmp}(p) & \text{otherwise} \end{cases} \tag{2.3}$$
Moreover, if an update function is associated to transition t_0, it is also applied to m^{tmp}.

  The marking m^{tmp} is relevant to classify transitions enabled in the arriving state which can be *persistent* or *newly enabled* that in turns affects the derivation of the vector of times to fire $\bar{\tau}'$.

  A transition t_p is persistent if it is enabled in m, m^{tmp} and m' and if it is not the fired transition t_0. In that case the remaining time to fire $\tau(t_p)$ is derived by reducing $\bar{\tau}(t_p)$ by the value $\bar{\tau}(t_0)$:
$$\bar{\tau}'(t_p) = \bar{\tau}(t_p) - \bar{\tau}(t_0) \tag{2.4}$$
Any transition t_e enabled by m' but not persistent is newly enabled and its time to fire $\bar{\tau}'(t_e)$ is set to a non-deterministic value sampled in the static firing interval $\bar{\tau}'(t_e) \in [EFT(t_e), LFT(t_e)]$. We call $T_P(s)$ and $T_N(s)$, the set of transitions persistent and newly enabled in a state s, respectively.

  Finally, the time to fire of any transition t_d that was enabled in m but is no more enabled in m' is not present in the vector $\bar{\tau}'$ and we say that t_d has been disabled.



(a) Initial marking

(b) After execution of t_1

(c) After execution of t_2

(d) After execution of t_3

Figure 2.2: Example of TPN step by step execution

2.1.3 Example of TPN execution

For the sake of illustration, we show in Figure 2.2 a step by step example of the execution of the model presented in Figure 2.1. Note that for simplicity, Figure 2.2 shows only markings and not states. Initially, the state (Figure 2.2a) is $s_0 = \langle m_0, \bar{\tau}_0 \rangle = \langle [p_1], [\bar{\tau}(t_1)] \rangle$ where $\bar{\tau}_0$ is composed only by the time to fire of the only enabled transition t_1. This value is sampled non-deterministically in the interval $[0, 4]$, for instance suppose that $\bar{\tau}(t_1) = 3$. Since t_1 is the only enabled transition, it is also the only firable transition and it fires at time $t = \bar{\tau}(t_1) = 3$. The marking is updated and the new reached state (Figure 2.2b) is $s_1 = \langle m_1, \bar{\tau}_1 \rangle = \langle [p_2, p_3], [\bar{\tau}(t_2), \bar{\tau}(t_3)] \rangle$. The two transitions t_2 and t_3 are enabled concurrently and are both newly enabled. Suppose for instance that the vector of sampled times to fire is $\bar{\tau}_1 = [2, 3.5]$. With these values, t_2 will fire first since $\bar{\tau}(t_2) < \bar{\tau}(t_3)$, but it is worth to notice that in general the model accepts also the opposite behavior, with a different sampled vector of times to fire. When t_2 fires the system reaches at time $t = 5$ the new state $s_2 = \langle m_2, \bar{\tau}_2 \rangle = \langle [p_3, p_4], [\bar{\tau}(t_3)] \rangle$, where $\bar{\tau}(t_3) = 1.5$ because it is persistent and updated according to Equation 2.4. This time it is t_3 to fire at $t = 6.5$, allowing to reach $s_3 = \langle m_3, \bar{\tau}_3 \rangle = \langle [p_4, p_5], [\bar{\tau}(t_4)] \rangle$. Finally t_4 will fire since is the only enabled transition and the system will reach again the initial marking.

2.2 Non-deterministic analysis

The non-deterministic analysis is an enumerative technique which supports reachability and timeliness analysis in TPN models (Vicario, 2001). The technique uses equivalence classes to enable discrete and compact enumeration of the state space. It will be used in Chapter 4 to enumerate reachable states and identify subsets of states which satisfy specific restrictions.

2.2.1 State classes

The technique is based on the concept of *state class*, a generalization of the syntactic structure of the state of the TPN, defined as follows:

Definition 2. *A state class is a pair $\langle m, D \rangle$ where m is the marking and D is the support for the vector of remaining times to fire of enabled transitions.*

The support D is expressed as a set of inequalities on the difference between firing times of different transitions and referring firing times to a ground reference value $\bar{\tau}(t_*)$ (Note that t_* is not a real transition, but is used to identify the reference time of the domain, i.e. the time instant at which the class is entered). In particular, they are expressed in the form shown in Equation 2.5.

$$D_m = \Big\{ \bar{\tau}(t_i) - \bar{\tau}(t_j) < b_{ij} \quad \forall t_i, t_j \in T_E(m) \cup \{t_*\} \quad \text{with } t_i \neq t_j \qquad (2.5)$$

Where $b_{ij} \in \mathbb{R} \cup \{\infty\}$ are the coefficients, $\bar{\tau}(t_i)$ and $\bar{\tau}(t_j)$ are the remaining firing times and $\bar{\tau}(t_*)$ is the reference time. This structure is usually called *Difference Bound Matrix*(DBM). The difference between a state class and the state of the TPN presented in section 2.1.2, is that the class is an equivalence class that specifies a support of possible times to fire while the state has sampled a specific value for each enabled transition. Thus a state class is a collection of TPN states.

2.2.2 Successors evaluation

Starting from the initial marking m_0 and the initial support D_0, the technique allows enumerating the reachability relation among state classes yielding a *State Class Graph*(SCG), which represents the continuous set of possible executions of the model. Specifically, the reachability relation between state classes is defined as follows.

Definition 3. *A state class S_{child} is reachable from a state class S_{parent} through a transition t_0 if and only if S_{child} collects all and only the states that are reachable from some states collected in S_{parent} through a feasible firing of t_0.*

Considering this reachability relation, the successors of a state class can be identified according to Theorem 1 which has been proved in (Vicario, 2001).

Theorem 1. *Given a state class $S_{parent} = \langle m_{parent}, D_{parent} \rangle$, with D_{parent} represented as:*

$$D_{parent} = \Big\{ \bar{\tau}(t_i) - \bar{\tau}(t_j) < b_{ij} \quad \forall t_i, t_j \in T_E(m_{parent}) \cup \{t_*\} \quad with\ t_i \neq t_j \tag{2.6}$$

Transition t_0 is an outgoing arc for S_{parent} if and only if $t_0 \in T_E(m_{parent})$ and $b_{i0} \geqslant 0 \quad \forall t_i \in T_E(m_{parent})$.

This theorem allows evaluating the set of outgoing arcs from the state class S_{parent}, that is the set of transitions that are firable. Specifically the theorem states that a transition t_0 is firable, and thus an output arc, if and only if the support D_{parent} accepts solutions in which the firing time $\bar{\tau}(t_0)$ is not greater than any other enabled transition firing time. This condition is tested simply by checking if b_{i0} can be greater than or equal to 0, because since b_{i0} represents the maximum delay of the firing of t_i after t_0, if it is positive is acceptable that it executes after t_0.

After having identified all possible outgoing arcs, the reached classes need to be evaluated. The successor computation is performed by evaluating the marking m_{child} as shown in section 2.1.2, and then evaluating D_{child}, assuming that a transition t_0 fires. Specifically, D_{child} is computed removing disabled transition and including enabled transitions with firing intervals set to their static values. For each persistent transitions t_i, the firing interval must be constrained within the minimum and the maximum delay that t_i can have after the firing of t_0, under the constraint that $\bar{\tau}(t_0) < \bar{\tau}(t_i)$. Let $\bar{\tau}_p$ be the vector of remaining times to fire in the parent class and $\bar{\tau}_c$ be the vector of remaining times to fire in the successor class. The time to fire of a persistent transition t_i after the firing of a transition t_0 is subject to the constraint $\bar{\tau}_c(t_i) - \bar{\tau}_c(t_*) = \bar{\tau}_p(t_i) - \bar{\tau}_p(t_0)$. According to this, D_{child} can be evaluated according to the following proposition (Vicario, 2001):

Proposition 1. *Let t_0 be a fireble transition in class $S_{parent} = \langle m_{parent}, D_{parent} \rangle$, with firing domain D_{parent}.*

$$D_{parent} = \Big\{ \bar{\tau}(t_i) - \bar{\tau}(t_j) < b_{ij} \quad \forall t_i, t_j \in T_E(m_{parent}) \cup \{t_*\} \quad with\ t_i \neq t_j \tag{2.7}$$

Let $D^{t_0}_{parent}$ be the restricted firing domain which augments the firing domain D_{parent} with a set of additional constraints imposing $\bar{\tau}(t_0)$ to be not longer than any transition $\bar{\tau}(t_i)$ enabled in $T_E(m_{parent})$:

$$D^{t_0}_{parent} = \begin{cases} \bar{\tau}(t_i) - \bar{\tau}(t_j) < b_{ij} \\ \bar{\tau}(t_0) - \bar{\tau}(t_j) < \min 0, b_{oj} \\ \forall t_i, t_j \in T_E(m_{parent}) \cup \{t_*\} \quad with\ t_i \neq t_j \end{cases} = \tag{2.8}$$

$$\begin{cases} \bar{\tau}(t_i) - \bar{\tau}(t_j) < B_{ij} \\ \forall t_i, t_j \in T_E(m_{parent}) \cup \{t_*\} \quad with\ t_i \neq t_j \end{cases}$$

The firing domain D_{child} of successor state class $S_{child} = \langle m_{child}, D_{child} \rangle$, reached through the firing of t_0, is given by the following equation:

$$D_{child} = \begin{cases} \bar{\tau}(t_i) - \bar{\tau}(t_*) \leqslant B_{i0} \\ \bar{\tau}(t_*) - \bar{\tau}(t_i) \leqslant B_{0i} \\ \bar{\tau}(t_i) - \bar{\tau}(t_j) \leqslant B_{ij} \\ \bar{\tau}(t_k) - \bar{\tau}(t_*) \leqslant LFT(t_k) \\ \bar{\tau}(t_*) - \bar{\tau}(t_k) \leqslant -EFT(t_k) \\ \bar{\tau}(t_k) - \bar{\tau}(t_i) \leqslant LFT(t_k) + B_{0i} \\ \bar{\tau}(t_i) - \bar{\tau}(t_k) \leqslant B_{i0} - EFT(t_k) \\ \bar{\tau}(t_k) - \bar{\tau}(t_h) \leqslant LFT(t_k) - EFT(t_h) \\ \bar{\tau}(t_h) - \bar{\tau}(t_k) \leqslant LFT(t_h) - EFT(t_k) \\ \\ \forall t_i, t_j \in T_P(m_{child}) \\ \forall t_k, t_h \in T_N(m_{child}) \end{cases} \tag{2.9}$$

These results allow computing the whole SCG of a TPN that describes all possible behaviors of the model according to the time restrictions specified in its definition. Note also that in each state, we can easily identify which transitions are enabled and which of them are newly enabled or persistent, a property that will be used in Chapter 4. For a more formal discussion on the construction of the SCG and on the complexity of this operation we refer to (Vicario, 2001). Additionally, an example of SCG evaluation is reported in Appendix A.

Finally, it is useful to know that it has been proven that if $EFT(t) \in \mathbb{Q} \geqslant 0$ and $EFT(t) \in \mathbb{Q} \geqslant 0 \cap \{\infty\}$ for every transition t, then the SCG is finite provided that the model generates a finite number of markings (Horváth et al., 2012).

Chapter 3

Stochastic Time Petri Nets: formalism and analysis

Many formalisms have been developed for the representation of models with concurrent non-Markovian timers (Vicario et al., 2009; D'Argenio et al., 2016; Jurdzinski et al., 2014; Buchholz and Telek, 2013; Homm and German, 2016; Biagi et al., 2018), each one having different advantages. In the first chapters of this thesis (in Chapter 6 we will use a different formalism) we adopt the *Stochastic Time Petri Nets* (STPN) formalism (Vicario et al., 2009) because the stochastic state classes analysis (Horváth et al., 2012), widely used in this work, was originally developed and implemented on it (Carnevali et al., 2011). Nevertheless, it worth noting that the same concepts that will be developed here, could be applied to other equivalent formalisms (Ballarini et al., 2013).

In this chapter we will first recall the STPN formalism, then we will recall to mind the concept of underlying processes and their classification, finally analysis techniques of interest in this work will be briefly described.

3.1 STPN

The STPN formalism allows one to describe continuous time systems with concurrent generally distributed timers, possibly with bounded support. It is built on top of the TPN formalism, extending it with stochastic parameters. Highlighting this relationship is a key point to understand the contribution presented in Chapter 4.

In the following we recall STPNs syntax and semantics.

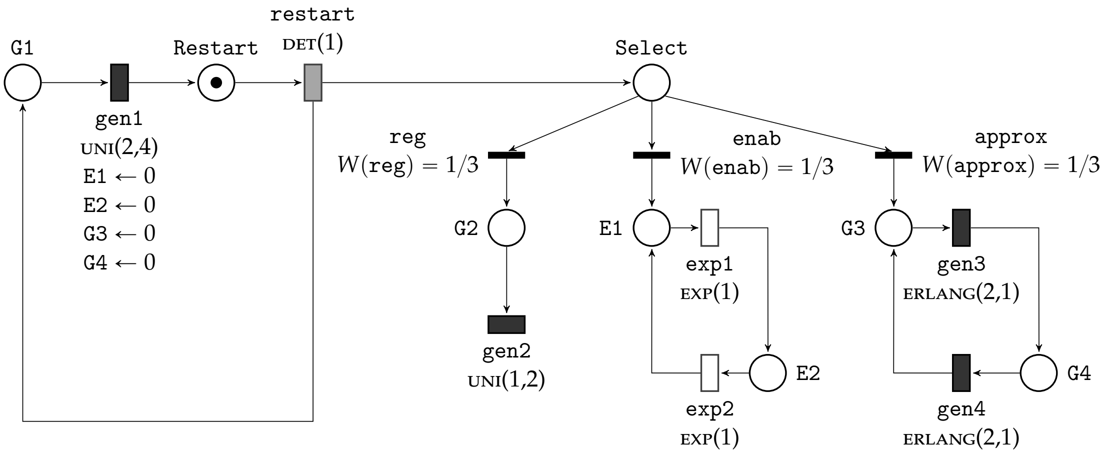


Figure 3.1: Example of STPN

3.1.1 Syntax

Definition 4. *An STPN is a tuple $\langle P, T, A^-, A^+, A^i, m_0, EFT, LFT, U, F, W\rangle$ where the first nine elements define a TPN, while F and W extend it associating each transition t with a Cumulative Distribution Function (CDF) $F(t)$ with support $[EFT(t), LFT(t)]$ and with a weight $W : T \to \mathbb{R} > 0$, respectively.*

The definition clearly shows how STPNs extend TPNs adding stochastic parameters, specifically adding a probability measure to the transition supports that were previously sampled non-deterministically. Additionally, adding weights allows a modeler to represent probabilistic choices. It is worth noticing that from the qualitative perspective, the set of behaviors that a model can generate is the same of the underlying TPN. The extension relies on having added a stochastic characterization of such behaviors allowing to evaluate a quantitative measure of the event that a specific behavior occurs.

Usually transitions are classified on the basis of the type of distribution assigned to them. A transition t is *immediate*(IMM) if $EFT(t) = LFT(t) = 0$, otherwise it is called *timed*. A timed transition t is *exponential*(EXP) if $F_t(x) = 1 - e^{-\lambda x}$ with support $[0, +\infty]$ and $\lambda \in \mathbb{R} > 0$ and *general*(GEN) otherwise. A general transition t is *deterministic*(DET) if $EFT(t) = LFT(t) > 0$, otherwise is called *distributed*. For each distributed transition t, we assume that F_t is the integral of the *Probability Density Function*(PDF), i.e. $F_t(x) = \int_0^x f_t(y)dy$.

An example of STPN is shown in Figure 3.1. Transitions are graphically represented by different symbols depending on the type of associated CDF. Exponentially distributed transitions (e.g. exp1 and exp2 in Figure 3.1) are represented by white thick bars. Deterministic transitions (e.g. restart) are represented by a grey thick bar while immediate transitions (e.g. reg, enab and approx), are represented as a black thin bar. Generally distributed transitions (e.g. gen1, gen2, gen3 and gen4) are finally represented by black thick bars. Immediate transitions have their weight explicitly reported in the figure. Specifically, transitions named `reg`, `enab` and `approx` have weights of $W(\texttt{reg}) = 1/3$, $W(\texttt{enab}) = 1/3$, $W(\texttt{approx}) = 1/3$, respectively. Finally, an update function is associate to transition `gen1`, which removes all tokens present in places `E1`, `E2`, `G3` and `G4` when it executes. The update function is annotated as text below the transition.

3.1.2 Semantics

Also the semantics of STPNs can be regarded as an extension of the semantics of TPNs, adding the concept of probability. The behavior of a STPN is defined by the state and by the transition rules. As in TPN, a STPN state $s = \langle m, \bar{\tau} \rangle$ is composed by a marking m and by a vector of times to fire $\bar{\tau}$ of enabled transitions. A transition t_0 is *enabled* if each of its input places contains at least one token and no inhibiting places contains any token. The set of enabled transitions in a marking m is named $T_E(m)$. A transition t_0 is *firable* if its time to fire $\tau(t_0)$ is not greater than that of any other enabled transition. The set of all firable transitions in a state s is called $T_F(s)$. Where multiple transitions are fireable at the same time, the choice is resolved through a random switch using the weights W.

$$P\{\text{"}t_0 \text{ fires first"}\} = \frac{W(t_0)}{\sum_{t_i \in T_F(s)} W(t_i)} \tag{3.1}$$

When a transition t_0 fires, the state $s = \langle m, \bar{\tau} \rangle$ is replaced by a new state $s' = \langle m', \bar{\tau'} \rangle$. We write this change as $s \xrightarrow{t_0} s'$. Marking m' is derived from m again from equation 2.2 and 2.3 as seen for TPN. Additionally, as in TPN, m, m' and m^{tmp} are used to classify enabled transitions as persistent or as newly enabled, and this classification is exploited in order to derive $\bar{\tau}'$, that is the vector of times to fire after the firing of a transition t_0. Specifically, the new time to fire $\bar{\tau}(t_p)'$ of each persistent transition t_p is evaluated reducing its time to fire $\bar{\tau}(t_p)'$ by the time elapsed in the previous state as shown in equation 3.2, that is equivalent to what was defined for TPNs in equation 2.4.

$$\tau'(t_p) = \tau(t_p) - \tau(t_0) \tag{3.2}$$

Finally, the time to fire $\bar{\tau}(t_e)'$ of each newly enabled transition t_e is sampled in the firing interval according to the CDF $F(t_e) = \mathcal{F}_{t_e}$:

$$\begin{gathered} EFT(t_e) \leqslant \bar{\tau}(t_e)' \leqslant LFT(t_e) \\ Prob\{\bar{\tau}(t_e)' \leqslant x\} = \mathcal{F}_{t_e}(x) \end{gathered} \tag{3.3}$$

Note that in TPNs, $\bar{\tau}(t_e)'$ for newly enabled transitions was instead chosen non-deterministically in the support interval.

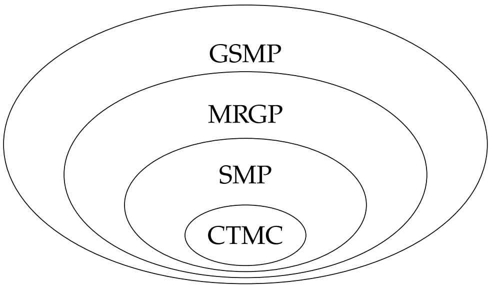


Figure 3.2: Relationship between stochastic processes classes

3.2 Underlying process of a STPN

When studying continuous time systems with discrete logical states, the most common measures of interest are the probability to be in a certain logical state, at time t or at steady state. In STPNs the logical state is represented by the marking whose evolution over time identifies an underlying continuous time stochastic process (Ciardo et al., 1994). According to this, analysis of an STPN is usually focused on the analysis of its underlying *marking process*, that is defined as $\{M(t), t \geqslant 0\}$, where where $M(t)$ is the marking at time t. The marking process can be classified based on its properties which depend on the type of distributions of timed transitions and on their relationships (Ciardo et al., 1994; Choi et al., 1994). Depending on the class of the process, different techniques can be applied, thus the identification of the process class is crucial to select which analysis technique can be properly applied to a model.

Specifically if the model includes only exponentially distributed timers, the underlying process is a *Continuous Time Markov Chain* (CTMC). If instead non- Markovian timers are presents, the process can accumulate memory over time and it can be classified in different classes depending on the presence of *regeneration points*. A regeneration point is a state where all non-Markovian timers lose their memory, which basically depends on the persistence of non-Markovian transitions. If non-Markovian transitions never persist at any firing, the process regenerates at each step and is thus a *Semi-Markov process* (SMP) (Kulkarni, 2016). If the process always regenerates with probabilities 1, it is in the class of Markov Regenerative Processes (MRGP). Whereas, when there is a non-null probability that a regeneration point is never reached, then the process becomes a generalized semi-Markov process (GSMP). As shown in Figure 3.2, it is worth noticing the relationship between these process classes. In CTMCs, timers never persist and according to this they regenerate at each firing, thus they can be regarded as a sub class of SMP and MRGP. In SMPs the process regenerates after each firing and thus can be regarded as a sub class of MRGP. It is important to understand this relationship because techniques that can be applied to analyze MRGPs can be used also for the analysis of SMPs and CTMCs, but in general are less efficient since they can't rely on the structural assumptions that techniques developed for such sub-class of process can exploit.

In this thesis we focus on the analysis of Markov Regenerative Processes. Techniques for their analysis, that are of interest for this work, will be briefly described in the following section.

3.3 Regenerative transient analysis with Markov renewal equations

Many techniques for the evaluation of transient and steady state measures of MRGPs have been developed (German, 2000; German et al., 1995; Horváth et al., 2012; Biagi et al., 2018; Choi et al., 1994; Amparore et al., 2014; Martina et al., 2016), but each of these techniques has some limitations. Most of the numerical transient analysis techniques are typically based on the solution of *Markov renewal equations* defined by *global kernels* and *local kernels* (Kulkarni, 2016), which respectively characterize the occurrence of regenerations and transient probabilities between them.

The marking process $\{M(t), t \geqslant 0\}$, specifies the logical location of an STPN at each time instant. If the marking process is an MRGP, its transient evolution can be completely characterized by:

1. the initial marking. Or in a more general setting, the **probabilities of the initial markings**

2. a **local kernel** matrix $L_{ij}(t) := P\{M(t) = j, T_1 > t | M(0) = i\}$, where T_1 is the next time at which the process will regenerate. $L_{ij}(t)$ is the probability to be in state j at time t, given that the process started in i and it has not regenerated yet. According to this the local kernel captures the evolution in the period between two subsequent regeneration points, that in the following will be called *regenerative epoch*.

3. a **global kernel** matrix $G_{ik}(t) := P\{M(T_1) = k, T_1 < t | M(0) = i\}$, which characterize the occurrence of regenerations. $G_{ik}(t)$ is the probability that, starting from a regeneration i, the first regeneration is reached at marking k within time t.

Specifically, transient probabilities of markings $\pi_{ij}(t) := P\{M(t) = j | M(0) = i\}$ are the solution of the following set of equations (Kulkarni, 2016):

$$\begin{gathered} \pi_{ij}(t) := P\{M(t) = j \,|\, M(0) = i\} = \\ L_{ij}(t) + \sum_{k \in \Theta} \int_0^t g_{ik}(x)\, \pi_{kj}(t - x)\, dx \end{gathered} \tag{3.4}$$

Where $g_{ik}(x) := dG_{ik}(x)/dx$ and Θ is the set of regenerative markings of the model. Equation 3.4 constitutes a set of Volterra integral equations of the second kind, known as generalized Markov renewal equations. While Equation 3.4 can be solved numerically by discretization, evaluation of the kernels is hard and can be done exactly only for sub-classes of MRGP models where some structural restrictions on the stochastic model are satisfied.

3.3.1 Kernels evaluation under enabling restriction

One of the most popular technique for the derivation of kernels is the *regenerative transient analysis under enabling restriction* (Choi et al., 1994; German et al., 1995), usually called more simply *analysis under enabling restriction*.

Definition 5. *A model is under enabling restriction if at most a single GEN timer is enabled in each state.*

If the enabling restriction is satisfied, each regenerative epoch of the process behaves either as a CTMC, if only EXP transitions are enabled in the current state, or as a CTMC subordinated to the activity interval of a GEN transition, if a GEN transition is enabled. If in a state, no GEN transitions are enabled, the process regenerates at the next firing. If a GEN transition is enabled, the process will regenerate when such transition fires or is disabled by the firing of an EXP transition. According to this, the underlying process of a model under enabling restriction is a MRGP. In the following we illustrate how kernels can be derived if the enabling restriction is satisfied (German et al., 1995).

First, if the model contains IMM timers, it is important to introduce the concept of *tangible* and *vanishing* markings.

Definition 6. *If the set of enabled transitions $T_E(m)$ of the marking m contains one or more IMM transitions, the marking is said to be vanishing, otherwise the marking is tangible.*

The reason behind this naming is that if an IMM transition is enabled, the marking process immediately evolves to the subsequent marking due the immediate execution of one of the enabled IMM transitions. In so doing, the time spent in a vanishing marking is zero, while the time spent in a tangible marking is greater than zero.

Function $M(t)$ of the marking process denotes the tangible marking at time t. The analysis technique can be applied if the set of tangible markings $\mathcal{S}$ is finite, thus the state space can be enumerated. Let T^G be the set of all GEN transitions in the STPN. The state space $\mathcal{S}$, can be partitioned into multiple disjoint sets $\mathcal{S}^E$, that is the set of marking where only EXP transitions are enabled, and a set $\mathcal{S}^g$ for each transition $g \in T_G$, that is the set of states where only g is enabled. Additionally, each state $i \in \mathcal{S}^g$ can be classified based on the state that can be reached from them. Let $\mathcal{S}(i)$ be the set of tangible states that are not regenerative and can be reached from i through the firing of an EXP transition that does not disable g. Let $\mathcal{S}_\epsilon(i)$ be the set of tangible states that are regenerative and can be reached from i through the firing of an EXP transition that disables g. Finally, let $\mathcal{S}_g(i)$ be the set of tangible states that are regenerative and can be reached from i through the firing of g. We need also to define Q as the infinitesimal generator matrix, defined by the rates of the EXP transitions, where q_{ij} with $i \neq j$ denotes the rate from marking i to marking j and q_{ii} the negative sum of all the other elements of the row. Moreover, transient probabilities of the *CTMC* subordinated to the activity of a GEN transition are denoted by $\eta_{ij}(t)$, that can be evaluated analyzing the *CTMC* in isolation through any of the available techniques for their analysis (Bolch et al., 2006). Finally, let Δ be the branching probabilities matrix, which entries Δ_{ij} describe the probability that an immediate transition will bring from state i to state j.

The local kernel and global kernel can be derived according to Equation 3.5 and Equation 3.6, respectively.

$$L_{ij}(t) = \begin{cases} e^{q_{ii}*t} & i \in \mathcal{S}^E \wedge j = i \quad (1a) \\ (1 - F_g(t))\eta_{ij}(t) & i \in \mathcal{S}^g \wedge j \in \mathcal{S}(i) \quad (1b) \\ 0 & \quad otherwise \quad (1c) \end{cases} \tag{3.5}$$

$$G_{ij}(t) = \begin{cases} \frac{q_{ij}}{-q_{ii}}(1 - e^{q_{ii}} * t) & i \in \mathcal{S}^E \wedge j \neq i \wedge q_{ii} < 0 \quad (2a) \\ (1 - F_g(t))\eta_{ij}(t) + \int_0^t \eta_{ij}(x) f_g(x) dx & i \in \mathcal{S}^g \wedge j \in \mathcal{S}_\epsilon(i) \wedge j \notin \mathcal{S}_g(i) \quad (2b) \\ \sum\limits_{m \in \mathcal{S}(i)} \Delta_{mj} \int_0^t \eta_{im}(x) * f_g(x) dx & i \in \mathcal{S}^g \wedge j \notin \mathcal{S}_\epsilon(i) \wedge j \in \mathcal{S}_g(i) \quad (2c) \\ (2b) + (2c) & i \in \mathcal{S}^g \wedge j \in \mathcal{S}_\epsilon(i) \wedge j \in \mathcal{S}_g(i) \quad (2d) \\ 0 & otherwise \quad (2e) \end{cases} \tag{3.6}$$

Specifically equation *(1a)* of the local kernel gives the probability that if in the current marking there are only EXP transitions enabled (since $i \in \mathcal{S}^E$), no transitions have fired yet at time t and thus the marking didn't change ($j = i$) and the model didn't regenerate yet. Since it behaves as a CTMC, this probability is given by $e^{q_{ii}*t}$. Equation *(1b)* considers the case that a GEN transition has been enabled in the initial state i ($i \in \mathcal{S}^g$) and in the current marking j the system has not regenerated yet $j \in \mathcal{S}(i)$, thus transition g has not fired. The probability that transition g is not yet fired at time t is given by $(1 - F_g(t))$, where $F_g(t)$ is the CDF of time to fire of transition g, while the probability that the subordinated *CTMC* is in j is given by the transient probabilities $\eta_{ij}(t)$. In all the other cases, the model regenerates. In particular equation *(2a)* considers the case that in the initial state i all transitions are EXP and one of them executes causing the system to regenerate. Equations *(2b), (2c) and (2d)* consider the case that a GEN transition g was enabled in the initial state i. In equation *(2b)* the model reaches state j, where g has been disabled due to the firing of one of the EXP transitions. In equation *(2c)* the model reaches state j due to the firing of transition g. Finally, in equation *(2d)*, the case of a state j that can be reached both due to the firing of an EXP that disables g or due to the firing of g is considered.

Thus following this approach kernels can be evaluated and transient probabilities of the model can then be evaluated through the Markov renewal equations 3.4.

3.3.2 Kernels evaluation under bounded regeneration restriction

An alternative technique for kernel evaluations is the *regenerative transient analysis with stochastic state classes* (Horváth et al., 2012). This technique can be applied when the *bounded regeneration restriction* is satisfied.

Definition 7. *A model is under bounded regeneration restriction if always a regeneration is eventually reached within a bounded number of discrete events.*

It is worth noticing that there are models in which both the *bounded regeneration restriction* and the *enabling restriction* are satisfied, models in which only one is satisfied, and models in which neither of the two is satisfied. According to this, depending on the model, the correct analysis method needs to be chosen accordingly.

The regenerative transient analysis under bounded regeneration restriction is based on the concept of *Stochastic state classes*, which samples the state of the *STPN* immediately after a firing. In particular, *Stochastic state classes* extend the concept of *state classes* introduced in Chapter 2, in a twofold manner. First, adding a timer called *age*, that accounts for the time elapsed since the initial regeneration. Moreover, it adds a joint probability density function for the age and the remaining times to fire of each enabled transition.

Definition 8. *A stochastic state class is a tuple $\Sigma = \langle m, D, f \rangle$ where: m is the marking; D is the support of the random vector $\langle \bar{\tau}, \tau_{age} \rangle$, where τ_{age} is the absolute time since the initial regeneration and $\bar{\tau}$ is the vector of the remaining times to fire of enabled transitions; finally f is the PDF of $\langle \bar{\tau}, \tau_{age} \rangle$, which we denote as state density function.*

Starting from an initial stochastic state class with $\tau_{age} = 0$ and independently distributed times to fire for the enabled transitions, enumeration of a reachability relation among stochastic state classes yields a *stochastic transient tree*, where the support of the vector $\bar{\tau}$ in each class is represented as shown in Chapter 2, i.e. a linear convex polyhedron that represents the solution of a set of linear inequalities constraining the difference between pairs of remaining times to fire. The following definition formalizes the concept of successors.

Definition 9. *A stochastic state class $\Sigma' = \langle m', D', f' \rangle$ is the successor of a stochastic state class $\Sigma = \langle m, D, f \rangle$ through firing a transition t with probability μ, which we write $\Sigma \overset{t,\mu}{\Rightarrow}$*

Σ', iff, given that the marking is m and the random vector $\langle \tau_{\text{age}}, \bar{\tau} \rangle$ is distributed over D according to f, t fires with probability μ, yielding a marking m' and a random vector $\langle \bar{\tau}', \tau'_{age} \rangle$ distributed over D' according to f'.

A stochastic state class is said to be *regenerative* if the Markov property is satisfied immediately after the class is entered, which occurs if and only if all active GEN times to fires have been enabled for a deterministic time (Paolieri et al., 2016).

Definition 10. *A stochastic state class Σ is called regenerative if the time elapsed from the enabling of each enabled GEN transition t_i until the firing that led to Σ is a deterministic value $d_i \in \mathbb{R}_{\geqslant 0}$, called the enabling time of t_i in Σ.*

A sub case of this definition, is that a stochastic state class is a regeneration in the case in which all GEN times to fires are newly enabled, thus their deterministic enabling time is $d_i = 0 \; \forall t_i$

The analysis technique enumerates stochastic state classes from each regeneration until any regeneration is reached, yielding a set of *stochastic transient trees* that are rooted in a regenerative stochastic state class and contain non-regenerative successors reached before any regeneration. Note that according to this, all leaf nodes represent regenerations. Under the bounded regeneration restriction, each tree is finite and collects all stochastic state classes that capture the behavior during a regenerative epoch of the *MRGP*. According to this, each tree allow to derive global and local kernel of a regenerative epoch of the *MRGP* and thus to evaluate transient probabilities of the model through the Markov Renewal Equations formulated in equation 3.4.

The calculus of successors computes:

- the probability μ that t is the transition that fires in Σ
- the successor class $\Sigma' = \langle m', D', f' \rangle$, given that t fires, which includes the marking m' and the joint PDF of remaining times to fire f' of transitions enabled in m' and the variable τ_{age} which is decreased by the sojourn time. Note that in so doing τ_{age} is negative instead of positive, because this allows one to treat it similarly to remaining times to fire (Horváth et al., 2012). According to this, the time of the last firing is given by $-\tau_{age}$

For additional details regarding the derivation of the joint PDF of successor classes and the evaluation of μ, see (Horváth et al., 2012). When stochastic transient trees enumeration is completed, kernels are evaluated. Let R be the set of regenerative states and M the set of reachable markings. Given a stochastic transient tree enumerated from regeneration $i \in R$, let INNER(i) be the set of stochastic state classes that are inner nodes and LEAVES(i) be the set of stochastic state classes that are leafs. Kernels can be derived from equations 3.7 and 3.8, for all $i, k \in R$, $j \in M$ and $t \geqslant 0$.

$$L_{ij}(t) = \sum_{\substack{\Sigma \in \text{INNER}(i) \text{ s.t.} \\ \Sigma \text{ has marking } j}} p_{in}(\Sigma, t) \tag{3.7}$$

$$G_{ik}(t) = \sum_{\substack{\Sigma \in \text{LEAVES}(i) \text{ s.t.} \\ \Sigma \text{ has regeneration } k}} p_{reach}(\Sigma, t) \tag{3.8}$$

For a class $\Sigma = \langle m, D, f \rangle$ that can be reached through a sequence of k firings, each one having probability μ_i to occur, the probability to be reached is given by equation 3.9.

$$\rho_\Sigma = \prod_{i=0}^{k-1} \mu_i \tag{3.9}$$

Which in turn allows one to derive the probability to reach Σ from i within time t, with equation 3.10, and the probability that Σ has been reached and not yet left at time t, with equation 3.11.

$$p_{reach}(\Sigma, t) = \rho_\Sigma \int_{\{\langle \tau_{age}, \bar{\tau} \rangle \in D | \tau_{age} \geqslant t)\}} f(\tau_{age}, \bar{\tau}) d\tau_{age} d\bar{\tau} \tag{3.10}$$

$$p_{in}(\Sigma, t) = \rho_\Sigma \int_{\{\langle \tau_{age}, \bar{\tau} \rangle \in D | \tau_{age} \geqslant t \cap min(\bar{\tau}) > t + \tau_{age})\}} f(\tau_{age}, \bar{\tau}) d\tau_{age} d\bar{\tau} \tag{3.11}$$

3.3.3 Approximated kernels evaluation

Analysis with stochastic state classes requires that the *bounded regeneration restriction* is satisfied for each regenerative epoch of the process. If this is not true even only for a single epoch, the analysis does not terminate because the number of nodes to be enumerated is not bounded. Termination can be guaranteed in probability, if an approximation is introduced for such regenerative epochs. The approximate analysis is based on a partial characterization of the regeneration epoch which can be derived by a partial enumeration of nodes of the not bounded regenerative epoch. First, an approximation threshold ϵ is chosen. Then the enumeration of successors is performed giving priority to nodes that have an higher probability to be reached within considered time limit t. When the evaluated stochastic transient tree has a total mass of probability greater than $1 - \epsilon$, enumeration is interrupted. According to this, approximated local kernels $\tilde{L}_{ij}(t)$ and global kernels $\tilde{G}_{ij}(t)$ can be evaluated from the approximated stochastic transient tree. Equations 3.12 and 3.13, show the relationship between exact kernel and local kernels.

$$\tilde{L}_{ij}(t) \leqslant L_{ij}(t) \tag{3.12}$$

$$\tilde{G}_{ij}(t) \leqslant G_{ij}(t) \tag{3.13}$$

It is worth noticing that the approximation can be applied also to reduce complexity of the analysis of models that satisfy the bounded restriction in order to speed up the evaluation.

The problem of this approach is that in general less defective kernels result in more precise approximated transient probabilities $\tilde{\pi}_{ij}(t)$ and this error grows with time:

$$\lim_{t\to\infty} \tilde{\pi}_{ij}(t) = 0 \tag{3.14}$$

3.4 Regenerative steady state analysis

From the local and global kernels, steady-state probabilities of an MRGP can be derived as follows (Martina et al., 2016; Logothetis et al., 1995; Kulkarni, 2016):

$$\lim_{t\to\infty} P\{M(t) = j\} = \frac{\sum_{i\in R} \pi_i \alpha_{ij}}{\sum_{i\in R, m\in M} \pi_i \alpha_{im}} \tag{3.15}$$

Where R is the set of regenerations of the model, M is the set of states, $\alpha_{ij} := \int_0^\infty L_{ij}(t)dt$ is the expected sojourn time spent in state j, after the regeneration i, π_i is the steady state solution of the *DTMC* embedded at regeneration points. Specifically, steady state probabilities on the *DTMC* embedded at regeneration points, can be evaluated as $\vec{\pi} = \vec{\pi}\mathbb{G}$ and $\sum_{i\in R} \pi_i = 1$ where $\mathbb{G} := \lim_{t\to\infty} \mathbb{G}(t)$. Note that, when evaluating the kernels with the stochastic state classes method, probabilities are given per class, thus it is necessary to first aggregate them by marking, and then apply Equation 3.15. Finally, note that the approximation technique described in Section 3.3.3 can't be used for the evaluation of steady state probabilities since error grows with time.

Chapter 4

Integration of transient solution techniques

As shown in Chapter 3, previous work on the evaluation of kernels of an *MRGP* requires that all the regenerative epochs satisfy the *bounded regeneration restriction*, or that all regenerative epochs satisfy the *enabling restriction*. Alternatively, also the method of supplementary variables (German and Lindemann, 1994; Telek and Horváth, 2001) might in principle encompass the case of multiple concurrently enabled GEN timers, but practical feasibility restrains applicability under the enabling restriction. Moreover sampling at equidistant time points (Zimmermann, 2012; Lindemann and Thümmler, 1999), permits evaluation for models where all timers have either DET or EXP durations. Finally, for models that break both the enabling and the bounded regeneration restriction, kernel components may be still defectively approximated by truncation of stochastic transient trees (Horváth et al., 2012), but with the problem that the introduced error increases with time (see Section 3.3.3). Additionally it is worth noticing that this approximation technique approximates all regenerative epochs with the same threshold ϵ, without considering that different regenerative epochs can have larger or smaller impact on the error committed on the transient probabilities, depending on how often and for how long they are visited.

In this chapter, we present a novel technique (Biagi et al., 2017) that exploits the non-deterministic analysis to automatically drive the integration of different solution techniques, that are applicable to different regenerative epochs. To this end, we characterize the structure of the state space through terminating and efficient non-deterministic analysis, identifying regenerative epochs and solution techniques that can be applied for kernel components corresponding to each regeneration (Section 4.1). This permits integration of the consolidated technique of enabling restriction with exact and approximate solutions based on stochastic state classes (Section 4.2). Moreover, we also introduce a novel technique that iteratively adapts the approximation of each kernel component so as to optimize the impact of truncation on the defect in the evaluation of transient probabilities (Sections 4.3 and 4.4) (Biagi et al., 2017). The approach permits to accurately evaluate kernels, and it is open to further adaptation strategies and to integration of other solution techniques, both numerical and simulative.

4.1 Identification and classification of regenerative epochs

Let's consider a model expressed in the STPN formalism. Given an initial marking m_0 and an initial PDF $f_{\bar{\tau}_0}$ for the vector $\bar{\tau}$ of the times-to-fire of the enabled transitions, the STPN semantics induces a probability space $\langle \Omega_{m_0}, \mathbb{F}_{\bar{\tau}_0}, \mathbb{P}_{m_0, f_{\bar{\tau}_0}} \rangle$, where Ω_{m_0} is the set of outcomes (i.e., feasible timed firing sequences of the model), $\mathbb{F}_{\bar{\tau}_0}$ is a *σ-algebra* on the outcomes and $\mathbb{P}_{m_0, f_{\bar{\tau}_0}}$ is a probability measure over the events (Paolieri et al., 2016). Note that $\mathbb{P}_{m_0, f_{\bar{\tau}_0}}$ is zero for outcomes that are not feasible under $f_{\bar{\tau}_0}$.

The set of states collected in a stochastic state class identifies a unique underlying non-deterministic state class (Vicario et al., 2009) that represents the marking and the support of the vector of the remaining times-to-fire of the enabled transitions when the class is entered. The association between non-deterministic and stochastic state classes is one-to-many (possibly one-to-infinite) and preserves qualitative properties referred to the set of feasible outcomes Ω_{m_0}, while abstracting from quantitative properties depending on the probability measure $\mathcal{P}_{m_0, f_{\bar{\tau}_0}}$.

Given that a stochastic state class is regenerative if it satisfies Definition 10 (see Section 3.3.2), which depends on Ω_{m_0} but not on $\mathcal{P}_{m_0, f_{\bar{\tau}_0}}$, state classes can be used to identify regenerations.

To this end, the state space of the underlying TPN is covered by a set of SCGs, which we call *First-Epoch State Class Graphs* (FESCGs), each rooted in a regenerative state class and containing all non-regenerative successors reached before any regeneration (which is also included in the graph). In other words, instead that using non-deterministic analysis to build a single SCG as shown in Chapter 2, we split it in a set of FESCGs, each one representing a regenerative epoch. Moreover, enumeration of FESCGs can suppress successor relations that correspond to null probability events, i.e., firings that in any associated stochastic state class would be possible in a null measure subset of the support. The following lemma can be used to identify which successor relations can be suppressed.

Lemma 1. *Let u be an STPN, v be its underlying TPN, R be the set of successor relations* $\Sigma = \langle m, D, f \rangle \xrightarrow{t,\mu} \Sigma' = \langle m', D', f' \rangle$ *in the stochastic transient tree of u enumerated from a regenerative stochastic state class* $\Sigma_0 = \langle m_0, D_0 \rangle$*, and* $S = \langle m, \bar{D} \rangle \xrightarrow{t} S' = \langle m', \bar{D}' \rangle$ *be a succession relation in the SCG of v enumerated from a regenerative state class* $S_0 =$

$\langle m_0, \bar{D_0} \rangle$, such that $\bar{D}$, $\bar{D}'$, and $\bar{D_0}$ are the projections of D, D', and D_0 that eliminate τ_{age}, respectively. Information regarding the probability μ can be derived according to the following ordered criteria:

1. *if t is IMM, $\mu > 0$*
2. *if t is not IMM and there is at least another enabled IMM transition in m, $\mu = 0$*
3. *if t is DET, $\mu = 0$ iff one of the following sub-statements is true:*
    a) *there is at least another enabled DET transition with a smaller time to fire*
    b) *there is at least one timer that is not DET, and D conditioned to be strictly greater than $\bar{\tau}_t$ and eliminating all DET timers (including t) has null measures in $\mathbb{R}^N$, where N is the number of distributed times-to-fire in Σ and S*
4. *if t is neither DET nor IMM, $\mu > 0$ iff both following sub-statements are true:*
    a) *let $\bar{\tau}(t_d)$ be the smaller time to fire among enabled concurrent DET transitions. D conditioned to be strictly smaller than $\bar{\tau}(t_d)$ and eliminating all timers except t has non-null measures in $\mathbb{R}^N$*
    b) *the projection of D that eliminates DET timers, conditioned to the firing of transition t, has a non-null measure in $\mathbb{R}^N$*

Proof. Statement 1 considers the case of *t* being an IMM transition. In that case, according to the semantics of STPN, its time to fire is $\bar{\tau}(t) = 0$ and thus it is equal or smaller the time to fire of any other concurrently enabled transition. If *t* it is the only IMM enabled transition, $\mu > 0$, because DET transitions have times to fire greater than zero, and transitions neither IMM nor DET, have probability 0 to sample a value that is exactly 0. For concurrently enabled IMM transitions, in any case its probability to fire is greater than 0, since it is evaluated according to Equation 3.1 and all weights are positive according to Definition 4. Following the same reasoning, *statement 2* exploits the fact the IMM transitions fire first. According to this, all following statements do not consider the case of IMM transitions since in such case is always easy to prove which transitions can fire or not.

If instead *t* is a DET transition, there are two cases in which it has probability to fire $\mu = 0$. The first case of *statement 3a* is when another transition is DET and its time to fire is smaller: from the semantics of STPN (Section 3.1.2) such transition will fire first and thus $\mu = 0$. On the contrary, if a concurrently enabled DET transition is enabled with the same or greater time to fire it does not prevent transition *t* to fire and thus $\mu > 0$. The other case of *statement 3b* is when there are concurrently enabled transitions that are not DET. Let D_t be D conditioned to be strictly greater of *t*, i.e., $D_t = D \cap \{\tau_t \leqslant \tau_{t_i} \forall\, t_i \in T_E(m)\}$, where τ_t is the time-to-fire of *t* and $T_E(m)$ is the set of transitions enabled by m. Let $\hat{D}_t$ be the projection of D_t that eliminates DET timers. (If) If $\hat{D}_t$ has null measure in $\mathbb{R}^N$, either *i*) the STPN includes some transition associated with a mixed distribution, or *ii*) $\mu = 0$, because it means that one of the concurrently enabled transition must fire before t. By Definition 4 (Section 3.1.1), the CDF of each GEN transition is absolutely continuous over its support, thus $\mu = 0$. (Only if) If, ab absurdo, $\hat{D}_t$ had non-null measure in $\mathbb{R}^N$, then the integral over $\hat{D}_t$ of the marginal distribution of distributed times-to-fire in Σ conditioned to the firing of t would not be zero, yielding $\mu \neq 0$.

Finally, if t is neither DET nor IMM, to verify if the transition can fire, it is necessary to verify if any of the concurrently enabled transition prevent it to fire. Specifically, for *statement 4a*, let $\tilde{D}_t^{t_d}$ be the domain conditioned to be strictly smaller than $\bar{\tau}(t_d)$ and where all transitions except t and t_d have been eliminated. (If) If $\tilde{D}_t^{t_d}$ has non-null measure in $\mathbb{R}^N$, $\mu > 0$, because it means that t can fires before t_d, thus t_d is not preventing the firing of the transition. (Only if) If, ab absurdo, $\tilde{D}_t^{t_d}$ had null measure in $\mathbb{R}^N$, $\mu = 0$, because it means that t can't fire before t_d. Note also that it is sufficient to consider the DET transition with the smaller time to fire among enabled DET transitions, because it is the worst case since other DET transitions are obliged to fire later. In conclusion, for all other transitions *statement 4b* can be proven in the same way as done for *statement 3b*. □

It is worth noticing that enumerated criteria need to be applied following specified order, thus criteria from 3 onwards assume that no IMM transitions are concurrently enabled. It is straightforward to show that a regenerative epoch complies with the enabling restriction iff at most one GEN transition is enabled in each state class of its FESCG. Conversely, compliance with the bounded regeneration restriction depends on the presence of cycles in the FESCG.

Lemma 2. *A regenerative epoch complies with the bounded regeneration restriction iff its FESCG does not include any cycle.*

Proof. (If) If, ab absurdo, a regenerative epoch did not satisfy the bounded regeneration restriction, the STPN would allow a timed firing sequence made of an unbounded number of firings that never visits a regeneration; given that an STPN and its underlying TPN have the same set of timed firing sequences Ω_{m_0}, also the TPN would allow that behavior. Given that each state class is associated with one or more stochastic state classes having the same marking and time domain, there would exist a state class associated with an unbounded number of stochastic state classes. As a consequence, the FESCG would include a cycle.

(Only if) If, ab absurdo, the FESCG of a regeneration included a cycle, then, by construction, that cycle would not visit any regenerative state class. Hence, there would exist a timed firing sequence that would allow an unbounded number of firings without visiting a regeneration point, and the corresponding regenerative epoch would not comply with the bounded regeneration restriction. □

For instance, Figure 4.1 shows an STPN and its underlying TPN. The SCG of the model, shown in the lower part of Figure 4.1, can hence be derived. The SCG consists of 5 regenerative and 5 non-regenerative state classes. In particular: the FESCG rooted in S_3 includes S_6 and S_1, satisfying the bounded regeneration restriction (it is cycle free) but not the enabling restriction (two GEN transitions are enabled in S_3); the FESCG rooted in S_5 includes S_8, S_{10}, and S_1, complying with the enabling restriction but not with the bounded regeneration restriction (due to the cycle S_8–S_{10}); and, the FESCG rooted in S_4 includes S_7, S_9, and S_1, satisfying neither the bounded regeneration restriction (due to the cycle S_7–S_9) nor the enabling restriction (two GEN transitions are enabled in S_4, S_7, and S_9). Note that the firing of transition `gen1` in state class S_3 would have probability zero in any associated stochastic state class and thus it is suppressed. That's because enabled transitions in the marking are $T_E([G1, G2]) = \{"gen1", "gen2"\}$, the support of `gen1` is $[2, 4]$ and it is concurrently enabled with `gen2` that has support $[1, 2]$, according to this the probability that `gen1` fires before `gen2` is 0.

4.2 An algorithm for transient analysis of MRGP

Given an STPN with an underlying MRGP, the kernel entries of each regenerative epoch can be derived through a different solution technique depending on whether the epoch satisfies the bounded regeneration restriction, or the enabling restriction, or neither of the two conditions. The applicable solution strategy can be efficiently selected through non-deterministic analysis of the underlying TPN of the model, by enumerating the SCG so as to identify the set Θ of regenerative state classes, the set Ψ of reachable markings, and the FESCG of each regenerative state class $i \in \Theta$:

- if the FESCG of i complies with the bounded regeneration restriction (e.g., the FESCG rooted in S_3 in the SCG in Figure 4.1), $L_{ij}(t)$ and $g_{ik}(t)$ are computed through the exact regenerative transient analysis using stochastic state classes (see Section 3.3.2), for any marking $j \in \Psi$, for any regenerative state class $k \in \Theta$, and for any time point t;

- if the FESCG of i satisfies the enabling restriction (e.g., the FESCG rooted in S_5 in the SCG in Figure 4.1), $L_{ij}(t)$ and $g_{ik}(t)$ are derived through the analysis under enabling restriction (see Section 3.3.1);

- if the FESCG of i breaks both the enabling and the bounded regeneration restrictions (e.g., the FESCG rooted in S_4 in the SCG in Figure 4.1), $L_{ij}(t)$ and

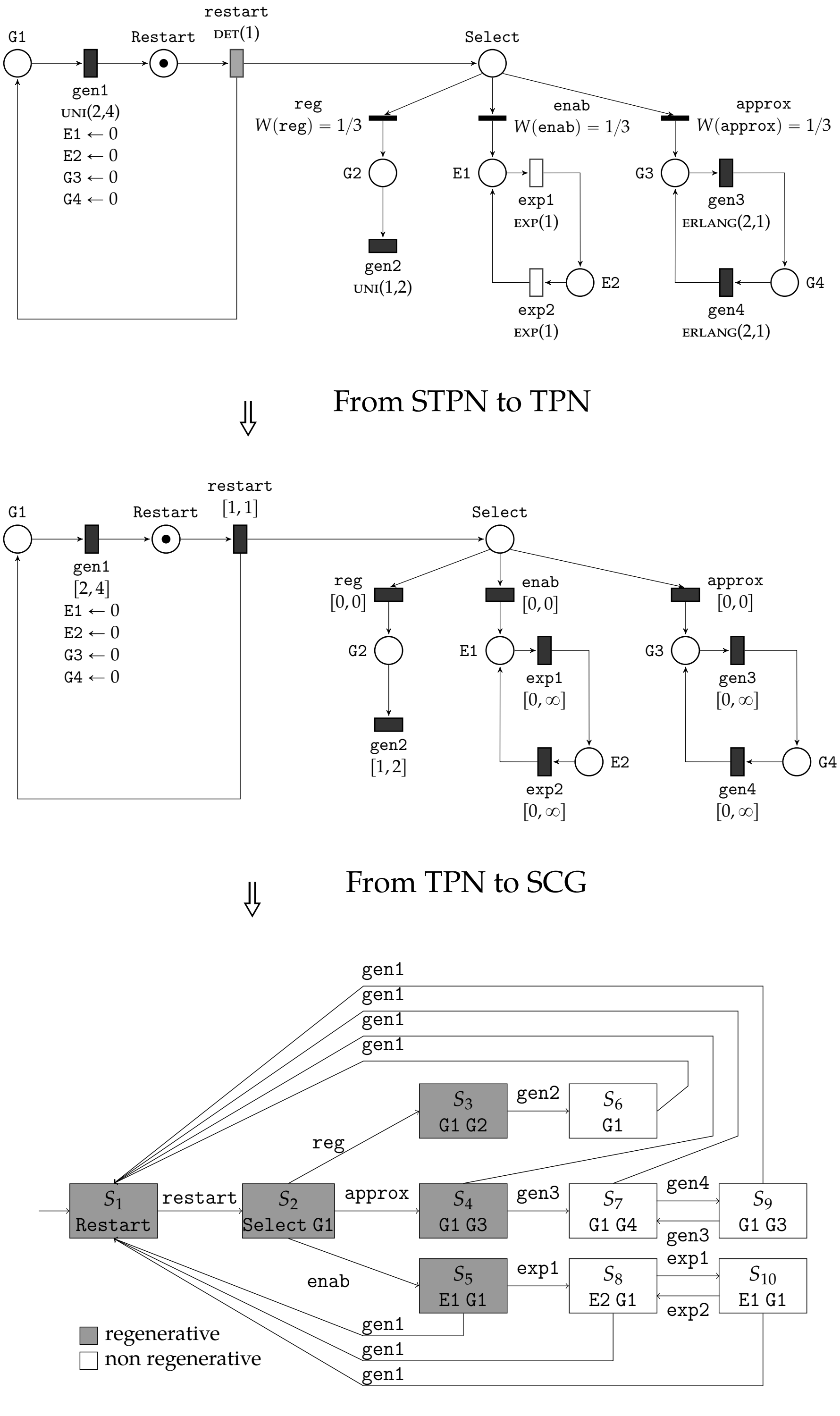


Figure 4.1: Evaluation of the SCG of a STPN. First the TPN is derived preserving the supports but abstracting weights of immediate transitions and PDFs of timed transitions. Then the SCG composed by a set of FESCGs is evaluated and regenerations identified

$g_{ik}(t)$ can still be estimated by stochastic simulation of the STPN model or they can be approximated by numerical solution as developed in Section 3.3.3.

Note that in doing so the derivation of kernel entries always terminates (even for models with an underlying marking process beyond the class of MRGPs), provided that the FESCG of each regenerative state class is finite, which in turn is guaranteed under the fairly general conditions mentioned at the end of Section 2.2. Also note that, in the present implementation, kernels of regenerative epochs that satisfy both the restrictions are derived through the method of stochastic state classes, but analysis under the enabling restriction could be applied as well; moreover, approximated analysis or simulation might be applied also to regenerative epochs that satisfy one or both the restrictions, as a way to reduce complexity of solution. In-depth comparison and experimentation of the impact of different choices on accuracy and complexity deserves further study.

When kernel entries have been evaluated, transient probabilities of reachable markings are finally derived by numerical integration of the Markov renewal equations of Equation 3.4.

4.3 Approximate evaluation of the kernels of an MRGP

In general, and in particular for regenerative epochs that do not satisfy either the bounded regeneration or the enabling restrictions, an approximation of kernel entries can be derived by truncating the enumeration of the stochastic transient tree computed in the exact regenerative transient analysis (Horváth et al., 2012). In this case, following the steps of Sect. 3.3.3, the approximated kernel entries $\tilde{L}_{hj}(t)$ and $\tilde{g}_{ik}(x)$ are computed on a subset of the classes in the stochastic transient tree of the regenerative state class i, and they thus comprise an under-approximation of the exact values $L_{hj}(t)$ and $g_{ik}(x)$. Specifically, denoting $\Delta_{ij}(t) := L_{hj}(t) - \tilde{L}_{hj}(t)$ and $\delta_{ik}(t) := g_{ik}(x) - \tilde{g}_{ik}(x)$, we have $\Delta_{ij}(t) \geqslant 0$ and $\delta_{ik}(t) \geqslant 0$ $\forall t$.

To characterize the impact of the approximation, the following Lemma provides a bound on $\epsilon_{ij}(t) := \pi_{ij}(t) - \tilde{\pi}_{ij}(t)$, with $\tilde{\pi}_{ij}(t)$ denoting the solution of Markov Renewal Equations (Equation 3.4) obtained with approximated kernel entries:

$$\tilde{\pi}_{ij}(t) = \tilde{L}_{ij}(t) + \sum_{k\in\Theta} \int_0^t \tilde{g}_{ik}(x)\, \tilde{\pi}_{kj}(t-x)\, dx \tag{4.1}$$

Lemma 3. *For each regenerative state class $i \in \Theta$, marking $j \in \Psi$, and time t, the error $\epsilon_{ij}(t)$ is non-negative and upper-bounded:*

$$0 \leqslant \epsilon_{ij}(t) \leqslant \phi_i(t) + \sum_{k\in\Theta} \int_0^t \left(\tilde{g}_{ik}(x)\epsilon_{kj}(t-x) + \phi_i(x)(\epsilon_{kj}(t-x) + \tilde{\pi}_{kj}(t-x))\right) dx \tag{4.2}$$

where $\phi_i(t) := \sum_{j\in\Psi}(L_{ij}(t) - \tilde{L}_{ij}(t)) + \sum_{k\in\Theta}(g_{ik}(t) - \tilde{g}_{ik}(t))$.

Proof. By combining Eqs. 3.4 and 4.1, we obtain: $\epsilon_{ij}(t) = \Delta_{ij}(t) + \sum_{k\in\Theta}\int_0^t(\tilde{g}_{ik}(t) + \delta_{ik}(x)) \cdot \epsilon_{kj}(t-x) + \delta_{ik}(x)) \cdot \tilde{\pi}_{kj}(t-x)dx$. Since $\Delta_{ij}(t) \geqslant 0$ and $\delta_{ik}(t) \geqslant 0$, $\phi_i(t) \geqslant \Delta_{ij}(t)$ $\forall\, j \in \Psi$ and $\phi_i(t) \geqslant \delta_{ik}(t)\ \forall\, k \in \Theta$. The upper bound of Eq. 4.2 can thus be obtained by replacing $\Delta_{ij}(t)$ and $\delta_{ik}(t)$ with $\phi_i(t)$.

To prove that $\epsilon_{ij}(t) \geqslant 0$, $\epsilon_{ij}(t)$ is rewritten as $\epsilon_{ij}(t) = A_{ij}(t) + \sum_{k\in\Theta}\int_0^t(\tilde{g}_{ik}(x) \cdot \epsilon_{kj}(t-x)dx$ where $A_{ij}(t) := \Delta_{ij}(t) + \sum_{k\in\Theta}\int_0^t \delta_{ik}(x)\pi_{kj}(t-x)dx$. Note that $A_{ij}(t) \geqslant 0$, being $\Delta_{ij}(t) \geqslant 0$, $\delta_{ik}(x) \geqslant 0$, and being $\pi_{kj}(t-x)$ a probability. For any discretization step $\tau \in \mathbb{R}_{>0}$, the expression of $\epsilon_{ij}(t)$ can be rewritten by replacing $t = M \cdot \tau$ and $x = m \cdot \tau$, with $m \in [0, M]$. By induction on M, it is easily proven that $\epsilon_{ij}(t)$ is monotonic non-decreasing with t. Moreover $\epsilon_{ij}(0) = A_{ij}(0) \geqslant 0$, which proves that $\epsilon_{ij}(t) \geqslant 0$. □

Note that, since $0 \leqslant \tilde{\pi}_{ij}(t)$ for every marking i, j and time t, summation of probabilities over all reachable markings provides a defective (i.e., lower than 1) evaluation of the total probability mass properly allocated; the complement to 1 of this quantity thus comprises a safe upper bound on the maximum value of each computed probability or summation over them.

4.4 Heuristic driven approximation

The quantity $\phi_i(t)$ in Eq. 4.2 can be safely estimated as the sum of probabilities to reach a truncation point in the partial enumeration of the stochastic transient tree of regenerative class i. According to this, the bounds of Eq. 4.2 can be used to define a truncation policy in the partial enumeration of regenerative epochs that break both the enabling and the bounded regeneration restrictions (*unrestricted epochs*) with a twofold aim: adapt the error accumulated on kernel entries of each regeneration i to the impact that this epoch takes on the final error $\epsilon_{ij}(t)$; and drive the selection of truncation points within each stochastic transient tree so as to control the trade-off between complexity of enumeration and accuracy of approximation. However, exact implementation of this policy would require repeated evaluation of approximated probabilities $\tilde{\pi}_{ij}(t)$, which in turn implies a major numerical complexity for the solution of Volterra integral equations. Lemma 3 can thus be more conveniently exploited as a ground for the definition of efficient heuristics driving truncation within each regenerative epoch. Note that, while this work emphasizes the use of approximation as a way to make the evaluation of kernel entries feasible, approximation driven by efficient heuristics may be applied also to reduce complexity in epochs that fit the bounded regeneration or the enabling restrictions.

Partial exploration of unrestricted epochs is performed by initially enumerating at most ν_{start} nodes in each tree, and then by iteratively identifying a non-regenerative leaf node and by enumerating at most ν_{iter} of its successors, until the number of classes enumerated in unrestricted epochs is larger than a threshold ν_{max} (*heuristic-based approximate analysis*). Given that the upper-bound of Eq. 4.2 suggests that the approximation error affects more those regenerative epochs that are visited more often, at each iteration we enumerate the successors of the non-regenerative leaf node with the largest *estimated* probability to be reached. Such estimate is evaluated by analyzing a Discrete Time Markov Chain (DTMC) $\mathbb{D}$ specified as follows:

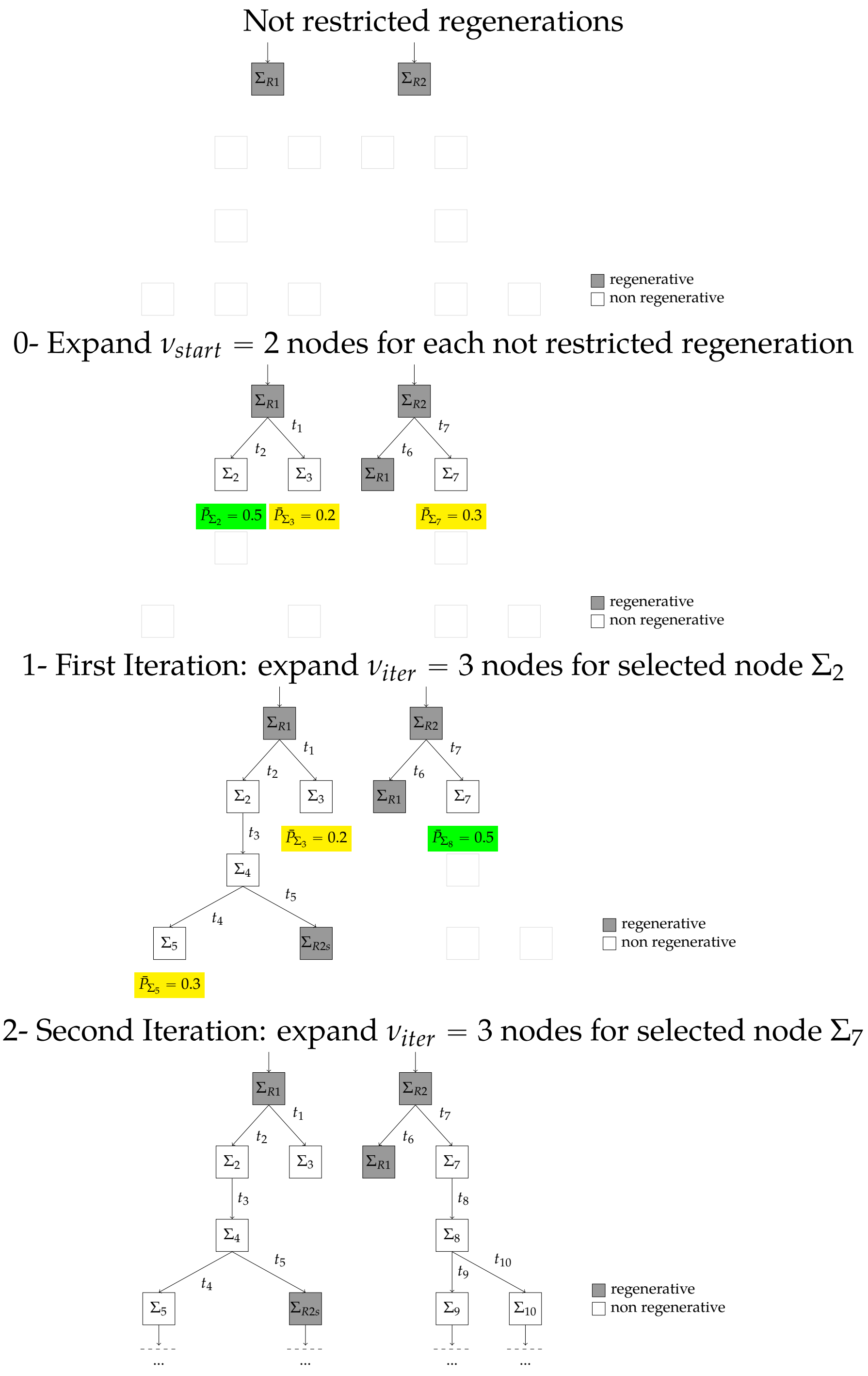
- $\mathbb{D}$ has a state for each regenerative state class $i \in \Theta$ and for each leaf node j (either regenerative or non-regenerative) belonging to any tree $\mathcal{T}_i \in \mathcal{T}$ (regenerative and non-regenerative leaf nodes are absorbing in every tree);
- $\mathbb{D}$ has an arc from each state representing a regenerative state class $i \in \Theta$ to each state representing a leaf node j in $\mathcal{T}_i$, associated with probability μ_{ij};
- if the epoch rooted in i is analyzed exactly, μ_{ij} is equal to $G_{ij}(\infty)$ under the bounded regeneration restriction and to $G_{ij}(t_n)$ under the enabling restriction; otherwise, if the epoch rooted in i is analyzed in approximate manner, μ_{ij} is equal to $\tilde{G}_{ij}(\infty)$ or to $\tilde{L}_{ij}(\infty)$ depending on whether j corresponds to a regenerative or non-regenerative stochastic state class, respectively.

Steady-state analysis of $\mathbb{D}$ yields the vector of state probabilities P. Then, the steady-state probability of the states that correspond to non-regenerative leaf nodes are normalized, obtaining the vector of state probabilities $\bar{P}$, i.e., for each state l of the DTMC $\mathbb{D}$ that corresponds to a non-regenerative leaf node in a tree $\mathcal{T}_i \in \mathcal{T}$, $\bar{P}_l = P_l / \sum_{h \in \mathcal{S}_L} P_h$, where $\mathcal{S}_L$ is the set of states that correspond to non-regenerative leaf nodes in any tree $\mathcal{T}_i \in \mathcal{T}$. Finally, the non-regenerative leaf node that corresponds to the state w with the largest probability $\bar{P}_w$ is selected as the node to be expanded.

It's worth adding that other heuristic criteria could be used as well to select the next node to visit in partial enumeration of stochastic state classes, possibly taking into account an estimate of the mean time until when a regeneration is reached.

4.4.1 Example of the iterative algorithm

An example of execution of the iterative algorithm for the truncated enumeration of stochastic transient trees is reported in Figure 4.2, assuming $\nu_{start} = 2$, $\nu_{iter} = 3$, $\nu_{max} = 10$ as parameters. Suppose we have two not restricted regenerations called $R1$ and R_2. Initially the algorithm starts to evaluate ν_{start} successors of nodes Σ_{R1} and Σ_{R2}, that are the stochastic state classes representing $R1$ and $R2$, respectively. Enumerating $\nu_{start} = 2$ successors, classes Σ_2, Σ_3, Σ_{R1} and Σ_7 are found. Specifically, Σ_{R1} is a regenerative class that thus need not to be expanded anymore, while Σ_2, Σ_3 and Σ_7 are not yet expanded, thus the heuristic needs to decide which is the best


Not restricted regenerations

0- Expand $\nu_{start} = 2$ nodes for each not restricted regeneration

1- First Iteration: expand $\nu_{iter} = 3$ nodes for selected node Σ_2

2- Second Iteration: expand $\nu_{iter} = 3$ nodes for selected node Σ_7

Figure 4.2: Example of transient stochastic classes trees expansion through iterative algorithm, assuming to have $\nu_{start} = 2$, $\nu_{iter} = 3$, $\nu_{max} = 10$

candidate to be explored. Since only $4 < \nu_{max}$ classes have been enumerated for not restricted epochs, another iteration need to be performed. Kernels are evaluated, the DTMC $\mathbb{D}$ is built and its steady state probabilities P are derived. P is than normalized considering only Σ_2, Σ_3 and Σ_7, which are the candidates to be expanded, obtaining $\bar{P}_{\Sigma_2} = 0.5$, $\bar{P}_{\Sigma_3} = 0.2$ and $\bar{P}_{\Sigma_7} = 0.3$. According to this Σ_2 is selected and 3 successors expanded since $\nu_{iter} = 3$. Other 3 classes are found, two not regenerative and one regenerative, where Σ_5 has successors not yet expanded. The number of expanded classes is $7 < \nu_{max}$, thus another iteration is required. Evaluated normalized steady state probabilities of the DTMC are $\bar{P}_{\Sigma_3} = 0.2$, $\bar{P}_{\Sigma_5} = 0.3$, and $\bar{P}_{\Sigma_8} = 0.5$, thus candidate for next iteration is Σ_8. Finally, since the number of enumerated classes is $10 \geqslant \nu_{max}$, the iterative algorithm is interrupted and kernels are evaluated on these partial enumerated transient stochastic trees.

Note that when expanding the successors of a regeneration, that is the root node of a stochastic transient tree, ν_{start} nodes are explored, while when a non-regenerative node is selected ν_{iter} nodes are explored. This difference allows one to configure heuristics that expand more nodes when dealing with a new tree, compared to inner nodes of an already partially expanded tree.

Moreover when expanding successors of a node, the expansion is done giving higher priority to nodes that have higher probability to be reached, considering only the current regenerative epoch, as done in (Horváth et al., 2012). It is worth to notice that at higher level, the heuristics selects next successor to be expanded based on the probability that such node will be visited considering the overall process, not only the single current regenerative epoch.

4.5 Experimental evaluation

The approach was implemented on top of the Sirio API of the ORIS Tool (Bucci et al., 2010). Due to the small state space, with a single epoch requiring approximation of kernel entries, the STPN of Figure 4.1 does not illustrate the full potential of the approach. Hence, experiments were performed on the STPN of Fig. 4.3, a variant of a 3-station exhaustive-service polling system (Ibe and Trivedi, 1990), where service sojourn is bounded by a DET timeout, polling times have a GEN distribution, and service times have an EXP or GEN distribution. For each station $s \in \{1,2,3\}$: place `Waiting`$_\mathtt{s}$ encodes the number of pending service requests; places `AtService`$_\mathtt{s}$ and `Vacant`$_\mathtt{s}$ encode whether the station is being served or not, respectively; and, place `Polling`$_\mathtt{s}$ encodes the state where the server is polling station s. In Fig. 4.3, all stations have no pending requests and the server is polling station 1.

The service at station s begins with the firing of transition `startService`$_\mathtt{s}$, with uniform distribution over $[1,2]$, and it may terminate either when the queue of pending requests (`Waiting`$_\mathtt{s}$) is empty or when `timeout`$_\mathtt{s}$ fires after a DET maximum duration of value 3. During the service interval, place $\texttt{Vacant}_\texttt{s}$ is empty and transition $\texttt{serve}_\texttt{s}$ is enabled, so that any number of requests can be served. Transition $\texttt{arrive}_\texttt{s}$ models the arrival of a new request as an EXP distribution with mean 20. Since the EXP distribution has a null minimum value, the maximum number of requests served during a service interval is limited only by the relation between the timeout value and the minimum duration of each service. Specifically, the number of requests served during a service interval sojourn is unbounded for stations 1 and 3, and it is bounded to 3 for station 2 where each service requires at least 1 time unit.

The underlying marking process regenerates whenever the server arrives to any station (i.e., at firing of $\texttt{empty}_\texttt{s}$ or $\texttt{startService}_\texttt{s}$) or leaves it (i.e., at firing of $\texttt{empty}_\texttt{s}$ or $\texttt{timeout}_\texttt{s}$), which directly implies that starting from any reachable state, w.p.1, a regeneration will be eventually reached, i.e. the process is an MRGP. The process behavior falls in different subclasses of MRGP during service sojourns at different stations. When the server is at station 1: the process satisfies the *enabling restriction*, given that $\texttt{timeout}_1$ is the only non-EXP transition enabled in each state; but it does not satisfy the bounded regeneration restriction, as for any natural number n, there exists a non-null probability that $\texttt{serve}_1$ and $\texttt{arrive}_1$ are fired more than n times before the expiration of $\texttt{timeout}_1$. When the server is at station 2: the process satisfies the *bounded regeneration restriction*, given that $\texttt{serve}_2$ cannot be fired more than 3 times before the firing of $\texttt{timeout}_2$; but the enabling restriction is not satisfied as $\texttt{timeout}_2$ and $\texttt{serve}_2$ can be concurrently enabled. When the server is at station 3: the process falls in the *unrestricted case* as $\texttt{timeout}_3$ and $\texttt{serve}_3$ are concurrently enabled, and $\texttt{serve}_3$ may fire an unbounded number of times before the

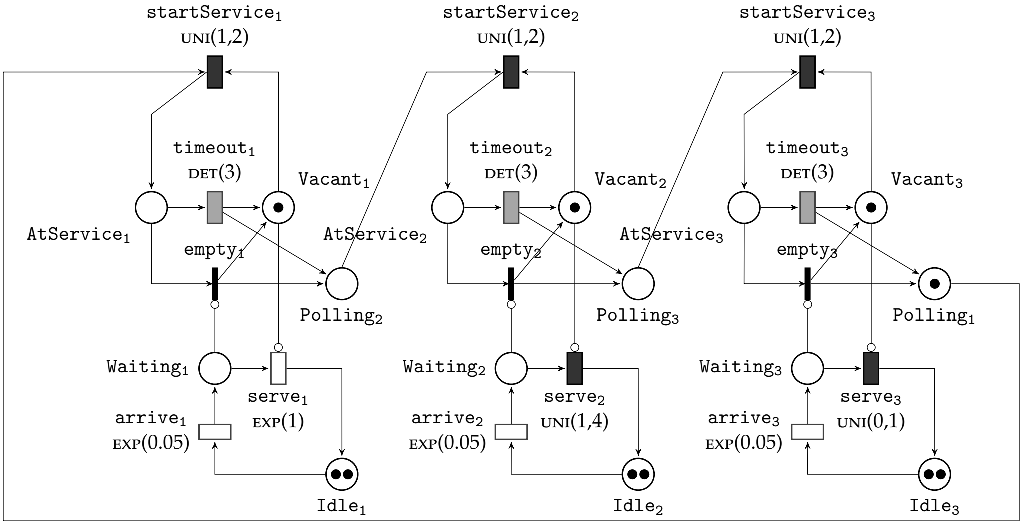


Figure 4.3: STPN of a 3-station exhaustive-service polling system with server timeout.

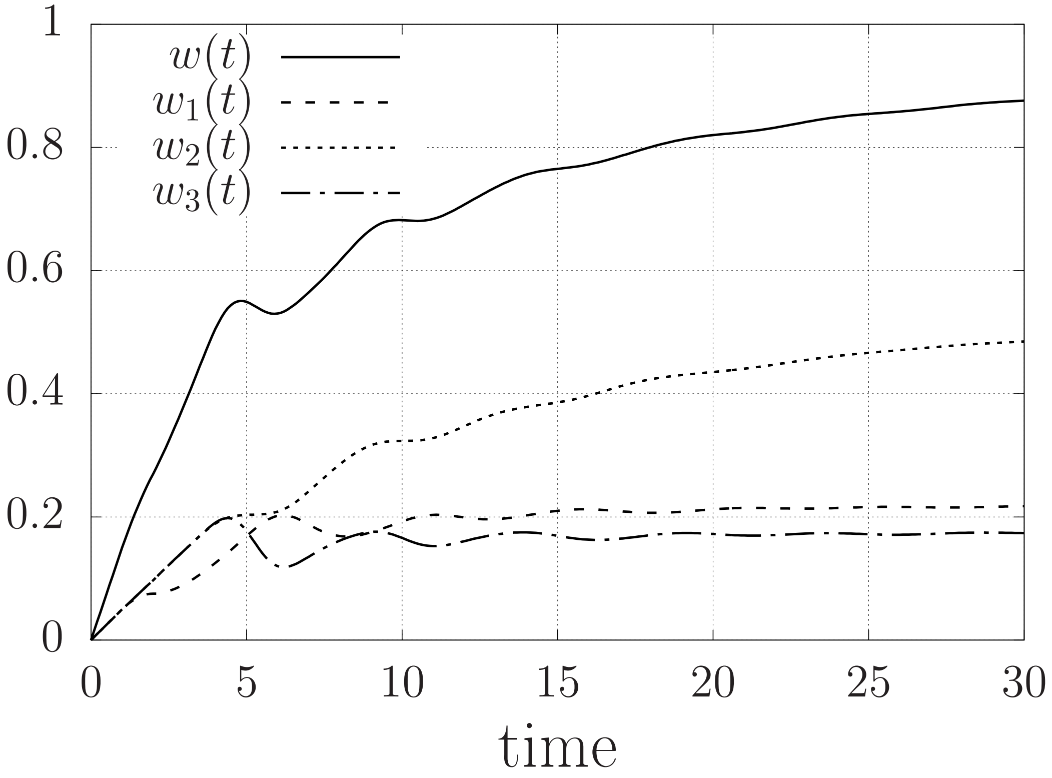


Figure 4.4: Average number of messages waiting to be served at time t at station s, i.e., $w_s(t)\ \forall\, s \in \{1, 2, 3\}$, and in the overall system, i.e., $w(t)$

firing of `timeout`$_3$.

Transient analysis is performed through the approach of Sect. 4.2 with the following parameters: time limit $t_n = 30$ (in so doing, each station is served at least twice), time step 0.1, $\nu_{start} = 20$ (number of stochastic state classes initially enumerated in each unrestricted epoch), $\nu_{iter} = 20$ (number of stochastic state classes enumerated in each unrestricted epoch at each iteration), and $\nu_{max} = 500$ (threshold on the total number of stochastic state classes enumerated in unrestricted epochs). Overall, the analysis evaluates the kernel entries of 135 regenerative epochs: 99 through the analysis under the bounded regeneration restriction, 18 through the analysis under the enabling restriction, and 18 through the heuristic-based approximate analysis. On a machine equipped with an Intel i5-5200U 2.20 GHz and 8 GB RAM, the evaluation takes nearly 40 min, spending less than 0.1 s to perform non-deterministic analysis and classification of regenerative epochs; nearly 40 s, 0.3 s, and 0.4 s to analyze the state space of regenerative epochs under the bounded regeneration restriction, under the enabling restriction, and beyond both restrictions, respectively; approximately 100 s, 180 s, and 2.5 s to evaluate the kernel entries of regenerative epochs under the bounded regeneration restriction, under the enabling restriction, and be-

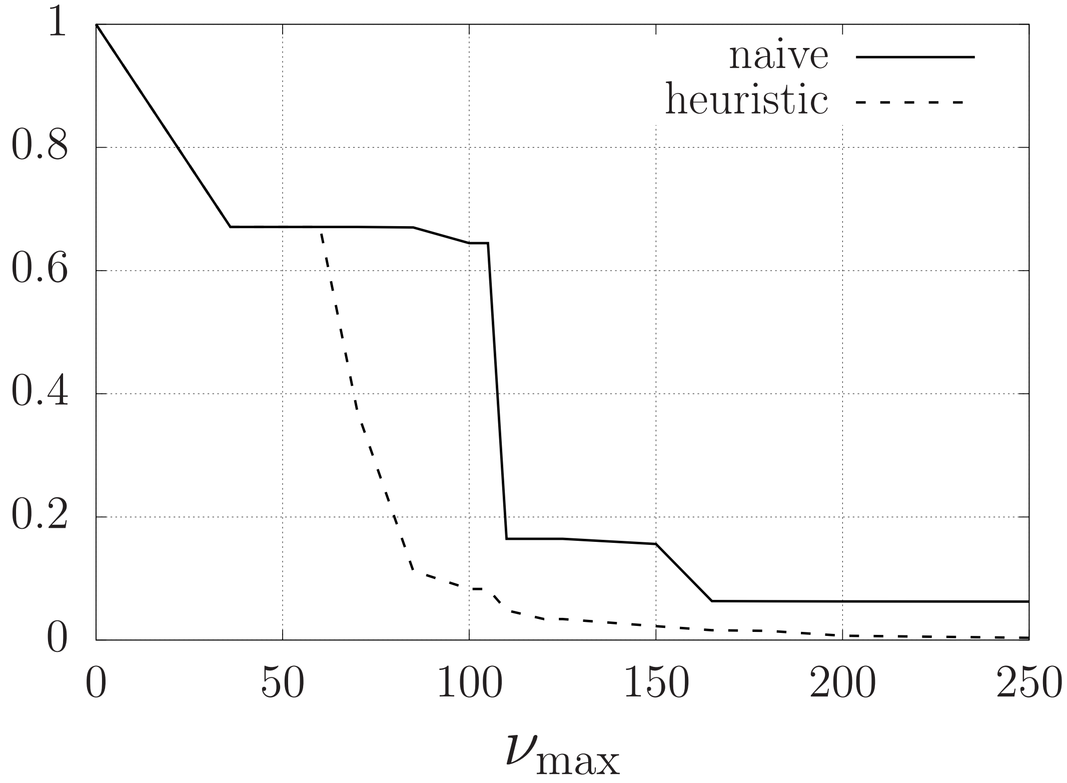


Figure 4.5: total error $\epsilon(t_n)$ (committed in the evaluation of transient probabilities of markings at the time limit t_n) as a function of the number of classes enumerated in unrestricted epochs

yond both restrictions, respectively; nearly 23 s to evaluate the heuristic criterion; and, approximately 34 min to solve the Markov renewal equations. Numbers show that non-deterministic analysis has relatively negligible computational complexity, and thus it can be efficiently used to select the solution technique applied to each regenerative epoch. Notably, the heuristic criterion has a significantly lower cost with respect to the evaluation of the kernel entries of restricted epochs, which mostly depends on the number of encountered regenerations. Overall, results suggest that approximate analysis could be applied also to epochs under enabling or bounded regeneration restrictions to limit state space exploration and reduce evaluation complexity.

To illustrate possible rewards of interest, Fig. 4.4 plots the average number of messages waiting to be served at time t in each station and in the overall system, evaluated according to Equation 4.3 and Equation 4.4, respectively, where i is the initial regeneration (i.e., a stochastic state class with the marking of Fig. 4.3, where all enabled transitions are newly-enabled) and Ψ is the set of markings reached within t_n.

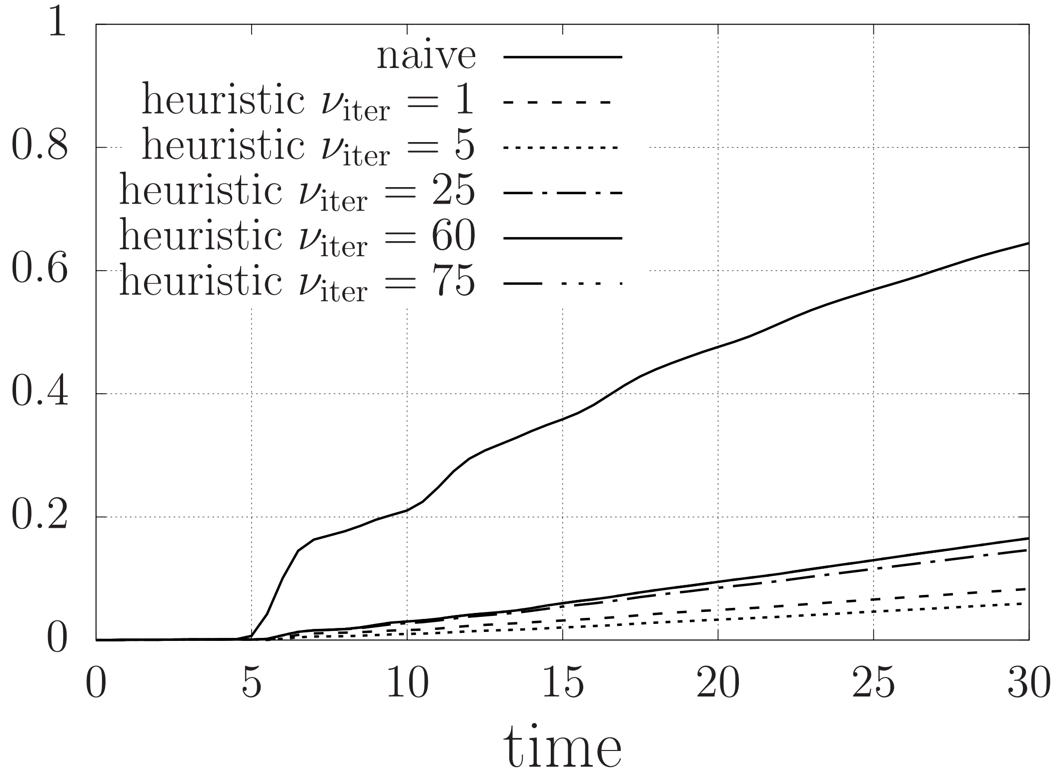


Figure 4.6: total error $\epsilon(t)$ obtained with 70 stochastic state classes enumerated in unrestricted epochs

$$w_n(t) = \sum_{j \in \Psi} \pi_{ij}(t) \cdot j(Waiting_n) \quad \forall\, n \in \{1,2,3\} \tag{4.3}$$

$$w_n(t) = \sum_{j \in \Psi} \pi_{ij}(t) \cdot j(Waiting_n) \tag{4.4}$$

To evaluate the impact of different heuristics in approximate analysis, we evaluate the total defect in the evaluation of transient probabilities of markings, i.e., $\epsilon(t) := \sum_{j \in \Psi} \epsilon_{ij}(t)$ where i is the initial regeneration and Ψ the set of markings, which can be easily computed a posteriori as $\epsilon(t) = 1 - \sum_{j \in \Psi} \pi_{ij}(t)$. Fig. 4.5 plots the total error at the time limit $t_n = 30$ as a function of the threshold ν_{max}, comparing results with those obtained with a *naive approximate analysis* that explores all stochastic transient trees of unrestricted epochs, enumerating ν_{max}/U stochastic state classes in each tree, where U is the number of unrestricted epochs. As expected, $\epsilon(t_n)$ decreases as ν_{max} increases, and the two approaches achieve approximately the same values of $\epsilon(t_n)$ for very small values of ν_{max}. Conversely, when ν_{max} becomes larger than 60, the heuristic-based analysis achieves significantly lower values of $\epsilon(t_n)$, in the order of

$8 \cdot 10^{-2}$ for $\nu_{max} = 100$ and $7 \cdot 10^{-3}$ from $\nu_{max} = 200$ on, with respect to values in the order of 0.65 and $6 \cdot 10^{-2}$ attained by naive analysis, respectively. Overall, these results could be used to select a convenient value of ν_{max} in a trade-off between the result accuracy and the computational complexity.

Fig. 4.6 plots the total error attained by the two approaches as a function of time, with $\nu_{max} = 100$, $\nu_{start} = 2$, and increasing values of ν_{iter}. All curves are around zero until time 5, due to the very low probability that the server has reached station 3 by that time. From time 5 on, the error attained by naive analysis rapidly increases, being nearly 0.21, 0.48, and 0.64 at $t = 10$, $t = 20$, and $t = 30$, respectively. Conversely, $\epsilon(t)$ increases with a much smaller slope for heuristic-based analysis. As expected, the cases with lower values of ν_{iter} achieve better results; for instance, for $\nu_{iter} = 1$, $\epsilon(t)$ is approximately equal to 0.016, 0.049, and 0.083 at $t = 10$, $t = 20$, and $t = 30$, respectively. Values of $\epsilon(t)$ slightly increase with ν_{iter}, though remaining nearly in the same order of magnitude, showing that heuristic-based analysis yields sufficiently accurate results while permitting to limit the computational cost.

Chapter 5

Equilibrium analysis of Markov regenerative processes

Long run dynamics of a system is usually studied through the evaluation of its steady state distribution. This distribution is defined as the probability to be in a particular discrete logical state at steady state, which provides a complete characterization of the system at equilibrium when dealing with Markovian processes. For non-Markovian processes, this distribution is not sufficient for a complete characterization at equilibrium, since the equilibrium also depends on the continuous component of the state, the active timers. Specifically, in non-Markovian models, the remaining time of enabled timers at steady state need also to be evaluated, while in Markovian models this is not necessary due the memoryless property that all enabled timers enjoy. The remaining time of enabled timers at steady state is called *equilibrium density of active timers* (*Equilibrium PDF* in the following), and gives the probability to have a specific vector of remaining times to fire at steady state. In other words, assuming to perform an observation of the system at steady state, this distribution gives the probability to have a specific remaining time for each enabled timer in a specific state. For a non-Markovian process, the steady state distribution along with the equilibrium PDF of remaining times in all possible states allows for its complete characterization at equilibrium. For Markovian process, such an evaluation is not necessary since all timers can be considered as newly enabled due to the memory less property.

In this thesis, we propose an analytical method to evaluate, in addition to the steady-state probability of discrete states, also the equilibrium PDF. This provides the basis to analyze system behavior starting from the equilibrium, such as the evaluation of survivability measures (Heegaard and Trivedi, 2009; Liu and Trivedi, 2006) in non-Markovian models with multiple concurrently enabled timers, which was not possible before.

In the following, we consider a model expressed as STPN, having an *MRGP* underlying process where the bounded regeneration restriction is satisfied in each possible regenerative epoch. The state of the underlying process will be represented through stochastic state classes (defined in Section 3.3.2, Definition 8). A stochastic state class $\Sigma = \langle m, D, f \rangle$ provides a powerful tool to describe the state of the system after a sequence of firings of model transitions, representing both the discrete logical state through the marking m and the continuous part of the state through the support D and the joint PDF f of enabled timers. The equilibrium PDF is directly related to D and f and for this reason a distinct equilibrium PDF needs to be evaluated for each stochastic state class. According to this, we evaluate the probability to be in a specific stochastic state class at steady state (Section 5.1) and afterwards we evaluate the equilibrium PDF in each stochastic state class (Section 5.2). Finally we show experimentally how this method allows one to perform analysis from equilibrium (Section 5.3).

5.1 Steady state probabilities of classes

The evaluation of the steady state distribution of classes is given by Equation 3.15. This time kernels are not aggregated per marking, rather each class is considered separately. In particular this can be done computing the limit of each global kernel entry $G_{ij}(t)$ as $t \to \infty$. These values can be obtained simply as the product of firing probabilities from regeneration i to all classes in LEAVES(i) that reach regeneration $k \in R$ as shown in Equation 5.1.

$$G_{ik}(\infty) = \sum_{\substack{\Sigma \in \text{LEAVES}(i) \text{ s.t.} \\ \Sigma \text{ has regeneration } k}} \rho_\Sigma \tag{5.1}$$

Where ρ_Σ is defined as in section 3.3.2 by Equation 3.9. Afterwards, for each regeneration $i \in R$ and class $j \in \text{INNER}(i)$ the expected time α_j spent in j can be evaluated.

Lemma 4. *Let $\Sigma_j = \langle m, D, f \rangle$ be a stochastic state class such that $\Sigma_j \in \text{INNER}(i)$. Then the expected sojourn time α_{Σ_j} in Σ_j is:*

$$\alpha_{\Sigma_j} = \rho_{\Sigma_j} \sum_{t \in T_E(m)} \mu^{(t)} \int_{D^{(t)}} \tau_{k_t} f^{(t)}(\tau_{age}, \bar{\tau}) d\tau_{age} d\bar{\tau} \tag{5.2}$$

where ρ_{Σ_j} is the product of firing probabilities of transitions that lead from regeneration i to class j; $\mu^{(t)}$ is the probability that $t \in T_E(m)$ fires in j; $D^{(t)} = \{(\tau_{age}, \bar{\tau}) \in D | \text{"}t \text{ fires first"}\}$ is the subset of the support D where t fires first; and

$$f^{(t)}(\tau_{age}, \bar{\tau}) := \frac{f(\tau_{age}, \bar{\tau})}{\int_{D^{(t)}} f(\tau_{age}, \bar{\tau}) d\tau_{age} d\bar{\tau}} \tag{5.3}$$

is the PDF conditioned on t that fires first.

Proof. Equation 5.3 follows from the definition of stochastic state class and from the law of total expectation. The events of the firing in Σ_j of transitions in $T_E(m)$ are mutually exclusive, thus if S_{Σ_j} is the sojourn time in Σ_j, $\alpha_{\Sigma_j} = \rho_{\Sigma_j} * E[S_j] = \rho_{\Sigma_j} \sum_{t \in T_E(m)} \mu^{(t)} E[S_j|\text{t fires first}]$ □

Similarly to Equation 3.15 of Section 3.4, the steady state distribution of a specific class $\Sigma_j \in \text{INNER}(i)$ can be evaluated from Equation 5.4.

$$p_j = \frac{\pi_i \alpha_j}{\sum_{i' \in R, j' \in \text{INNER}(i')} \pi_i' \alpha_{j'}} \tag{5.4}$$

Where $\bar{\pi}$ is such that $\sum_{k \in R} \pi_k = 1$ and $\bar{\pi} = \mathbb{G}(\infty)\bar{\pi}$.

5.2 Equilibrium PDFs

In this section we will describe the derivation of the equilibrium PDF. First briefly recall the concept of renewal process (Kulkarni, 2016) (Section 5.2.1), then we will use this definition for the derivation of the stochastic process of times to fire across renewals (Section 5.2.2) and the derivation of the equilibrium PDF (Sections 5.2.3 and 5.2.4).

5.2.1 Renewal processes

Let's consider a series of events occurring randomly over time and S_n be the time when the n-th event occurs. Assume that $S_0 = 0$ and $S_n \leqslant S_{n+1}$. $S_1, S_2, ...$ are also called renewal times. We define $T_n = S_n - S_{n-1}$ as the *inter-event time*, that is a random variable describing the time between event $n-1$-th and event n-th, where $n \geqslant 1$. Note that $S_n = \sum_{i=1}^{n} T_i$. Let $N(t)$ be the total number of events observed in $(0, t]$, assuming that the event at time 0 is not counted in $N(t)$, but any event at time t is counted. The stochastic process $\{N(t), t \geqslant 0\}$ is called the *counting process* generated by $\{T_n, n \geqslant 1\}$. It is also worth noticing that $N(t) = max\{k : S_k \leqslant t\}$ and that $N(t) = k \Leftrightarrow S_k \leqslant t \leqslant S_{k+1}$. The *renewal process* is a special case of a counting process.

Definition 11. *A counting process $\{N(t), t \geqslant 0\}$ generated by $\{T_n, n \geqslant 1\}$ is called renewal process if $\{T_n, n \geqslant 1\}$ is a sequence of nonnegative identically and independently distributed(i.i.d.) random variables.*

5.2.2 Stochastic process of times to fire across renewals

Without loss of generality, we start considering a generic stochastic state class $\Sigma = \langle m, D, f \rangle$, where $n = |T_E(m)|$ transitions are enabled and respectively named

$t_1, ..., t_n$, and assuming that t_1 is the one that will fire first. Conditioned on this event, the PDF of times to fire in Σ is defined in Equation 5.5.

$$f^{(t_1)}(\bar{\tau}) = \frac{\int_{D^{(t_1)}} f(\tau_{age}, \bar{\tau}) d\tau_{age}}{\int_{D^{(t_1)}} f(\tau_{age}, \bar{\tau}) d\tau_{age} d\bar{\tau}} \tag{5.5}$$

Where $D^{(t_1)} = \{(\tau_{age}, \bar{\tau}) \in D | t_1 \text{ fires first}\}$ is the subset of the support D where τ_1 is minimum and thus t_1 will fire first. The PDF is derived restricting the support to the subset $D^{(t_1)}$, normalizing the PDF (denominator of Equation 5.5) and then marginalizing τ_{age} (numerator of Equation 5.5).

Let us consider successive visits to the stochastic state class Σ. Each time, observed values of $\langle \tau_{age}, \bar{\tau} \rangle$ are i.i.d. according to the PDF f of Σ, since the *MRGP* encounters a regeneration point between visits and then performs exactly the same sequence of transition firings leading to Σ. Note that this sequence corresponds to a unique path in the transient stochastic tree. Moreover, for the same reason, also visits to Σ that end with the firing of t_1 observe the same PDF $f^{(t_1)}$ of $\bar{\tau}$ derived in Equation 5.5 and their sojourn times are i.i.d. random variables. We focus our attention on the time intervals of these i.i.d. sojourn times: as time advances, we move from a sojourn in Σ to the next one, always under the hypothesis that t_1 is the transition that fires in Σ. This admits to build a renewal process $\{N(t), t \geqslant 0\}$ where times between events are distributed as a sojourn in Σ that ends with the firing of t_1. The PDF of inter-event times of this renewal process is given by the marginal PDF of τ_1 given that it fires first as shown in Equation 5.6.

$$g(\tau_1) = \int_{D^{(t_1)}} f^{(t_1)}(\tau_1, \tau_2, ..., \tau_n) d\tau_2 ... d\tau_n \tag{5.6}$$

Where all times to fire except τ_1 have been marginalized.

As $N(t)$ evolves across each renewal $S_0, S_1, S_2, ...$, a new vector of times to fire $\bar{\tau}^{(i)}$, $i = 0, 1, 2, ...$ is sampled independently at each S_i, according to the same PDF $f^{(t_1)}$. The objective is to study the evolution of $\bar{\tau}$ over time, subject to the fact that also the renewal times S_i are random. Let $\{\bar{r}(t), t \geqslant 0\}$ be the n-dimensional stochastic process describing, for each $t \geqslant 0$, the current value of the times to fire vector:

$$\bar{r}(t) := \bar{\tau}^{(N(t))} - (t - S_{N(t)}) \tag{5.7}$$

We denote its PDF by $h(t, \bar{\tau})$ and according to this we can write:

$$P(r_1(t) \leqslant x_1, ..., r_n(t) \leqslant x_n) := \int_{-\infty}^{x_1} ... \int_{-\infty}^{x_n} h(t, \bar{\tau}) dx_1 ... dx_n \tag{5.8}$$

5.2.3 Equilibrium PDF conditioned to the firing of a transition

The goal is to compute the equilibrium PDF of $\bar{r}(t)$, that is the function $\hat{f}^{(t_1)}(\bar{\tau})$ defined in Equation 5.9.

$$\hat{f}^{(t_1)}(\bar{x}) = \lim_{t\to\infty} h(t, \bar{x}) \tag{5.9}$$

This gives the PDF of the times to fire in Σ at equilibrium, conditioned to the fact that the sojourn will end with the firing of t_1.

First, it is necessary to highlight the fundamental relation between $h(t, \bar{x})$, the object of our analysis, and $f^{(t_1)}$.

Lemma 5. *If $h(t, \bar{x})$ is the PDF of $r\bar{(t)}$ for each $t \geqslant 0$, $f^{(t_1)}(\bar{x})$ is the PDF of $\bar{\tau}$ at each renewal, and $g(x)$ is the PDF of τ_1 (the interarrival time of the renewal process, i.e. the time between renewals), the following renewal equation holds:*

$$h(t, \bar{x}) = f^{(t_1)}(\bar{x} + t) + \int_0^t h(t - u, \bar{x})g(u)du \tag{5.10}$$

Proof. Equation 5.10 can be derived by a renewal argument: for the first renewal time S_1 we have that either $S_1 > t$ or $S_1 \leqslant t$.

If $S_1 > t$, then the first renewal has not occurred, so that $N(t) = 0$ and $\bar{r}(t) = \bar{\tau}^{(0)} - t$. The PDF of $\bar{r}(t)$ at time t is then given by $f^{(t_1)}(\bar{x} + t)/P\{\tau_1 > t\}$, i.e., the PDF $f^{(t_1)}$ used to sample $\bar{\tau}^{(0)}$ but conditioned to the event $\{\tau_1 > t\}$ that t_1 have not yet fired, and where each component is shifted by time t (we denote by $\bar{x} + t$ the vector $(x_1 + t, ..., x_n + t)$). According to this and since $S_1 := \tau_1$, we have that $h(t, \bar{x}|S_1 > t)P(S_1 > t) = f^{(t_1)}(\bar{x} + t), .$

If $S_1 \leqslant t$, the process $\bar{r}(t)$ "probabilistic restarts" after S_1, when a new time to fire vector $\bar{\tau}^{(1)}$ is sampled. If $S_1 = u$, $u \leqslant t$ and at least one renewal is encountered by time t, $N(t) = N(t-u) + 1$, $T_{N(t-u)+1} = T_{N(t-u)} + u$ and thus:

$$\bar{r}(t) = \bar{\tau}^{(N(t-u)+1)} - (t - T_{N(t-u)+1}) = \bar{\tau}^{(N(t-u)+1)} - [(t-u) - T_{N(t-u)}] \tag{5.11}$$

for $u \leqslant t$. Given that times to fire vectors $\bar{\tau}^{(N(t-u)+1)}$ and $\bar{\tau}^{(N(t-u))}$ have the same PDF $f^{(t_1)}$, it holds that $h(t, \bar{x}) = h(t - u, \bar{x})$, for $u \leqslant t$. By conditioning on all the possible values of $S_1 = u$ and decreasing t accordingly, we have:

$$h(t, \bar{x}|S_1 \leqslant t)P(S_1 \leqslant t) = \int_0^t h(t - u, \bar{x})g(u)du \tag{5.12}$$

By putting together the two cases, we obtain Equation 5.10. □

Lemma 5 establishes a connection between $h(t, \bar{x})$ and $f^{(t_1)}$ and also reveals the recursive structure of $h(t, \bar{x})$ across renewals. This kind of renewal-type equation is well-known for renewal processes and provides a strategy to compute $h(t, \bar{x})$ at the equilibrium through the following result (Kulkarni, 2016, Theorem 8.17), called the *Key Renewal Theorem*

Theorem 2. *Let $g(x)$ be the PDF of the interarrival time, and let h be a solution to the renewal type equation $h(t) = d(t) + \int_0^t h(t-u)g(u)du$. Then, if d is the difference of two non-negative bounded monotone functions and $\int_0^\infty |d(u)|du < \infty$,*

$$\lim_{t\to\infty} h(t) = \frac{1}{E[S]} \int_0^\infty d(u)du \tag{5.13}$$

where $E[S] = \int_0^\infty ug(u)du$ is the mean interarrival time.

This theorem can be directly applied to Equation 5.10 for $d(t) = f^{(t_1)}(\bar{x} + t)$ and requires only mild conditions on $f^{(t_1)}(\bar{x} + t)$. When the PDFs f_t used to sample newly-enabled transitions are piecewise expolynomials (products of exponentials and polynomials), the joint PDF f of timers, and thus $f^{(t_1)}$, is also piecewise continuous with bounded variation (Carnevali et al., 2009).

By combining Lemma 5 and Theorem 2, we obtain the equilibrium PDF $\hat{f}^{(t_1)}$ of $\bar{\tau}$ in Σ when t_1 is the transition that fires at the end of each sojourn:

$$\hat{f}^{(t_1)}(\bar{x}) := \lim_{t\to\infty} h(t, \bar{x}) = \frac{1}{E[S^{(t_1)}]} \int_0^\infty f^{(t_1)}(\bar{x} + u)du \tag{5.14}$$

where $E[S^{(t_1)}] = \int_0^\infty ug(u)du$ it the mean sojourn time in Σ when t_1 fires. The identity of Equation 5.14 is a major step for the analysis of the joint PDF of $\bar{\tau}$ at steady state. Combined with Equation 5.5 to obtain $f^{(t_1)}$ from f and with Equation 5.6 to obtain g, it provides a straightforward derivation of the equilibrium PDF under the hypothesis that t_1 is always the transition that fires first in Σ.

5.2.4 Equilibrium PDF when multiple transitions can fire

We can now remove the hypothesis previously made that t_1 is the transition to fire at each visit through the following theorem.

Theorem 3. *Let $\Sigma = \langle m, D, f \rangle$ be a stochastic state class where transitions $t_1, ..., t_n$ can fire with probability $\mu^{(t_1)}, ..., \mu^{(t_n)}$, respectively. Then, the equilibrium PDF of $\bar{\tau} = (\tau_1, ..., \tau_n)$ is given by*

$$\hat{f}(\bar{x}) = \frac{1}{E(S)} \sum_{i=1}^{n} \mu^{(t_i)} \int_0^\infty f^{(t_i)}(\bar{x} + u)du \tag{5.15}$$

where $E(S)$ is the expected sojourn time in Σ and, for all $i = 1, ..., n$, $f^{(t_i)}$ is the PDF of $\bar{\tau}$ conditioned on the firing of t_i according to Equation 5.5.

Proof. We focus only on sojourns in class Σ and ignore the rest of the time line. The probability that a sojourn ends with the firing of t_i is $\mu^{(t_i)}$ for $i = 1, ..., n$, with $\sum_{i=1}^{n} \mu^{(t_i)} = 1$; conditioned on this event, the expected sojourn time in Σ is $E[S^{(t_i)}]$. Then, the steady state probability of sojourns in Σ that end with the firing of t_i is given by

$$p_i = \frac{\mu^{(t_i)} E[S^{(t_i)}]}{\sum_{j=1}^{n} \mu^{(t_j)} E[S^{(t_j)}]} \tag{5.16}$$

which is the mean fraction of time spent in such sojourns. Since $\hat{f}^{(t_i)}$ is the equilibrium PDF when sojourns are conditioned to end with t_i,

$$\begin{aligned} \hat{f}(\bar{x}) &= \sum_{i=1}^{n} p_i \hat{f}^{(t_i)}(\bar{x}) = \sum_{i=1}^{n} \left(\frac{\mu^{(t_i)} E[S^{(t_i)}]}{\sum_{j=1}^{n} \mu^{(t_j)} E[S^{(t_j)}]} \right) \hat{f}^{(t_i)}(\bar{x}) \\ &= \frac{1}{\sum_{j=1}^{n} \mu^{(t_j)} E[S^{(t_j)}]} \sum_{i=1}^{n} \mu^{(t_i)} \int_0^{\infty} f^{(t_i)}(\bar{x} + u) du \end{aligned} \tag{5.17}$$

which, since $\sum_{j=1}^{n} \mu^{(t_j)} E[S^{t_j}] = E[S]$, gives Equation 3. □

We have thus derived the equilibrium PDF in a specific stochastic state class Σ.

5.3 Experimental validation

Steady-state probabilities and equilibrium PDFs represent the equilibrium distribution of the MRGP. When used as initial distribution for transient analysis, this distribution must result in constant transient probabilities that are equal to steady-state ones. We now show how to perform transient analysis from this distribution and evaluate the correctness of the approach.

In Section 5.1, steady-state probability p_Σ of each class $\Sigma \in \cup_{i \in R}$INNER$(i)$, where R is the set of regenerations of the process. Given that the MRGP is in class $\Sigma = \langle m, D, f \rangle$, the marking is equal to m and the times to fire vector $\bar{\tau}$ has equilibrium PDF given by $\hat{f}(\bar{x})$, which is computed form f according to Equation 3. To compute transient probabilities from the equilibrium, it is necessary to modify the Markov Renewal Equation of Equation 3.4 and the approach for the enumeration of transient stochastic trees as follows.

First, for each inner node $\Sigma = \langle m, D, f \rangle\ c \in \cup_{i \in R} INNER(i)$, we compute a tree of stochastic state classes until the next regeneration. We construct the initial class START(Σ) of this tree using a marking m and PDF of $\langle \tau_{age}, \bar{\tau} \rangle$ equal to $g(x_{age}, \bar{x}) = \delta(x_{age}) \hat{f}(\bar{x})$, i.e., $\tau_{age} = 0$ and the times to fire vector $\bar{\tau}$ has PDF at equilibrium. For each $\Sigma \in \cup_{i \in R} INNER(i)$ we denote the inner nodes of the tree computed from START(Σ) (until following regenerations) as STARTINNER(Σ), while the leaves of the tree are denoted as STARTLEAVES(Σ).

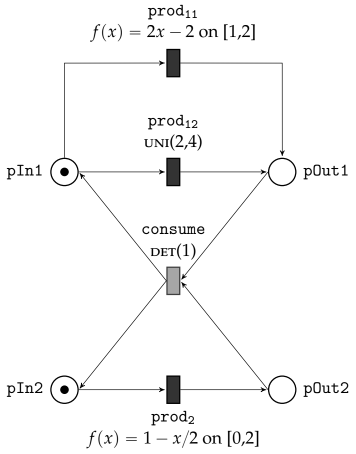


Figure 5.1: STPN model of a parallel Producer-Consumer

Then, we extend the Markov renewal equations of Equation 3.4 by introducing an additional regeneration $\hat{r}$ that represents the state of the MRGP at equilibrium. The process starts in $\hat{r}$ at time $t = 0$, but never returns to this artificial regeneration: by construction, the next regeneration belongs to R and, afterward, the MRGP cycles through its original trees of stochastic state classes. To achieve this behavior, we set MRGP kernel entries as follows. Let $\hat{R} = R \cup \hat{r}$ and set, for $i = \hat{r}$,

$$L_{ij}(t) = \sum_{\Sigma \in \cup_{i' \in R} INNER(i')} p_{\Sigma} \left(\sum_{\substack{\Sigma' \in \text{STARTINNER}(\Sigma) s.t. \\ \Sigma' \text{ has marking } j}} p_{in}(\Sigma', t) \right) \tag{5.18}$$

$$G_{ik}(t) = \sum_{\Sigma \in \cup_{i' \in R} INNER(i')} p_{\Sigma} \left(\sum_{\substack{\Sigma' \in \text{STARTLEAVES}(\Sigma) s.t. \\ \Sigma' \text{ has marking } j}} p_{reach}(\Sigma', t) \right) \tag{5.19}$$

for all $k \in R$, $j \in M$, and $t \geqslant 0$. Since $\hat{r}$ is never reached again, we set $G_{ik}(t) = 0 \forall i \in \hat{R}$ when $k = \hat{r}$. Kernel entries in the additional row $\hat{r}$ model a random choice of the initial stochastic state class Σ according to the discrete distribution given by p_{Σ} for $\Sigma \in \cup_{i \in R} INNER(i)$; for a given class Σ, the tree computed from START(Σ) is used to characterize the system evolution from the equilibrium in Σ until the next regeneration. As in Section 3.3.2, measures $p_{in}(\Sigma, t)$ and $p_{reach}(\Sigma, t)$ provide the probability that the MRGP is in the stochastic state class Σ at time t, or that it has reached Σ by time t, respectively.

Consider the STPN model of Figure 5.1. There are 4 possible markings $M =$

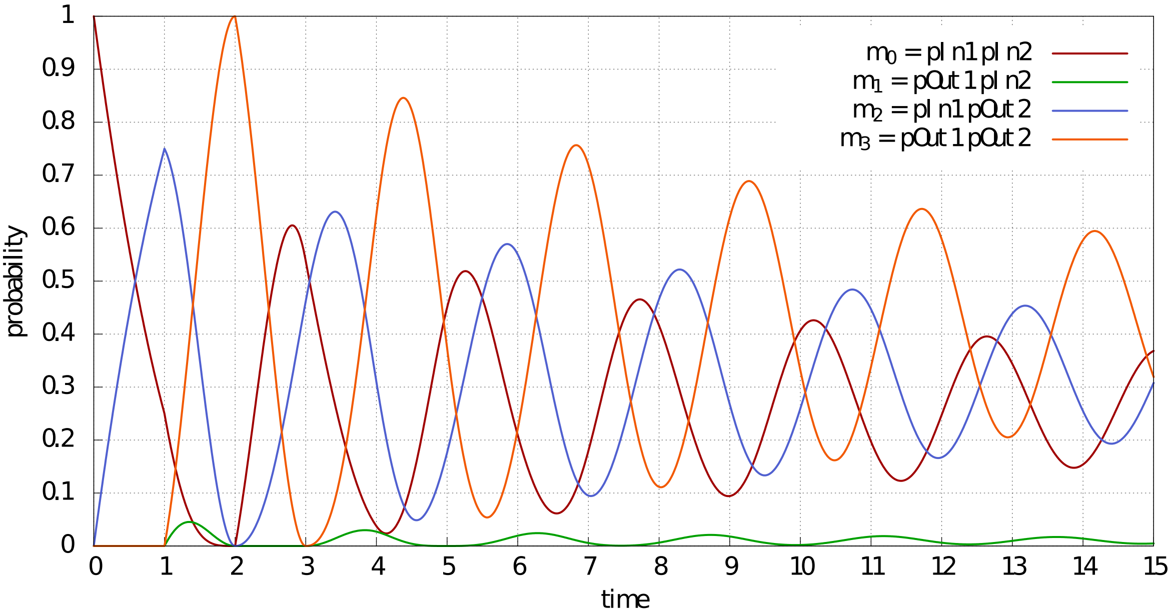


Figure 5.2: Transient probabilities of the model states from the initial marking $i = [pIn_1, pIn2]$

$\{[pIn_1, pIn2], [pOut_1, pIn2], [pIn_1, pOut2], [pOut_1, pOut2]\}$ and two regenerative states $R = \{[pIn_1, pIn2], [pOut_1, pOut2]\}$. Steady state probabilities of markings are $\pi_{[pIn_1, pIn2]} = 0.263$, $\pi_{[pOut_1, pIn2]} = 0.010$, $\pi_{[pIn_1, pOut2]} = 0.317$ and $\pi_{[pOut_1, pOut2]} = 0.410$. Figure 5.2 shows evaluated transient probabilities $\pi_{ij}(t)$ for $0 \leqslant t \leqslant 15$ of the MRGP for $i = [pIn_1, pIn2]$, that is the initial regeneration, and for each $j \in m$. Figure 5.3 shows instead evaluated transient probabilities $0 \leqslant t \leqslant 15$ and each $j \in M$, where the initial state is the additional regeneration $\hat{r}$ and additional kernels rows are evaluated using Equations 5.18 and 5.19. As expected, using both the steady state distribution and equilibrium PDFs, we can evaluate transient probabilities starting from the equilibrium and according to this evaluated distributions are constant and correspond to the steady state probabilities.

5.4 Additional remarks

In this chapter, we presented a solution to compute a closed-form expression of the equilibrium distribution of MRGPs. The solution leverages the calculus of stochastic state classes and it has been validated through an implementation based on the ORIS tool (Bucci et al., 2010).

In future work, this technique can provide a basis for the evaluation of survivability measures (Heegaard and Trivedi, 2009) on non-Markovian models, that we consider to be the major contribution of this approach. Another possible minor application could be to use the same theoretical concepts for the estimation of serving

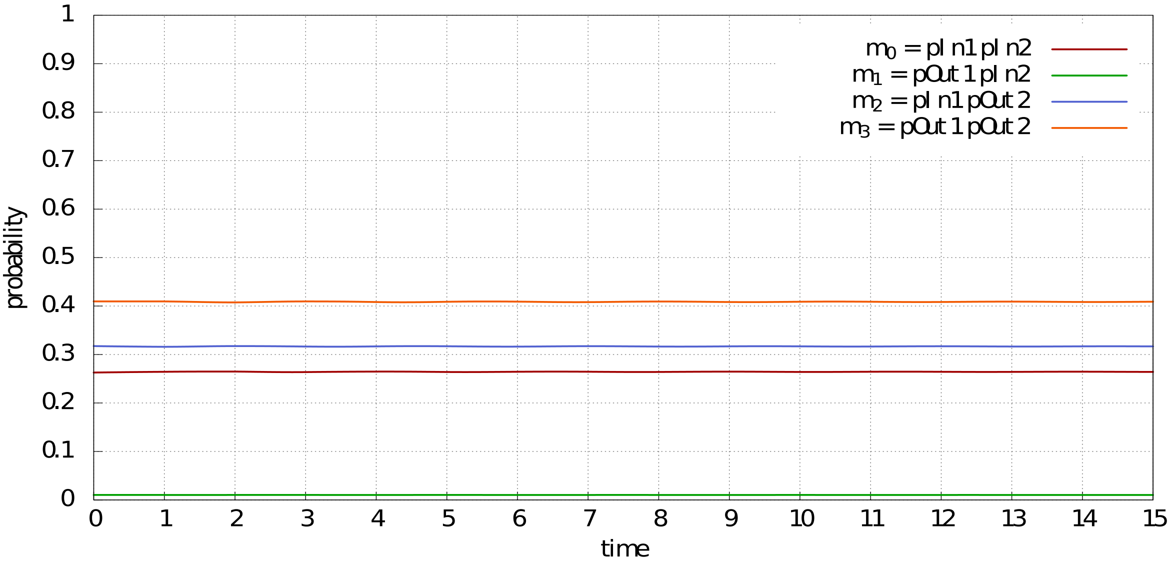


Figure 5.3: Transient probabilities evaluated using steady state distribution and equilibrium PDF as initial state

time of a queue, based on observations. Specifically, suppose to have a queue with multiple servers, where customers arrive and need to wait a certain amount of time to be served. When a customer arrives, he doesn't know the service time distribution, but wants to infer it based on observations that he can do in the short term. Initially, he can only observe how long customers a the top of the queue need to be served. This first observed time is not a sample of the real service time, but are instead remaining times at equilibrium. According to this, the theory discussed in this chapter could be used to immediately estimate service times having only observed values at equilibrium.

Chapter 6

Analysis of hierarchical semi-Markov processes with parallel regions

In the previous chapters models were represented using the STPN formalism, which has a great expressiveness but is not widespread outside the research context. Other approaches (Jansen et al., 2003; Homm and German, 2016; Gnesi et al., 2000) are based instead on derivations of more common formalisms like state charts (Harel, 1987) and UML state machines (Group, 2018). The use of these UML derived formalisms, can help in the spread of techniques of non-Markovian analysis, allowing engineers to study more realistic models of systems.

Additionally, the application of non-Markovian analysis techniques, requires the user to have deep knowledge about the technique to understand which restrictions the model needs to satisfy (bounded, enabling or neither) and thus which technique can be applied and which one is more efficient. Also using approaches that automatically verify which technique can be applied as shown in Chapter 4, the user needs to understand how some assumptions can make the model analyzable efficiently and how much these assumptions impact on final results. Alternatively, the usage of other types of formalism, can be exploited so as to guarantee that each possible model expressed in such formalism can be analyzed or at least that it is easier to verify whether it is analyzable. Finally, formalisms can also be defined so as to guarantee a particular structure of the model, which can be exploited to implement analysis techniques that are more efficient and admit to analyze a broader class of processes, compared to more general techniques described in Chapters 3 and 4.

The approach of (Homm and German, 2016) exploits the hierarchical structure of a UML derived formalism in order to develop a compositional approach for the evaluation of steady state probabilities of the model, notably without the requirement to build the whole state space. The limitation of such work was mainly related to the restricted number of features adopted from the UML state machine formalism. Specifically the derived formalism defines a state machine, where each state can be simple or composite. Simple states have a sojourn time distribution describing how long the system remains in a particular state, while composite states are described by a set of parallel regions, where each region is in turn described by another state machine. Additionally, regions can have final states or exit states describing the condition that needs to be met in order to leave the composite state. One major limitation of the approach is that it doesn't allow to have different successors of a state based on how concurrency is resolved. Additionally the technique imposes restrictions on how exit states can be included in the model. In this chapter, we show how such a formalism can be extended in order to increase its expressiveness (Biagi et al., 2018). Specifically we remove the limitation to have exit states only at the lower level of the hierarchy of a composite state, allowing to define more complex concurrency patterns. This is achieved by applying the concept of time advancement mechanism known from state classes (Horváth et al., 2012). It is also combined with a computation of probabilities to reach nested states, which is needed for the final computation of steady-state probabilities. We also add the possibility to have exit states on the border, making it possible to have different successor states based on which region finishes first, substantially increasing the expressivity of the formalism. Finally we introduce the possibility to have history states, enabling the definition of more condensed models. The new technique has also been implemented in order to experimentally validate it and to better characterize the advantages of having such a compositional approach.

The following sections are organized as follows: in Section 6.1 we present the extended formalism and give a formalization of it; in Section 6.2 we describe the analysis technique; finally in Section 6.3 we report an application example and study the efficiency of the technique.

6.1 Hierarchical semi-Markov process with parallel regions

The adopted formalism is derived from UML state machines (Group, 2018), allowing only the usage of a subset of its features but also adding some extensions to model quantitative aspects as done in, e.g., (Jansen et al., 2003). A previous version of the formalism was presented in (Homm and German, 2016), but here it is enriched with the concept of history states which allows for a more concise representation, and for exit states on the border, which allow for distinct successor states based on which region concludes first.

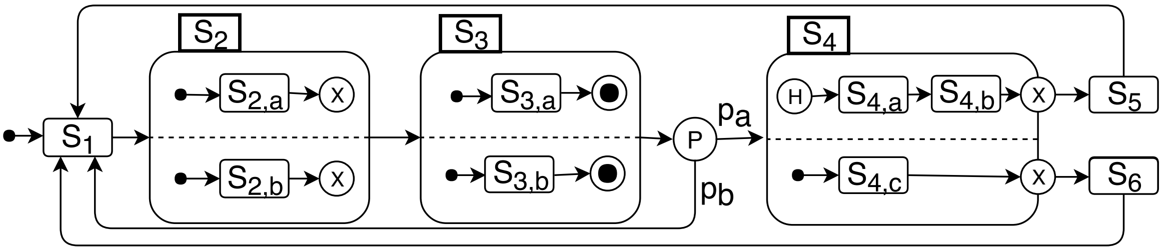


Figure 6.1: Example model defined with the formalism

6.1.1 Description and graphical representation

The model describes a state machine, thus a system that can be in exactly one state at any given time. The time elapsed in a particular state is stochastically distributed, and when such time expires the system changes its state. Each state can be *simple* or *composite*. Time spent in a simple state is described by a stochastic distribution and there are no additional details regarding the internal representation of the state. Instead the internal state of a composite state is described by a set of one or more parallel regions, each one described again as a state machine. When a composite state is entered, each region has its own internal state and only when a specific condition between regions is met the composite state is left. Recursively applying this concept allows to define hierarchical models, since in the same composite state it is possible to have sub-states that in turn are composite, thus having more levels of detail in the same composite state. The highest of such levels will be called hereafter *top level*.

An example model is shown in Figure 6.1. At any given time at top level, the system can be in states S_1, S_2, S_3, S_4, S_5 or S_6, where S_1 is the initial state identified by an arrow with a black circle on its origin. States S_1, S_5 and S_6 are simple states, while states S_2, S_3 and S_4 are composite states where their internal representation is described through regions and sub-models. Additionally, the successor of state S_3 is a *P-pseudonode*, introduced in (Jansen et al., 2002), that is used to handle discrete probabilistic branching. Specifically, after state S_3 with probability p_a the next state is S_4 and with probability $p_b = 1 - p_a$ the next state will be S_1. In this example, composite states are all defined by a set of two regions, each one describing a nested state machine. When state S_2 is entered, both regions have their own current state, $S_{2,a}$ and $S_{2,b}$, respectively. In this case the end condition is defined by exit pseudostates, which denote that the composite state is left when one of the exit pseudostate is reached first. Exit pseudostates are graphically represented as a circle with a cross. Composite state S_3 is modeled with a different kind of end condition, by using final pseudostates, which denotes that the state is left when both final states are reached. Final states are graphically represented as a circle with an inscribed black circle. Finally, state S_4 uses again exit pesudostates, but they are placed on the border of the composite state, meaning that the state reached when state S_4 is left depends on which region completes at first its execution. If the upper region ends first, the next state will be S_5, otherwise, if the lower region ends first, S_6 will be the next state. Moreover, the upper region of state S_4 has an initial state depending on its history. A history state allows a region to keep track of the state it was in when it was last exited. When the system enters again in such a region, the region returns to this same state. The first time the state is entered a *default history state* is specified, in this case it is $S_{4,a}$. This is represented as described by the UML specification (Group, 2018). Note also that if the absorbing state was reached in the previous execution, next time the region is entered the default history state will be selected. According to this, history states have a meaning only in regions where exit pesudostates are present. As a restriction, we require that all regions on a level directly below a composite state must have the same type of end pseudostate.

6.1.2 Formal definition

First we recall the definition of semi-Markov processes (SMP) (Kulkarni, 2016). Consider a stochastic process $\{X(t), t \geqslant 0\}$ with a countable number of states. It starts in an initial state X_0 at time $t = 0$, stays for a sojourn time T_1 and then changes its state to a new value X_1. In general it stays in a state X_n for a duration T_{n+1} and then jumps to a state X_{n+1}.

Definition 12. *A stochastic process $X(t), t \geqslant 0$ is called SMP if it has a countable number of states and the sequence $\{X_0, (X_n, T_n), n \geqslant 1\}$ satisfies $P(X_{n+1} = j, T_{n+1} \leqslant t | X_n = i, T_n, X_{n-1}, T_{n-1}, ..., X_1, T_1, X_0) = P(X_1 = j, T_1 \leqslant t | X_0 = i) = G_{i,j}(y)$.*

The matrix $\mathbf{G}(y) = [G_{i,j}(y)]$ is called global kernel of the SMP process.

Definition 13. *A hierarchical SMP with parallel regions (HSMP) is a tuple $\theta = \langle R, \rho, S, P, F, \phi, \eta \rangle$, where R is the set of regions; $\rho : R \to S$ is a function that identifies the initial state of a region; $S = S_s \cup S_c \cup S_a$ is the set of states where S_s are simple states, S_c are composite states, $S_a = S_a^E \cup S_a^F$ are absorbing states, S_a^E are absorbing states of type exit, S_a^F are absorbing states of type final; $P : S \times S \to \mathbb{R}$ is the matrix that describes the discrete probability that the successor of a state will be a specific state; $F : S_s \to CDF$ associates each simple state to a cumulative distribution function; $\phi : S_c \to \theta$ associates the composite state with another HSMP that describes its internal representation.*

The definition is recursive since each composite state is described through a nested HSMP. It is worth noting that for convenience this definition defines branching with a matrix P and not as a pseudostate. It would also be easy to associate state transitions with timing. If we consider the model of a particular region, without considering sub-states of composite states, the underlying process constitutes a semi-Markov process. The reason is that the state space of the process is composed by the states of the region, where the sojourn time depends only on the current state and not on the history as required by Definition 12. In particular if the current state is a simple state $s \in S_s$, the distribution of time to reach the next state is simply given by $F(s)$. If instead the current state is a composite state it can be evaluated as will be shown in Section 6.2 and the distribution does not depend on the history.

Finally it should be noted that exits on borders and history states are not directly formalized in the above definition, because as we will see in Section 6.2 they need special handling to be considered.

6.2 Analysis technique

The steady-state analysis of an *HSMP* presented in (Homm and German, 2016) is extended here notably to increase its applicability. Specifically it can now be applied to analyze models with history states, with exits on borders and with a complex structure of exit states, not handled by the previous version of the technique.

The analysis technique allows to evaluate steady-state probabilities for top level states and for states nested inside composite states of the *HSMP*. As in the steady-state analysis of an *SMP* (Kulkarni, 2016), the idea behind the analysis is to build the *embedded DTMC* of the top level *SMP*, evaluate its steady-state probabilities and then evaluate steady-state probabilities of the top level *SMP* using mean sojourn times of states. Subsequently, the steady state probability of nested states can be evaluated by evaluating the ratio of time spent in each nested state.

The analysis is organized in the following 6 steps:

1. **Sojourn time distributions** are evaluated for each composite state, without considering possible exits in parallel regions (Section 6.2.1)

2. **Exit distributions** are computed, defined as the probability to leave a region at time t due to an exit in a parallel region (Section 6.2.2)

3. **Reaching probabilities of sub-states** are computed, considering also possible exits in parallel regions (Section 6.2.3)

4. If exits on the border are present, **probabilities to exit from a specific exit** are computed (Section 6.2.4)

5. **Mean sojourn times** are evaluated, considering also possible exits in parallel regions (Section 6.2.5)

6. **Steady-state probabilities** of the *HSMP* are evaluated (Section 6.2.6)

In the following each step will be explained in detail, first without considering history states, then describing required extensions for analyzing models with history states in Section 6.2.7

6.2.1 Evaluation of the sojourn time distributions

The first step of the analysis is to evaluate the sojourn time distributions of composite states, without considering possible exits that may occur in parallel regions on the same level or at higher hierarchy levels. This is done with a bottom-up approach starting from the deepest level where only simple states are present and then going up through the hierarchy exploiting sojourn times of composite states evaluated in previous steps. Following this approach, when evaluating the sojourn time distribution of a state $s_k \in S^i_c$ of an *HSMP* $\theta^i = \langle R^i, \rho, S^i, P^i, F^i, \phi^i, \eta^i \rangle$, we first evaluate the distribution of time to reach the end state of each region and then compose them. In particular, being $\phi^i(s_k) = \theta^j = \langle R^j, \rho^j, S^j, P^j, F^j, \phi^j, \eta^j \rangle$ the *HSMP* describing the internal representation of the composite state s_k, we need to evaluate it for each region $r_q \in R^j$. Let ψ_{r_q} be the distribution of time to reach the end state of the region r_q, it can be derived from the transient probabilities of the underlying *SMP*. Transient probabilities of an *SMP* can be evaluated according to Equation 6.1 (Kulkarni, 2016).

$$\mathbf{V}_{r_q}(t) = \mathbf{E}_{r_q}(t) + \int_0^t d\mathbf{G}_{r_q}(u)\mathbf{V}_{r_q}(t-u) \tag{6.1}$$

Where $\mathbf{V}_{r_q}(t)$ is the matrix of transient probabilities to be in a state of the SMP at time t given an initial state, $\mathbf{E}_{r_q}(t) = I - diag(H_1(t), ..., H_{|T_{r_q}|}(t))$ is the local kernel of the process, T_{r_q} is the set of states of the process and $H_i(t)$ is sojourn time distribution of the state $s_i \in T_{r_q}$. Let $\bar{e}$ be the vector with 1 in the position of absorbing states of the SMP and 0 otherwise and $\bar{l}$ the vector having 1 in the position of the initial state of the SMP given by $\rho(r_q)$ and 0 otherwise, the sojourn time distribution of the region is given by $\psi_{r_q}(t) = 1 - \bar{l} * \mathbf{V}_{r_q}(t) * \bar{e}$. After evaluating $\psi_{r_q}(t)$ for each region of the composite state s_k, its sojourn time distribution can be derived. If end states are final states, the distribution of the sojourn time of s_k is the maximum over the sojourn time of its regions, evaluated as $\Omega_{s_k}(t) = \prod_{q=1}^{|R^j|} \psi_{r_q}(t)$, otherwise if all end states are exit states its distribution of sojourn time is given by the minimum over the sojourn times of its region evaluated as $\Omega_{s_k}(t) = 1 - \prod_{q=1}^{|R^j|}(1 - \psi_{r_q}(t))$

6.2.2 Evaluation of exit distributions

The evaluation of the mean sojourn time of a state needs to take into account its distribution of sojourn time, but also the possibility that an exit occurs during its sojourn in some parallel region. According to this we evaluate the distribution of

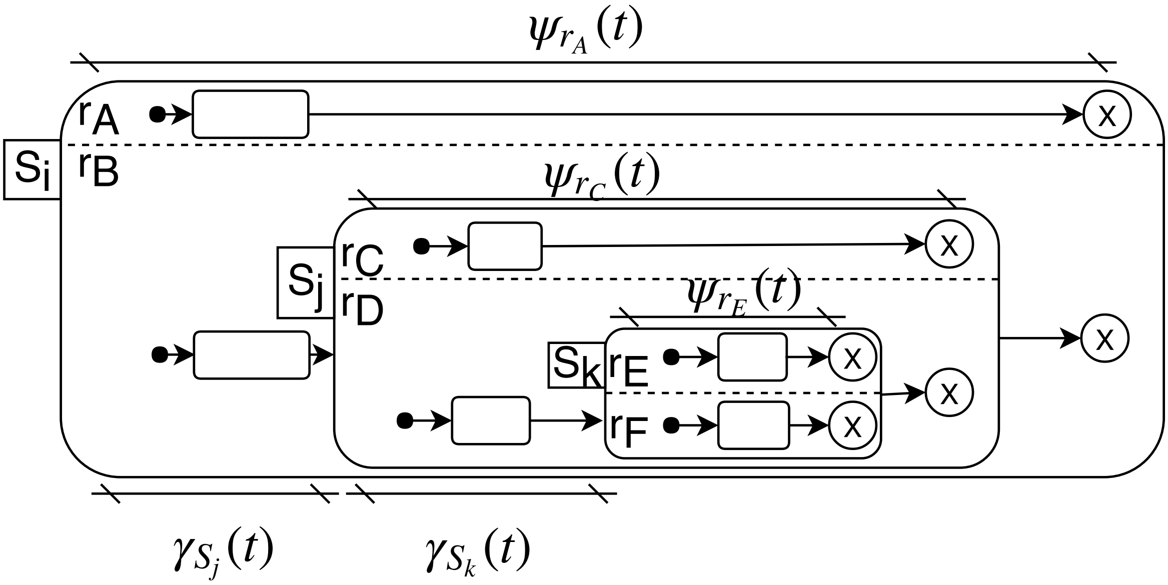


Figure 6.2: Evaluation of the exit distribution of a region

probability $F^{exit}_{r_i}(t)$ that sojourn in a region r_i is interrupted at time t due to an exit occurring in a parallel region of the same composite state or in a higher level parallel region. Consider Figure 6.2 where $F^{exit}_{r_F}(t)$ needs to be evaluated. An early exit in such region can occur if an exit occurs in region r_E, r_C or r_A. $F^{exit}_{r_i}(t)$ can be evaluated with a top-down approach exploiting the fact that the behavior in each region is independent from other regions except that when an exit occurs. Additionally we assume that the evaluated distribution needs to have as time origin the time at which r_i is entered. Since higher level parallel regions were entered before the entrance in region r_i, we need to condition the probability that such regions will cause an exit to the fact that time has already passed when region r_i is entered.

First, it is useful to recall that the conditioning of a distribution to the passage of time can be done using the *time advancement* operation introduced in (Horváth et al., 2012):

Definition 14. *Let τ_A and τ_B be two random variables of the sojourn time in two concurrent states A and B distributed according to the probability density functions $f_{\tau_A}(t)$ and $f_{\tau_B}(t)$, respectively. Let's assume that $\tau_A < \tau_B$. When A is exited, the remaining sojourn time τ'_B in B is reduced by τ_A, $\tau'_B = \tau_B - \tau_A$ and is distributed as $f_{\tau'_B}(t) = \int_{Min(\tau_A)}^{Max(\tau_A)} f_{\tau_B}(t + x) f_{\tau_A}(x)dx$. This operation is called time advancement or time shift.*

Let $\overline{\psi}$ be the set of sojourn time distributions of each region of the model evaluated in Section 6.2.1. The top-down algorithm for the evaluation of exit distributions for region r_i is reported in Listing 6.1.

Listing 6.1: Evaluation of exit distributions

```
1
2  procedure evaluateExitDistribution($r_i$, $\overline{\psi}$)
3    //Bottom-up reasearch of parents
4    parentStates = []; //Stack
5    parentRegions = [$r_i$]; //Stack
6    parent = getParentState($r_i$); //Find the state that directly contains $r_i$
7    while( parent is not null)
8      parentStates.push(parent);
9      containingRegion = getParentRegion(parent);
10     if( containingRegion is not null)
11       parentRegions.push(containingRegion);
12       parent = getParentState(containingRegion);
13     else
14       parent = null;
15   //Top-down evaluation of the distribution
16   $F_{r_i}^{exit}(t)$ = $U(t-\infty)$;
17   currentState = parentStates.pop();
18   while( currentState is not null)
19     currentRegion = parentRegions.pop();
20     regions = getRegions(currentState);//Get regions of the composite state
21     regions = regions.removeElement(currentRegion);
22
23     if( currentRegion has exit pseudostate)
24       for( $r$ in regions)
25         $F_{r_i}^{exit}(t)$ = min($F_{r_i}^{exit}(t)$, $\psi_r(t)$);
26
27     nextState = parentStates.pop();
28     if( nextState is not null)
29       smp = buildSMP(currentRegion); //Build the model of the region
30       $\gamma(t)$ = smp.evaluateTransientTo(nextState); //Evaluate time to be absorbed
31       $F_{r_i}^{exit}(t)$ = timeAdvancement($F_{r_i}^{exit}(t)$, $\gamma(t)$); //Apply the time advancement
32
33     currentState = nextState;
34   return $F_{r_i}^{exit}(t)$;
```

In order to better understand the algorithm we apply it to the model of Figure 6.2. We want to evaluate the exit distribution for region r_F, so procedure parameters are r_F and $\overline{\psi}$. From line 3 to 13, the model is visited with a bottom-up approach creating two stacks of parent states of the considered region and regions containing such parent states. At the end of this first step $parentStates = [S_k, S_j, S_i]$ and $parentRegions = [r_F, r_D, r_B]$. Note that the two stacks are equipped with the classic *push()* and *pop()* operations. Then the procedure from line 15 to 32 effectively evaluates the exit distribution by iterating with a top-down approach until the target region is reached. The exit distribution is initialized at line 15, with a unit step function in ∞. Then the top-down approach iterates from S_i over S_j to S_k. At lines 19 and 20 all parallel regions of the current region are found. Then from line 22 to line 24, if the current region has a final pseudostate, it has no exit and so also its parallel regions do not contribute to the exit distribution. Otherwise the total exit distribution is given by the minimum between the previous regions exit distributions and the sojourn time distribution of these regions. Finally if another level in the hierarchy is present, we need to apply a time advancement to the evaluated distribution, equal to the time required to be absorbed in the state containing the target region,

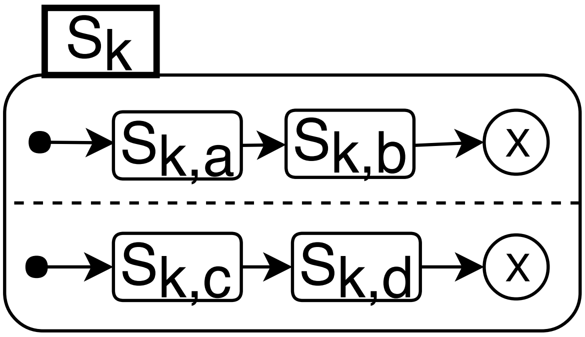


Figure 6.3: Example of a composite state with exit states

so that at the next step all considered distributions will continue to have the same time origin. To help understand, Figure 6.2 highlights the distributions used in the evaluation of $F_{r_i}^{exit}(t)$. At the first step $F_{r_i}^{exit}(t) = \psi_{r_A}(t)$ since r_A is the only parallel region to r_B and the minimum between the step function in ∞ and $\psi_{r_A}(t)$ is the sojourn time distribution of r_A. The algorithm goes down to the next level, and first the distribution $\gamma_{S_j}(t)$ that represents the time to be absorbed in S_j is evaluated, then a time advancement operation is applied to $F_{r_i}^{exit}(t)$ by subtracting $\gamma_{S_j}(t)$. At the next step the minimum between the previous $F_{r_i}^{exit}(t)$ and $\psi_{r_C}(t)$ is evaluated and then a time advancement of $\gamma_{S_k}(t)$ is applied. Finally the minimum between the previously evaluated $F_{r_i}^{exit}(t)$ and $\psi_{r_E}(t)$ gives the final result of the evaluation.

This algorithm evaluates the $F_{r_i}^{exit}(t)$ distribution for each region of the model. The approach has a limitation: it can't be applied if a lower level region contains cycles which includes a composite state. The reason is that if cycles with a composite state are present, when evaluating the exit distribution we need to evaluate the time to be absorbed by the target state in order to apply the time advancement conditioning. But if cycles are present, it is possible to enter in a sub-region, exit and then enter again and according to this the evaluation of the absorbing time is no longer compositional, but requires to analyze the model as a whole instead of through a succession of isolated evaluations. Note that this restriction applies only to the case of regions with exit states, if only final states are present there is no need to evaluate the absorbing probabilities.

Finally it is important to highlight that the early version of the technique presented in (Homm and German, 2016) didn't use this algorithm and thus it was limited to the case in which exit states were possible only at bottom levels of composite states.

6.2.3 Evaluation of the reaching probabilities of nested states

Consider the composite state of Figure 6.3. When we evaluate the steady-state probabilities of sub-states $S_{k,b}$ or $S_{k,d}$ we need to consider both the ratio between the

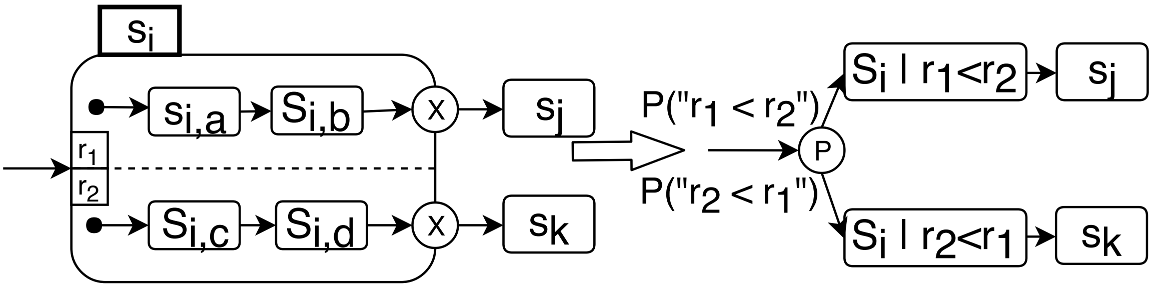


Figure 6.4: Example of a composite state with exits on the border reaching distinct successor

sojourn time spent in the parent state and in the nested state and also the probability that the nested state will be visited. The reason is that in both regions of the composite state it is possible that an exit occurs in the other parallel region before the nested state will be reached. More generally, the probability μ_{s_i} to visit a sub-state s_i given that the parent state is visited is $\mu_{s_i} \leqslant 1$ if it is not the initial state of the region and there are parallel regions or higher level regions with exit states, thus in that case it needs to be evaluated. The value μ_{s_i} of a state s_i in a region r_j can be evaluated using Equation 6.2.

$$\mu_{s_i} = \int_{D_{\tau_1 < \tau_2}} \gamma_{s_i}(\tau_1) f_{r_j}^{exit}(\tau_2) d\overline{\tau} \tag{6.2}$$

Where $\gamma_{s_i}(t)$ is the probability to be absorbed in state s_i at time t given that the region r_j is entered a time $t = 0$, $f_{r_j}^{exit}$ is exit density evaluated from the distribution $F_{r_j}^{exit}(t)$, $\overline{\tau} =< \tau_1, \tau_2 >$ and $D_{\tau_1 < \tau_2}$ is the joint domain of the two functions restricted to $\tau_1 < \tau_2$, thus where the absorption occurs before any exit.

6.2.4 Evaluation of probabilities to exit from border points

Exit states on the border allow us to define different successors of a composite state, based on which region completes first. Consider the composite state s_i shown in the left part of Figure 6.4. If region r_1 exits first, the successor state will be s_j, and if region r_2 exits first, the successor state will be s_k.

Evaluation of sojourn time distribution of a composite state with exits on the border is not different compared to exits not on the border. Its sojourn time continues to be the minimum between the sojourn times of its regions and this will be later used to evaluate the mean sojourn time of the state. The presence of exits on the border will instead affect evaluation of the distribution and probabilities in the parent region that contains such a composite state. From the parent region point of view, the composite state can be represented as an equivalent model as the one shown in the right part of Figure 6.4. In particular the probability of reaching a specific successor can be evaluated as the probability that one region is faster than the other. Moreover if a specific successor is reached, it means that time spent in state s_i is conditioned on the fact that a particular region was faster. Thus depending on the successor, time spent in s_i is different, and this is represented as two distinct states in which the sojourn time is conditioned to one region being faster than the other. According to this the evaluation of models including composite states with exits on borders requires to evaluate the probability that a region will be faster than all other parallel regions, evaluate the conditioned sojourn time distributions and replace such a composite state with the proposed equivalent model.

Consider a composite state s_i with exits on the border and R^i the set of its regions. Let $\alpha_{r_j}^{s_i}$ be the probability that region r_i is faster than all other $|R^i| - 1$ parallel regions, it can be evaluated according to Equation 6.3,

$$\alpha_{r_j}^{s_i} = \int_{D_{\tau_j first}} \frac{d\psi_{r_j}(\tau_j)}{d\tau_j} * \prod_{r \in R^i, i \neq j} \frac{d\psi_r(\tau_i)}{d\tau_i} d\overline{\tau} \tag{6.3}$$

where $D_{\tau_j first}$ is the joint domain of the $|R^i|$ densities restricted to the sub-region where $\tau_j < \tau_i \ \forall i$. Note that Equation 6.3 can be efficiently implemented replacing the right factor with the minimum and solving a two dimensioned integral instead of a multi dimensioned one. Finally, the sojourn time distribution for the state conditioned on having a region r_j that is faster, is given by $\psi_{r_j}(t)$.

6.2.5 Evaluation of the mean sojourn times

Now the mean sojourn times of each state can be evaluated through a bottom-up approach, taking into account exits in parallel regions. Specifically, the mean sojourn time for any top level state can be evaluated as the mean of its sojourn time distribution $\sigma_{s_i} = \int_0^\infty \Omega_{s_i}(t)dt$. Also if all parallel regions on the same and on a higher level have only final pseudostates, we can use the same formula. In all other cases, the possibility that an exit occurs in a parallel region must be considered. In the latter case, the mean sojourn time σ_{s_i} for a composite state s_i contained in a region r_j, can be evaluated as $\sigma_{s_i} = \int_0^\infty v_{r_j}^{s_i}(t) * F_{r_j}^{exit}(t)dt$, where $v_{r_j}^{s_i}(t)$ is the transient probability to be in state s_i, given that region r_j is entered at time $t = 0$ without considering exits in parallel regions, while $F_{r_j}^{exit}(t)$ takes into account possible exits in parallel regions.

6.2.6 Embedded DTMC and evaluation of steady state probabilities

Consider the top level *HSMP* θ and its transition matrix P. The embedded DTMC of such *HSMP* can be built considering only time point in which the top level state changes. Then the steady-state probabilities $\overline{u}$ of the embedded DTMC of the top state can be evaluated by the system of linear equations $\overline{u} = \overline{u}P$ with the additional constraint $|\overline{u}| = 1$. Finally steady-state probabilities can be evaluated using a top-down approach, considering the steady-state probabilities $\overline{u}$ of the embedded DTMC and weighting them by the mean sojourn times. The top-down approach starts evaluating steady-state probabilities for top level states as $\pi_{s_i} = \frac{u_{s_i} * \sigma_{s_i}}{\sum_{s_j \in S} u_{s_j} * \sigma_{s_j}}$ (Kulkarni, 2016). Then steady-state probabilities of a sub-state s_i contained in one of the regions of the parent state s_j can be evaluated as $\pi_{s_i} = \pi_{s_j} * \mu_{s_i} * \frac{\sigma_{s_i}}{\sigma_{s_j}}$.

It can be noted that in the early version of the technique presented in (Homm and German, 2016), μ_{s_i} was not considered and according to this, the technique didn't support the analysis of models where a parallel exit could prevent a sub-state to be reached.

6.2.7 Analysis with history states

The concept of history states was introduced in (Harel, 1987), as a convenient mechanism to keep track of a state configuration when a region was exited due to some parallel region. In the original formulation a history state could be of two types: *shallow*, that keeps track only of the top most level configuration or *deep* that keeps track of all sub-levels. In a non-Markovian system, the system state is not only given by the current location but also by the time elapsed in that state. According to this, we can define an additional subdivision of history states types: *Preemptive Repeat Different*(PRD), keeping memory of the location but not of time, and *Preemptive Resume* (PRS) keeping track also of time, thus when the configuration is restored also remaining time is restored. In this work we consider only PRD history states, while PRS history states will be studied in future work. For the sake of simplicity, in the following we also refer only to *shallow* history states, but similar concepts can be applied to *deep* history states.

If a history state is present like in state s_4 of Figure 6.1 when the state is entered, the last configuration when it was exited is resumed. In order to keep track of the last configuration when last exited, we propose to encode this information into the state space. In practice, states of the system need to be differentiated based on the history so as to keep track of it, thus allowing to diversify the future behavior based on such encoded information and start in the correct configuration when the state is visited again. An example is shown in Figure 6.5, where on the left a model with one history state and in the center the corresponding state space are shown. If s_1 is exited due to region r_2, we don't need to keep track of history since its exit state was reached. If instead s_1 is exited due to region r_1, we need to keep track of which was the last configuration of r_2, $s_{1,b}$ or $s_{1,c}$. According to this when s_2 is reached we need to keep track of this information. In the state space diagram in the center of Figure 6.5 nodes represent current overall configurations of the model considering both

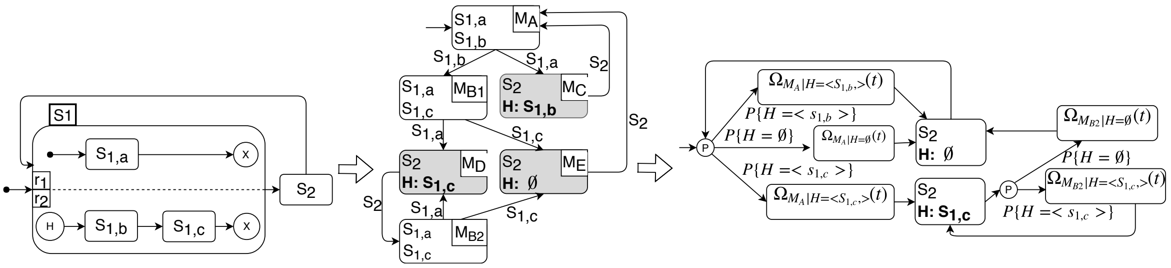


Figure 6.5: Example of model with history states and encoding of history in the state space

higher and lower level states, arcs represent the transition from one state to another and their label report the state whose sojourn time ending causes the change of configuration. In the upper right corner of nodes, a unique node name is assigned in order to simplify the explanation and in states that need to keep track of history, history information is represented in bold font. The initial node is M_A, where the system is in s_1 and in particular in state $s_{1,a}$ in region r_1 and in state $s_{1,b}$ in region r_2. If the first state to complete its sojourn is $s_{1,a}$, s_1 is exited reaching M_C where we need to keep memory of the state of r_2 that was $s_{1,b}$. Instead if $s_{1,b}$ completes first, M_{B1} is reached. From M_{B1} the system will in any case go to s_2 but with different history depending on which completes first. If $s_{1,c}$ completes first, node M_E is reached and next time r_2 will start from its default history state since it has reached its end state. If instead $s_{1,a}$ completes first, next time r_2 will start again from $s_{1,c}$ and this is encoded in the history. When this happens the initial configuration of s_2 is different from the one represented by node M_{B1}, because in M_{B1} the remaining sojourn time of state $s_{1,a}$ was conditioned on the fact that $s_{1,b}$ completes first. This is why node M_{B2} is different from M_{B1} even though the configurations represent the same locations, because in the latter the remaining time of $s_{1,a}$ has a different distribution.

In practice, when history states are present, we need to enrich the state space with additional information about history and in it must be observed that probabilities to go from M_{B1} to M_D or to M_E require to be evaluated as well as probabilities from M_A to M_C and from M_{B2} to M_D or M_E. Then the *SMP* of the top level of this model can be built, as shown in the right side of Figure 6.5 based on the above analysis. It is worth noting that the sojourn time in state s_1 is different if we are in M_{B2} or M_A and also depends on which state we exit. This is similar to what was seen in Section 6.2.4 for exits on the border, since depending on the successor, the sojourn time changes and thus can be handled following a similar approach. When entering in M_A, a pre-selection is needed like the one shown on the right of Figure 6.4 and then the sojourn time is different. In particular it decides which is the probability to have a particular history when the state will be exited, then the elapsed sojourn time will be conditioned on the exit having such a particular history. The probability to have a specific configuration when exiting a composite state with history states need to be evaluated. More generally, suppose to have n regions r_1, ..., r_n all having history states and suppose that r_j exits, since the evolutions of regions are independent by construction, the probability can be evaluated according to Equation 6.4.

$$P\{''H =< h_1, ..., h_{j-1}, h_{j+1}, h_n > ''\} = \int_0^{\infty} f_{exit}^{r_j}(t) \prod_{q=1...n, q \neq j} v_{h_q}^{r_q}(t)dt, \tag{6.4}$$

where h_q is the location of region r_q when the composite state was exited and thus this allows to evaluate the probability to have a particular history $\overline{h} =< h_1, ..., h_{j-1}, h_{j+1}, h_n >$ when the state was exited. Note that if a region has not a history state, we can simply remove it from the equation and consider instead the probability that the region was not exited before region r_j. Finally the sojourn time elapsed in the composite states with a history state need to be evaluated conditioned on the specific successor. Let s_k be the composite state with n regions r_1, ..., r_n all having history states. Given that the configuration when an exit occurs in r_q, was $\overline{h}$ and thus the successor will encode that history, the sojourn time distribution can be evaluated according to Equation 6.5.

$$\Omega_{s_k|H=<h_1,...,h_{j-1},h_{j+1},h_n>}(t) = \int_0^t \frac{f_{exit}^{r_j}(\tau) * \prod_{q=1...n,q\neq j} v_{h_q}^{r_q}(\tau)}{P\{''H =< h_1, ..., h_{j-1}, h_{j+1}, h_n > ''\}} d\tau \tag{6.5}$$

It should be noted that analyzing models with history states substantially increases the complexity of the analysis, in particular if the number of composite states with history states are more than one since it is required to encode the cartesian product of all possible histories causing an increase of the number of states and requiring to evaluate probabilities to have a particular history for all such states. However, on the positive side, the concept of history states requires an extension of the state space just on the level of states with a history state inside. If history states appear on different levels, the extension can also be considered separately. History state analysis can be combined with all other modeling elements of *HSMPs* presented in this chapter.

6.3 Experimental evaluation

6.3.1 Unavailability analysis of a fault tree

A Java numerical implementation of the approach has been developed so as to experimentally validate its correctness. The techniques can be implemented also as an analytical approach, but this requires a restriction of its applicability (Homm and German, 2016). The code is publicly available on *GitHub*[1].

[1] `https://github.com/biagimarco/hierarchicalSMP`

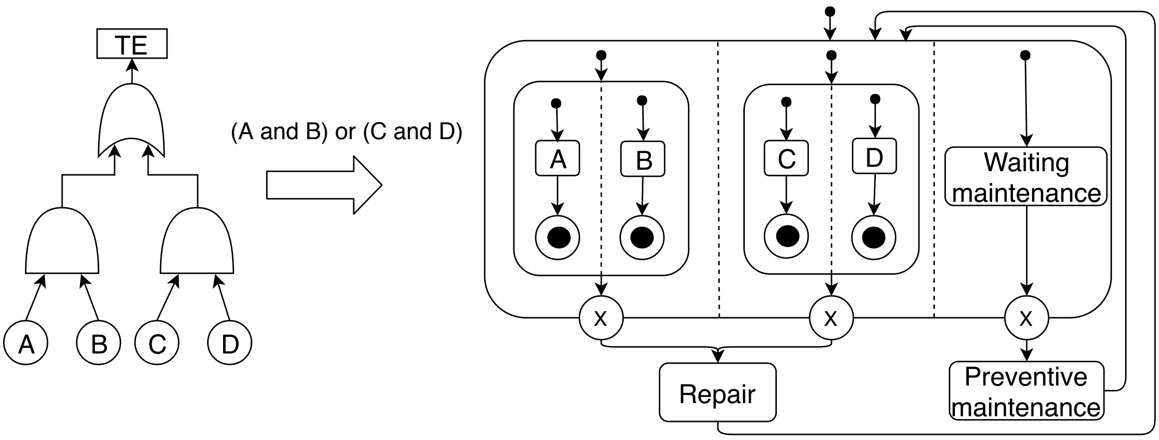


Figure 6.6: A fault tree and the equivalent *HSMP* model with repair and maintenance

As an example, we show how the analysis can be applied for the evaluation of steady-state probabilities of a repairable static fault tree with preventive maintenance (Ruijters and Stoelinga, 2015). Consider the static fault tree shown on the left side of Figure 6.6. The fault tree represents a system composed by four components A, B, C and D. If components A and B or components C and D fail at the same time, the whole system fails and becomes unavailable. In that case, a repair operation is performed in order to restore the initial state of the system. Since the unplanned repair operation is slow and more expensive, a preventive maintenance procedure has been adopted in order to periodically maintain the system and thus reduce unplanned repair operations. As shown in the right side of Figure 6.6, this system can be represented with an *HSMP* model. Specifically the fault tree is converted in parallel regions where *AND* gates are represented by final states and the *OR* gate is represented by exit states. It is worth noting that static fault trees with only *AND* and *OR* gates can always be modeled as an *HSMP*. A third parallel region models the maintenance period. Exits on the border are used in order to differentiate between unavailability due to a failure of the system and unavailability due to preventive maintenance. Distributions of the system are $F(\text{"A"}) = Exp(1/180)$, $F(\text{"B"}) = Exp(1/240)$, $F(\text{"C"}) = Exp(1/180)$, $F(\text{"D"}) = Exp(1/360)$, $F(\text{"Waiting maintenance"}) = Det(\xi)$, $F(\text{"Repair"}) = Unif(1,3)$, $F(\text{"Preventive maintenance"}) = Unif(0,1)$, where $Exp(\lambda)$ is the exponential distribution of rate λ, $Unif(a,b)$ is the uniform distribution with support $[a,b]$, $Det(\xi)$ is a deterministic time ξ. We want to measure the probability to find the system unavailable and the probability that the system is unavailable due to a preventive maintenance or due to a failure. Figure 6.7 shows these three measures, varying the value ξ of the time between two subsequent maintenance procedures. If the maintenance occurs too often, the probability to find the system unavailable is higher. Increasing ξ, that means decreasing the frequency of preven-

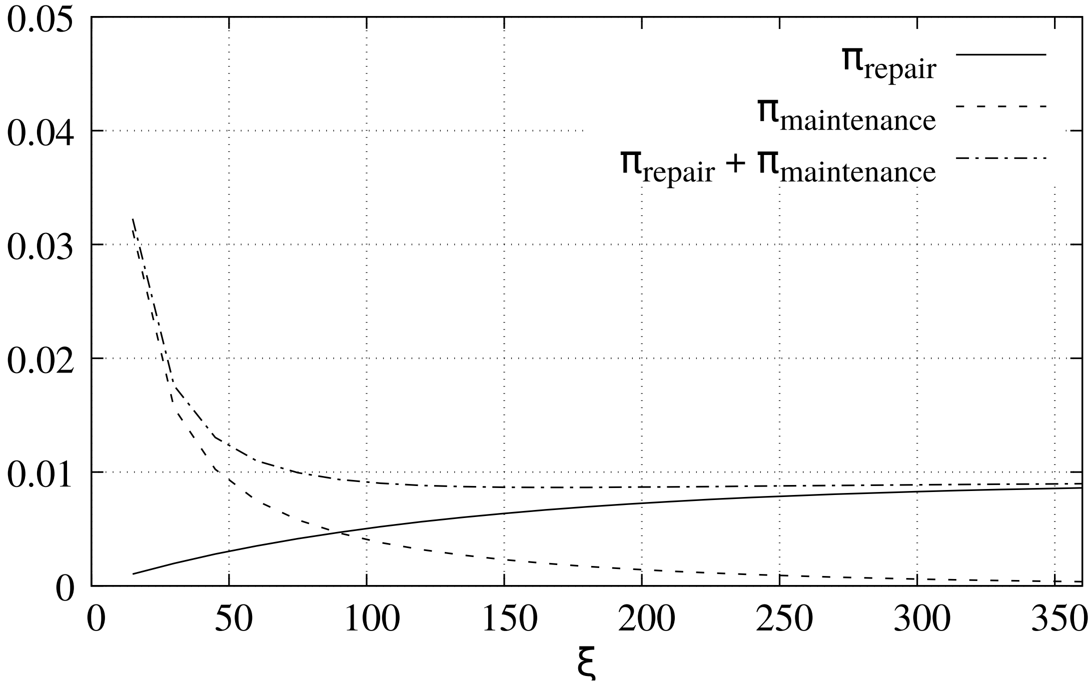


Figure 6.7: Steady-state probabilities that the fault tree is unavailable due to a failure, due to preventive maintenance, or both

N_r \ N_s	1	2	3	4	5	6
1	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$
	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$
2	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$
	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$\simeq 1s$	$\simeq 3s$
3	$< 1s$	$< 1s$	$< 1s$	< 1	$< 1s$	$< 1s$
	$< 1s$	$< 1s$	$\simeq 5s$	*N.D.*	*N.D.*	*N.D.*
4	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$	$< 1s$
	$< 1s$	$\simeq 10s$	*N.D.*	*N.D.*	*N.D.*	*N.D.*

Table 6.1: Evaluation times varying N_r and N_s. Upper rows with *HSMP* analysis, lower rows with regenerative analysis

tive maintenance, reduces the total probability to find the system unavailable but also increases the probability that the system is unavailable due to a failure and not due to a preventive maintenance.

In order to experimentally validate the approach, the same system has also been modeled as a Petri net and analyzed using the *Oris tool API* (Carnevali et al., 2011). Evaluated results match for all possible values of ξ.

6.3.2 Computational experience with composability

The strong point of this technique is compositionality, allowing to analyze a model without the need to build its whole state space. In the following a comparison with a technique that instead builds the whole state space is performed. The technique that we chose for comparison is the regenerative steady-state analysis based on stochastic state classes (Martina et al., 2016), since currently it is the only technique allowing the evaluation of steady-state probabilities of models with multiple concurrent non-Markovian timers and without the enabling restriction.
The experiment has been performed on a model having a top level composed by a single composite state with a self loop. The composite state consists of N_r parallel regions, each one composed by a sequence of N_s states leading to a final state. Thus the lower level model is composed by $N_r * (N_s + 1)$ states. Sojourn times are all uniform distributions with support $[0, 1]$.

The times required to analyze the model with varying N_r and N_s are reported in Table 6.1. Experiments were performed on a single core of a 2.20 GHz Intel i5-5200U with 8 GB RAM, and each single run was performed with a timeout of 2 minutes. Increasing the number of regions N_r increases the concurrency degree of the model and thus the regenerative analysis needs to evaluate the state space considering all possible orders in which the transitions can be executed. As one can see, with $N_r = 3$ and $N_r = 4$, 2 minutes are no more sufficient to analyze the model with the regenerative analysis while with the hierarchical approach it requires less than 1 second. The experiment gives an impression of the advantage of this technique when applied to highly concurrent models.

Chapter 7

Conclusions

In this thesis we have been dealing with the problem of analysis of non-Markovian models, with the objective to reduce the limitations of current analysis technique for such models.

We have described some of the existent analysis techniques and how their integration can be driven by exploiting non-deterministic analysis. This approach can automatically choose the best technique to analyze a model and also be used to analyze models that were not analyzable before. Additionally, for models for which no exact solution can be derived, we have presented a novel approach based on the partial enumeration of stochastic state classes, which are iteratively explored according to a heuristic criterion based on the probability that a regeneration is reached. In this way, the approximation is limited to the kernel entries of a subset of regenerative epochs, and transient probabilities of markings can be safely and accurately approximated. Experimental results show that the heuristic-based approximate analysis provides accurate results while maintaining a moderate computational cost, suggesting that approximation could be used also for regenerative epochs characterized by finite stochastic transient trees, in order to reduce the number of stochastic state classes needed to compute the kernels.

Moreover, we presented a novel approach to compute a closed-form expression of the equilibrium distribution of MRGPs. The approach leverages the calculus of stochastic state classes and notably this constitutes a basis to compute survivability measures for MRGP (Heegaard and Trivedi, 2009).

Finally we have shown how a different formalism called HSMP, inspired by UML state machines (Group, 2018), can be analyzed leveraging its hierarchical structure with a compositional approach, enabling the development of an approach that does not need to generate the whole state space. In particular we have extended the work of (Homm and German, 2016), improving the analysis technique by removing some of its limitations and extending it so as to boost its modeling power in the direction to be more similar to that of UML state machines. Specifically, composite states now have no limitations related to exit states, also the possibility to have exit states on the border of composite states has been introduced, allowing to define different successors of such states based on which region finishes first. Additionally, the concept of history states has been added, enabling the definition of more condensed models. A future development of this approach will be to further extend the technique so as to reduce the gap between the modeling power of UML state machines. For example composite states mixing regions with final states and exit states may be allowed, or final and exit states in the same region may be introduced. Another direction would be to study how to analyze models with PRS history states introduced in this work.

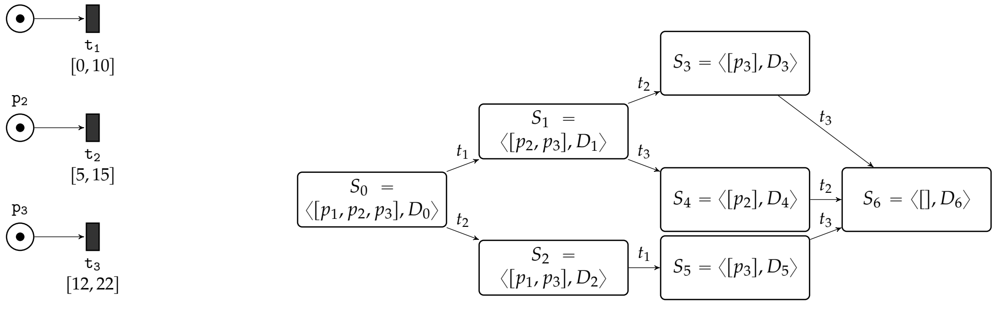
Appendix A

State Class Graph evaluation example

We report a partial example of evaluation of an SCG, in order to show how the analysis can be practically performed, applied to the TPN of Figure A.1a. The resulting SCG is shown in Figure A.1b, where the marking of each class is explicitly reported. The support is instead not explicitly shown for readability reasons. Initially, in the state class S_0 all timers are newly enabled and according to this the support can be easily defined by the hyper-rectangle with constraints defined according to the static interval $EFT(t) \leqslant \bar{\tau}(t) \leqslant LFT(t)$, as shown in Equation A.1.

$$D_0 = \begin{cases} 0 \leqslant \bar{\tau}(t_1) \leqslant 10 \\ 5 \leqslant \bar{\tau}(t_2) \leqslant 15 \\ 12 \leqslant \bar{\tau}(t_3) \leqslant 22 \end{cases} \tag{A.1}$$

Note that for simplicity, the support is not expressed in the form of Equation 2.5, but in an aggregated and simplified form. In practice, each point of the hyper-rectangle represent a possible vector of times to fire of the three enabled transitions t_1, t_2 and t_3. Using the ground reference $\bar{\tau}(t_*)$ and explicating constraints between timers, we


p1
t1
[0, 10]
p2
t2
[5, 15]
p3
t3
[12, 22]
$S_0 = \langle [p_1, p_2, p_3], D_0 \rangle$
$S_1 = \langle [p_2, p_3], D_1 \rangle$
$S_2 = \langle [p_1, p_3], D_2 \rangle$
$S_3 = \langle [p_3], D_3 \rangle$
$S_4 = \langle [p_2], D_4 \rangle$
$S_5 = \langle [p_3], D_5 \rangle$
$S_6 = \langle [], D_6 \rangle$
t_1
t_2
t_3

(a) TPN model

(b) Resulting SCG

Figure A.1: Example of analysis applied to a TPN

can also rewrite it as shown in Equation A.2.

$$D_0 = \begin{cases} 0 \leqslant \bar{\tau}(t_1) - \bar{\tau}(t_*) \leqslant 10 \\ 5 \leqslant \bar{\tau}(t_2) - \bar{\tau}(t_*) \leqslant 15 \\ 12 \leqslant \bar{\tau}(t_3) - \bar{\tau}(t_*) \leqslant 22 \\ -5 \leqslant \bar{\tau}(t_2) - \bar{\tau}(t_1) \leqslant 15 \\ 2 \leqslant \bar{\tau}(t_3) - \bar{\tau}(t_1) \leqslant 22 \\ -3 \leqslant \bar{\tau}(t_3) - \bar{\tau}(t_2) \leqslant 17 \end{cases} \tag{A.2}$$

This domain implies that only t_1 or t_2 can be the first transition to fire, because $b_{13} = -2 < 0$ (which is the negative of the left element of the fifth inequality of Equation A.2) implies that t_3 can't fire before t_1. In this case, this can be easily seen also from the model since t_3 is forced to fire after $t = 12$ while t_1 is forced to fire not after $t = 10$. Another point of view is that reducing the support to the subspace where $t_3 < t_1$, the resulting state space is empty since no solutions of this type are admitted by the set of constraints. According to this, if follows that state class S_0 has two successors, S_1 if t_1 fires and S_2 if t_2 fires, as show in Figure A.1b. The evaluation of the marking of such state classes can be done following the transition rules described in Section 2.1.2, while the evaluation of the supports is more elaborate, and requires instead the application of Proposition 1. In particular, domains D_1 and D_2 are shown in Equations A.3 and A.4, respectively.

$$D_1 = \begin{cases} 0 \leqslant \bar{\tau}(t_2) - \bar{\tau}(t_*) \leqslant 15 \\ -3 \leqslant \bar{\tau}(t_3) - \bar{\tau}(t_2) \leqslant 17 \\ 2 \leqslant \bar{\tau}(t_3) - \bar{\tau}(t_*) \leqslant 22 \end{cases} \tag{A.3}$$

$$D_2 = \begin{cases} 0 \leqslant \bar{\tau}(t_1) - \bar{\tau}(t_*) \leqslant 5 \\ 2 \leqslant \bar{\tau}(t_3) - \bar{\tau}(t_1) \leqslant 17 \\ 2 \leqslant \bar{\tau}(t_3) - \bar{\tau}(t_*) \leqslant 17 \end{cases} \tag{A.4}$$

The same passages need to be done also for all other successors allowing to build the SCG of the model shown in Figure A.1b.

Appendix B

Publications

Journal papers

1. **M. Biagi**, L. Carnevali, M. Paolieri, E. Vicario, "Performability evaluation of the ERTMS/ETCS - Level 3", *Transportation Research Part C: Emerging Technologies*, pages: 314-336, 2017. **Candidate's contributions**: carried out researches regarding the standard ERTMS/ETCS, theoretical analyses and experimentations

2. **M. Biagi**, L. Carnevali, F. Tarani, E. Vicario, 'Model-based quantitative evaluation of repair procedures in gas distribution networks", *ACM Transactions on Cyber Physical Systems*, 2018. **Candidate's contributions**: experimentation including sensitivity analysis

Peer reviewed conference papers

1. **M. Biagi**, L. Carnevali, T. Papini, M. Paolieri, E. Vicario "Exploiting non- deterministic analysis in the integration of transient solution techniques for Markov Regenerative Processes", *International Conference on Quantitative Evaluation of SysTems (QEST)*, pages: 20-35, 2017. **Candidate's contributions**: initial idea, theoretical analyses and experimentations

2. **M. Biagi**, L. Carnevali, E. Vicario, M. Paolieri "An introduction to the ORIS tool", *Proceedings of the 11th EAI International Conference on Performance Evaluation Methodologies and Tools(VALUETOOLS)*, pages: 9-11, 2017. **Candidate's contributions**: improvements to the tool

3. **M. Biagi**, L. Carnevali, F. Santoni, E. Vicario "Hospital Inventory Management Through Markov Decision Processes@ runtime", *International Conference on Quantitative Evaluation of SysTems (QEST)*, pages: 87-103, 2018. **Candidate's**

contributions: part of the theoretical analyses and part of the experimentation on Hidden Markov Models

4. **M. Biagi**, L. Carnevali, K. Tadano, E. Vicario "Evaluation of stochastic bounds on the remaining completion time of products in a buffered sequential workflow", *IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2018. **Candidate's contributions**: theoretical analyses and experimentation

Workshop papers

1. **M. Biagi**, L. Carnevali, M. Paolieri, F. Patara, E. Vicario, "A stochastic model-based approach to online event prediction and response scheduling", *European Workshop on Performance Engineering (EPEW)*, pages: 32–47, 2016. **Candidate's contributions**: part of the theoretical analyses and experimentations
2. **M. Biagi**, L. Carnevali, T. Papini, K. Tadano, E. Vicario "An inspection based compositional approach to the quantitative evaluation of assembly lines", *European Workshop on Performance Engineering (EPEW)*, pages: 152–166, 2017. **Candidate's contributions**: part of the theoretical analyses and experimentations
3. **M. Biagi**, R. German, E. Vicario "Extending the steady state analysis of hierarchical semi-Markov processes with parallel regions", *European Workshop on Performance Engineering (EPEW)*, 2018. **Candidate's contributions**: idea, theoretical analyses and experimentation

Papers under review

1. **M. Biagi**, L. Carnevali, M. Paolieri, F. Patara, E. Vicario "A continuous-time model-based approach for activity recognition in pervasive environments", *IEEE transactions on human-machine systems*, 2018. **Candidate's contributions**: offline analysis, formalization of the technique and experimentation
2. **M. Biagi**, L. Carnevali, M. Paolieri, E. Vicario "The ORIS Tool: Quantitative Evaluation of Non-Markovian", *IEEE transactions on software engineering*, 2018. **Candidate's contributions**: improvements to the tool and definition of a case study

Bibliography

Amparore, E. G., Buchholz, P., and Donatelli, S. (2014). A structured solution approach for Markov regenerative processes. In *International Conference on Quantitative Evaluation of Systems*, pages 9–24. Springer.

Ballarini, P., Bertrand, N., Horváth, A., Paolieri, M., and Vicario, E. (2013). Transient analysis of networks of stochastic timed automata using stochastic state classes. In *International Conference on Quantitative Evaluation of Systems*, pages 355–371. Springer.

Berthomieu, B. and Diaz, M. (1991). Modeling and verification of time dependent systems using time Petri nets. *IEEE Transactions on Software Engineering*, 17(3):259–273.

Biagi, M., Carnevali, L., Paolieri, M., Papini, T., and Vicario, E. (2017). Exploiting non-deterministic analysis in the integration of transient solution techniques for Markov regenerative processes. In *International Conference on Quantitative Evaluation of Systems*, pages 20–35. Springer.

Biagi, M., Vicario, E., and German, R. (2018). Extending the steady state analysis of hierarchical semi-Markov processes with parallel regions. In *European Workshop on Performance Engineering*. Springer.

Bolch, G., Greiner, S., De Meer, H., and Trivedi, K. S. (2006). *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. John Wiley & Sons.

Bucci, G., Carnevali, L., Ridi, L., and Vicario, E. (2010). Oris: a tool for modeling, verification and evaluation of real-time systems. *STTT*, 12(5):391–403.

Buchholz, P. and Telek, M. (2013). Rational automata networks: a non-Markovian modeling approach. *INFORMS Journal on Computing*, 25(1):87–101.

Budde, C. E., D'Argenio, P. R., and Hermanns, H. (2015). Rare event simulation with fully automated importance splitting. In *European Workshop on Performance Engineering*, pages 275–290. Springer.

Carnevali, L., Grassi, L., and Vicario, E. (2009). State-density functions over DBM domains in the analysis of non-Markovian models. *IEEE Transactions on Software Engineering*, 35(2):178–194.

Carnevali, L., Ridi, L., and Vicario, E. (2011). Sirio: A framework for simulation and symbolic state space analysis of non-Markovian models. In *2011 Eighth International Conference on Quantitative Evaluation of SysTems*, pages 153–154. IEEE.

Choi, H., Kulkarni, V. G., and Trivedi, K. S. (1994). Markov regenerative stochastic Petri nets. *Performance Evaluation*, 20(1-3):337–357.

Ciardo, G., German, R., and Lindemann, C. (1994). A characterization of the stochastic process underlying a stochastic Petri net. *IEEE Transactions on Software Engineering*, 20(7):506–515.

D'Argenio, P. R., Lee, M. D., and Monti, R. E. (2016). Input/output stochastic automata. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 53–68. Springer.

German, R. (2000). *Performance analysis of communication systems with non-Markovian stochastic Petri nets*. John Wiley & Sons, Inc.

German, R. and Lindemann, C. (1994). Analysis of stochastic Petri nets by the method of supplementary variables. *Performance Evaluation*, 20(1-3):317–335.

German, R., Logothetis, D., and Trivedi, K. S. (1995). Transient analysis of Markov regenerative stochastic Petri nets: A comparison of approaches. In *Petri Nets and Performance Models, 1995., Proceedings of the Sixth International Workshop on*, pages 103–112. IEEE.

Gnesi, S., Latella, D., and Massink, M. (2000). A stochastic extension of a behavioural subset of uml statechart diagrams. In *High Assurance Systems Engineering, 2000, Fifth IEEE International Symposim on. HASE 2000*, pages 55–64. IEEE.

Group, O. M. (2018). Unified modeling language specification version 2.5.1.

Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of computer programming*, 8(3):231–274.

Heegaard, P. E. and Trivedi, K. S. (2009). Network survivability modeling. *Computer Networks*, 53(8):1215–1234.

Homm, D. and German, R. (2016). Analysis of hierarchical semi-Markov processes with parallel regions. In *International GI/ITG Conference on Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*, pages 92–106. Springer.

Horváth, A., Paolieri, M., Ridi, L., and Vicario, E. (2012). Transient analysis of non-Markovian models using stochastic state classes. *Performance Evaluation*, 69(7-8):315–335.

Ibe, O. C. and Trivedi, K. S. (1990). Stochastic Petri net models of polling systems. *IEEE Journal on Selected areas in Communications*, 8(9):1649–1657.

Jansen, D., Hermanns, H., and Katoen, J.-P. (2003). A QoS-oriented extension of UML statecharts. In *International Conference on the Unified Modeling Language*, pages 76–91. Springer.

Jansen, D. N., Hermanns, H., and Katoen, J.-P. (2002). A probabilistic extension of UML statecharts. In *International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems*, pages 355–374. Springer.

Jurdzinski, M., Groesser, M., Baier, C., Menet, Q., Brihaye, T., Bouyer, P., and Bertrand, N. (2014). Stochastic timed automata. *Logical Methods in Computer Science*, 10.

Kulkarni, V. G. (2016). *Modeling and analysis of stochastic systems*. Chapman and Hall/CRC.

Laprie, J.-C. (1992). Dependability: Basic concepts and terminology. In *Dependability: Basic Concepts and Terminology*, pages 3–245. Springer.

Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., and Hoffmann, M. (2014). Industry 4.0. *Business & Information Systems Engineering*, 6(4):239–242.

Lee, J., Bagheri, B., and Kao, H.-A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3:18–23.

Lindemann, C. and Thümmler, A. (1999). Transient analysis of deterministic and stochastic Petri nets with concurrent deterministic transitions. *Performance Evaluation*, 36:35–54.

Liu, Y. and Trivedi, K. S. (2006). Survivability quantification: The analytical modeling approach. *International Journal of Performability Engineering*, 2(1):29–44.

Logothetis, D., Trivedi, K. S., and Puliafito, A. (1995). Markov regenerative models. In *Computer Performance and Dependability Symposium, 1995. Proceedings., International*, pages 134–142. IEEE.

Martina, S., Paolieri, M., Papini, T., and Vicario, E. (2016). Performance evaluation of fischer's protocol through steady-state analysis of Markov regenerative processes. In *Modeling, Analysis and Simulation of Computer and Telecommunication*

Systems (MASCOTS), 2016 IEEE 24th International Symposium on, pages 355–360. IEEE.

Paolieri, M., Horváth, A., and Vicario, E. (2016). Probabilistic Model Checking of Regenerative Concurrent Systems. *IEEE Transactions on Software Engineering*, 42(2):153–169.

Rubino, G. and Tuffin, B. (2009). *Rare event simulation using Monte Carlo methods*. John Wiley & Sons.

Ruijters, E. and Stoelinga, M. (2015). Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools. *Computer science review*, 15:29–62.

Telek, M. and Horváth, A. (2001). Transient analysis of age-mrspns by the method of supplementary variables. *Performance Evaluation*, 45(4):205–221.

Trivedi, K. S. and Bobbio, A. (2017). *Reliability and Availability Engineering: Modeling, Analysis, and Applications*. Cambridge University Press.

Vicario, E. (2001). Static analysis and dynamic steering of time-dependent systems. *IEEE Transactions on Software Engineering*, (8):728–748.

Vicario, E., Sassoli, L., and Carnevali, L. (2009). Using stochastic state classes in quantitative evaluation of dense-time reactive systems. *IEEE Transactions on Software Engineering*, 35(5):703–719.

Zimmermann, A. (2012). Modeling and evaluation of stochastic Petri nets with timenet 4.1. In *Performance evaluation methodologies and tools (VALUETOOLS), 2012 6th international conference on*, pages 54–63. IEEE.