Technical University of Denmark

DTU

Logics and Models for Stochastic Analysis Beyond Markov Chains

Zeng, Kebin; Nielsen, Bo Friis; Nielson, Flemming

Publication date:
2012

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

DTU Library
Technical Information Center of Denmark

Logics and Models for Stochastic Analysis Beyond Markov Chains

Kebin Zeng
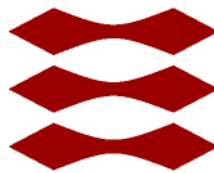
Summary (English)

Within the last twenty years, logics and models for stochastic analysis of information systems have been widely studied in both theory and practice. The quantitative properties, such as performance and reliability, are evaluated over discrete–time and continuous–time Markov chains. This thesis lifts the stochastic analysis techniques from the class of Markov chains to the more general classes of stochastic processes having PHase–type (PH) distributions and Matrix–Exponential (ME) distributions, such as a Markov renewal process with ME kernels that cannot be formulated as a Markov process with finite or countable state space.

PH distributions are known for many explicit analytic properties, such that systems having PH distributed components can still be formulated as Markov chains. This thesis presents several results related to PH distributions. We first show how to use the explicit analytic form of discrete PH distributions as computational vehicle on measuring the performance of concurrent wireless sensor networks. Secondly, choosing stochastic process algebras as a widely accepted formalism, we study the compositionality of continuous PH distributions in order to support modelling concurrent stochastic systems having PH representations as building blocks. At last, we consider discrete–time point processes having PH distributed interarrival times with multiple marks, we propose time-lapse bisimulation, a state-based characterisation of the equivalence relation between the point processes. We clarify that time-lapse bisimulation is a new contribution to the existing bisimulation family, which captures probabilistic behaviour over time for labelled discrete–time Markov chains.

ME distributions is a strictly larger class than PH distributions, such that many

results from PH distributions also are valid for ME distributions. ME distributions have a very appealing property, called minimality property: generally a ME representation of a PH distribution will be of lower dimension than PH representations, and one can always find a ME representation with the minimal dimension. However, because of the generality of ME distributions, we have to leave the world of Markov chains. To support ME distributions with multiple exits, we introduce a multi-exits ME distribution together with a process algebra MEME to express the systems having the semantics as Markov renewal processes with ME kernels. The most appealing feature is that all the components before and after compositions are secured to have a minimal state space representation. To support quantitative verification, we also propose stochastic model checking algorithms for our problem.

Summary (Danish)

Gennem de sidste 20 år er logik og modeller for stokastisk analyse af informationssystemer blevet studeret bredt både i teori og praksis. De kvantitative egenskaber såsom ydeevne og pålidelighed er blevet bedømt ved hjælp af diskrete og kontinuerte Markovkæder. Denne afhandling overfører stokastiske analyseteknikker for Markovkæder til brug for den mere generelle klasse af Markovprocesser bestående af Fasetype (PH) fordelinger og MatrixEksponentielle (ME) fordelinger. Således kan Markovprocesser med ME fordelte tider ikke formuleres som en Markovproces med hverken endeligt eller tælleligt tilstandsrum.

PH fordelinger er kendt for at have mange eksplicitte, analytiske egenskaber. Det medfører, at systemer, der har PH fordelte komponenter, kan formuleres som Markovkæder. Denne afhandling præsenterer flere resultater relateret til PH fordelinger. Vi viser først, hvordan en eksplicit, analytiske form af diskrete PH fordelinger kan bruges som beregningsmæssigt værktøj til at måle ydeevnen af trådløse, parallelle sensornetværk.

Ved hjælp af stokastisk procesalgebra, som er et bredt accepteret paradigme, undersøger vi dernæst kompositionaliteten af kontinuerte PH fordelinger. Dette gøres for at redegøre for modellering af parallelle, stokastiske systemer, der bygger på PH repræsentationer. Til slut betragter vi diskrete mærkede punktprocesser med PH fordelte mellemankomsttider. Vi foreslår *time-lapse bisimulation*, som er en tilstandsbaseret karakterisering af lighedsrelationen mellem punktprocesserne. Vi redegør for, hvordan *time-lapse bisimulation* er et nyt bidrag til den eksisterende samling af bisimlationer, der beskriver den probabillistiske opførsel over tid af mærkede, diskrete Markovkæder.

Selvom klassen af ME fordelinger er strengt større end klassen af PH fordelinger, kan mange af de resultater, der gælder for PH fordelinger, overføres til klassen af ME fordelinger. ME fordelinger har en særlig tiltalende egenskab nemlig minimalitetsegenskaben. Denne betyder, at en ME repræsentation af en PH fordeling generelt vil være af lavere dimension end PH repræsentationen, og der eksisterer altid en ME repræsentation med minimal dimension. Dog tvinger den generelle formulering af ME fordelinger os til at forlade Markovkædernes verden. For at kunne håndtere ME fordelinger med flere udgange introducerer vi en multi–exits ME fordeling. Ved brug af MEME procesalgebra er vi nu i stand til at formulere systemer med samme semantik som en Markov–fornyelsesproces med ME kerne. Fordelen ved dette er, at samtlige komponenter både før og efter kompositioner kan repræsenteres ved en minimal repræsentation. For at kunne udføre kvantitativ efterprøvning præsenterer vi foreslået en stokastisk algoritme til modeltjek for vores problem.

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in partial fulfilment of the requirements for acquiring a Ph.D. degree in Informatics. The Ph.D. study has been carried out under the supervision of Associate Professor Bo Friis Nielsen (Technical University of Denmark) and Professor Flemming Nielson (Technical University of Denmark) in the period from November 2009 to October 2012, and was partially supplied by MT-LAB, a VKR Centre of Excellence.

This thesis studies logics and models for analysing stochastic aspects of information systems in a general setting. Particularly, we rely on two classes of probability distributions, those of phase–type distributions and matrix–exponential distributions, and their generalised stochastic processes.

This thesis consists of several scientific contributions with my excellent research collaborators, including my two supervisors and Professor Holger Hermanns from Saarland University. The scientific work in this thesis are based on our published work [EZN11], the work under submission [ZENN12], and the unpublished work [ZNN12, ZNNH12].

Lyngby, 31-October-2012

Kebin Zeng

Acknowledgements

Working on such an interdisciplinary PhD project has been a wonderful and memorable experience. I am enjoyable of being in academics, learning how to perform independent research, writing papers and proposals, giving talks, working in a group, staying up the last minute of deadline, and relaxing in supervisors' jokes...

I would like at first to thank God for letting me take such an interesting research topic, choosing nice colleagues as angels around, and leading me through difficulties. You are the one who let me finish the degree, I will keep on trusting you in future.

This thesis would not have been possible without the help, support and patience of my principle supervisor, Associate Professor Bo Friis Nielsen from both academic and personal aspects. I would like to give special thanks to his supervision and care for introducing me to the field of stochastic processes. I also want to give special thanks to my co-supervisor, Professor Flemming Nielson, for his kind advices and insights to the formal methods in theoretical computer science. From you, I have learned not only the knowledge but also rigorous academic spirit. I would like to thank Professor Holger Hermanns at Saarland university in Germany for hosting my research visit and offering insightful comments. The external collaboration with you has became one of the most important contributions in this thesis.

I would like to thank my colleagues and friends in language-based technology group at DTU informatics, statistics group at DTU informatics, and dependable group at Saarland university. I am particular thankful to my officemates Luz

Symbols and Abbreviations

$\mathbf{e}$	a column vector of ones
$\mathbf{e}_i$	a unit column vector with i-th element equals 1
$\mathbb{E}$	the expectation
$f(\cdot)$	the (joint) probability density/mass function
$F(\cdot)$	the (joint) cumulative distribution function
$G(\cdot)$	the probability generating function
$H(\cdot)$	the Laplace–Stieltjes transform
$\mathbb{N}$	the set of all natural numbers excluding 0
$\mathbb{N}_0$	the set of all natural numbers including 0
$\mathbb{P}$	the probability
$\Re$	the set of all real numbers
$\Re_{>0}$	the set of positive real numbers
$\Re_{\geq 0}$	the set of non-negative real numbers
$\varphi,\ \psi,\ \Phi,\ \Psi,\ \ldots$	the logical formulas
$\boldsymbol{\alpha},\ \boldsymbol{\beta},\ \boldsymbol{\gamma},\ \ldots$	the initial row vectors
α_i	the i-th element of a vector $\boldsymbol{\alpha}$
$\boldsymbol{\pi},\ \boldsymbol{\pi}_1,\ \boldsymbol{\pi}_2,\ \ldots$	the probability vectors
$H,\ W,\ X,\ \ldots$	the random variables
$\mathbf{A},\ \mathbf{P},\ \mathbf{Q},\ \mathbf{S},\ \ldots$	the matrices
$\mathrm{A}_{i,j}$	the (i,j)-th element of a matrix $\mathbf{A}$
$\mathbf{A}^{tr}$	the transpose of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	the inverse of matrix $\mathbf{A}$
$\mathbf{I}$	the identity matrix
$\mathbf{0}$	the zero matrix

CPH	Continuous PHase–type
CSL	Continuous Stochastic Logic
CTMC	Continuous–Time Markov Chain
DPH	Discrete PHase–type
DTMC	Discrete–Time Markov Chain
EXP	Exponential
LMAC	Lightweight Medium ACess
LST	Laplace–Stieltjes Transform
MAP	Markovian Arrival Process
ME	Matrix–Exponential
MECPH	Multi–Exits Continuous PHase–type
MEME	Multi–Exits Matrix–Exponential
MMAP	Marked Markovian Arrival Process
MRAP	Marked Rational Arrival Process
MRP	Markov Renewal Process
PH	PHase–type
PHPA	PHase–type Process Algebra
PP	Point Process
RAP	Rational Arrival Process
SMC	Semi-Markov Chain
SMP	Semi-Markov Process
SPA	Stochastic Process Algebra
WSN	Wireless Sensor Network

Contents

Introduction

In the last twenty years, there has been substantial work on the stochastic extension of labelled transition systems for analysing quantitative properties, such as performance, reliability, and timeliness. Through stochastic models and logical expressions, quantitative properties are evaluated and verified for computer systems and networks. The techniques for exploring transient and stationary behaviour of Markov chains have been successfully and widely applied. In Markov chains, the distributions of the time for jumping from one state to another correspond to (discrete–time) Geometric distributions or (continuous–time) Exponential distributions, which are memoryless probability distributions. Departing from memoryless probability distributions, this thesis focuses on developing the formal verification techniques for more general and advanced probability distributions and their related stochastic processes.

Phase–type (PH) distributions were considered first in [Neu75, Neu81], and are defined as the distributions of the time until absorption in a Markov chain with a finite number of transient states and one absorbing state. Exponential, Erlang, Hypo-exponential, Hyper-exponential and Coxian distributions are examples of Continuous PH (CPH) distributions. Amazingly, CPH distributions can be used to approximate any kind of probability distributions on $[0, \infty)$ [JT88]. In Chapter 3, we present two contributions related to PH distributions :

- Discrete PH (DPH) distributions are applied as a computational vehicle

in a practical case study on the LMAC protocol for concurrent wireless sensor networks [EZN11].

- A theoretical survey on the compositionality of CPH distributions is formalised into a process algebra, called phase–type process algebra, to support compositional reasoning in large-scale systems with CPH distributed random variables as primitives [ZNN12].

A Point Process (PP) is a stochastic process such that its realisation consists of a set of isolated points in time space [DVJ88]. A Markovian Arrival Process (MAP) is a PP with PH distributed interarrival times [Neu79, Luc91]. The MAPs are further extended to allow multiple marks/labels, thus defining marked MAPs (MMAP) [HN98]. Bisimulation relations allow one to reduce a system to an equivalent but smaller system, which is obtained by replacing each state in a system by its bisimulation equivalence class. Such a state aggregation technique is commonly used as a preprocessing step for model checking. This thesis introduces the notions of stochastic equivalence between DPHs and discrete–time MMAPs. Inspired from stochastic equivalence and a weakening of current family of bisimulations, we define a new bisimulation relation, named time-lapse bisimulation, on labelled discrete–time Markov chains to capture probabilistic timed behaviour. Time-lapse bisimulation counts the number of internal actions on traces, which is coarser than strong bisimulation and not comparable with weak bisimulation. The details of the equivalence relations are presented in Chapter 4 and submitted in [ZENN12].

The class of Matrix–Exponential (ME) distributions [AB97, BN03] includes and generalises CPH distributions, with the dimensionality of representations as a crucial merit. A minimal ME representation of a CPH distribution will be of lower order, or of the same order, than the corresponding CPH representations [BN03], and an efficient algorithm exists to find a minimal representation of ME distributions. Because of lacking stochastic interpretation in terms of Markov chains, ME distributions have got much less attention than PH distributions in stochastic analysis. Extending ME distributions with multiple exits, Chapter 5 introduces the minimality and compositionality of multi–exits ME distributions, together with a process calculus MEME having Markov Renewal Process (MRP) semantics, such that its corresponding stochastic model checking algorithms are also clarified.

This thesis is organised in the following manner. In Chapter 2, we introduce interesting probability distributions and stochastic processes, ranging from discrete–time processes to continuous–time processes. In Chapter 3, we present two contributions for PH distributions, one for discrete PH and one for continuous PH. We first present a case study applying DPH distributions for concurrent wireless sensor networks in Section 3.1, and then illustrate the compositionality

results for CPH distributions in phase–type process algebra in Section 3.2. In Chapter 4, we develop a new bisimulation relation called time-lapse bisimulation on labelled discrete–time Markov chains derived from the notions of stochastic equivalence. In Chapter 5, we clarify the stochastic model checking techniques for Markov renewal processes with ME kernels. We conclude in Chapter 6.

Stochastic Processes

In this chapter, we introduce interesting stochastic processes for both discrete–time and continuous–time. The related stochastic processes are defined using random variables with matrix representations. Unless otherwise specified, let $\mathbb{N}$ denote the set of natural numbers and let $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. Throughout the thesis, we use bold letters for vectors and matrices, α_i for the i-th element of a vector $\boldsymbol{\alpha}$, and $\mathrm{P}_{i,j}$ for the (i,j)-th element of a matrix $\mathbf{P}$. With appropriate dimensions, $\mathbf{0}$ denotes a zero matrix, $\mathbf{I}$ denotes an identity matrix, $\mathbf{e}$ denotes a column vector of ones, and $\mathbf{e}_i$ be a unit column vector such that the i-th element is 1 and the remaining elements are 0. For any matrix $\mathbf{A}$, $\mathbf{A}^{tr}$ is the transpose of $\mathbf{A}$.

Let $\{X_n\}_{n \in T}$ be a family of random variables, which take values in some countable set E, indexed by some set T. We say the process is a discrete–time process if $T = \mathbb{N}_0$, and a continuous–time process if $T = \Re_{\geq 0}$.

2.1 Discrete–Time Processes

2.1.1 Discrete–Time Markov Chains

A Markov process [GS01] is a stochastic process with the property that the future is independent of the past, conditioning only on its present value. Since there is no general acceptance on the terminologies, in this thesis, a Markov chain is a Markov process having discrete state space according to [Pan09].

DEFINITION 2.1 (***Discrete–Time Markov Chain***) A Discrete–Time Markov Chain (DTMC) is the process $\{X_n\}_{n \in T}$ satisfying the Markov property $\mathbb{P}(X_n = x_n \mid X_0 = x_0, X_1 = x_1, \ldots, X_{n-1} = x_{n-1}) = \mathbb{P}(X_n = x_n \mid X_{n-1} = x_{n-1})$ for all $n \geq 1$.

A DTMC is called *homogeneous* if it does not depend on time, i.e. $\mathbb{P}(X_n = j \mid X_{n-1} = i) = \mathbb{P}(X_1 = j \mid X_0 = i)$ for all n, i, j. This thesis only considers homogeneous DTMC.

To characterise a DTMC, we require an *initial probability vector* $\boldsymbol{\pi} = \{\pi_i\}$ and a *transition matrix* $\mathbf{P} = \{\mathrm{P}_{i,j}\}$, such that π_i is the probability that the system starts at state i, and $\mathrm{P}_{i,j} = \mathbb{P}(X_n = j \mid X_{n-1} = i)$ is the probability of a transition from state i to j. A valid $\boldsymbol{\pi}$ requires $\forall i : \pi_i \geq 0$ and $\sum_i \pi_i = 1$. A valid $\mathbf{P}$ requires $\forall i, j : \mathrm{P}_{i,j} \geq 0$ and $\forall i : \sum_j \mathrm{P}_{i,j} = 1$. The *transient probabilities* of a DTMC shall be calculated following the Chapman-Kolmogorov equations. Let $\boldsymbol{\pi}(t)$ be the transient probability distribution at time instant t such that $\boldsymbol{\pi}(0) = \boldsymbol{\pi}$, we have $\boldsymbol{\pi}(t) = \boldsymbol{\pi}\mathbf{P}^t$. The *limiting probability distribution* $\boldsymbol{\pi}(\infty)$ of a DTMC is to consider $\boldsymbol{\pi}(t)$ for $t \to \infty$, which characterises the long term behaviour. A DTMC is called *irreducible* iff it is possible to reach all states with nonzero probabilities from any state. For irreducible DTMCs, the limiting probabilities are independent of the initial probabilities, which is also called *steady-state or equilibrium probabilities*, such that $\boldsymbol{\pi}(\infty) = \boldsymbol{\pi}(\infty)\mathbf{P}$ and $\boldsymbol{\pi}(\infty)\mathbf{e} = 1$ hold.

DTMCs have been well-studied for decades, we refer the other definitions and properties to [KT75, GS01].

2.1.2 Discrete Phase-type Distributions

Let $\{X_n\}_{n \in \mathbb{N}_0}$ denote a DTMC with transient states $1, \ldots, p$ and one absorbing state $p + 1$. A Discrete PHase–type (DPH) distribution [Neu75] is the distribution of the time until absorption in the DTMC. DPH distributions are defined

by considering a probability transition matrix $\mathbf{P}$ of the form

$$\mathbf{P} = \begin{bmatrix} \mathbf{T} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{2.1}$$

where $\mathbf{T}$ is a sub-stochastic matrix between the transient states such that $\mathbf{I} - \mathbf{T}$ is non-singular, and $\mathbf{t} = \mathbf{e} - \mathbf{Te}$ is the vector of probabilities of transitions to the absorbing state. Let $(\boldsymbol{\pi}, \pi_{p+1})$ be the initial vector such that for $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_p)$ $\forall \pi_i \in \boldsymbol{\pi} : \pi_i \geq 0$ and $\boldsymbol{\pi}\mathbf{e} + \pi_{p+1} = 1$. Unless otherwise specified, we consider $\pi_{p+1} = 0$ to shorten expressions, but the extension to the case where $\pi_{p+1} > 0$ is straightforward.

DEFINITION 2.2 (*Discrete Phase–type Distribution*) We say that $W = \inf\{n \geq 1 | X_n = p + 1\}$ has a Discrete PHase–type (DPH) distribution with representation $(\boldsymbol{\pi}, \mathbf{T})$ and write $W \sim \mathrm{DPH}(\boldsymbol{\pi}, \mathbf{T})$ (cf. Formula (2.1)).

Let $\mathbb{P}$ denote probability and $\mathbb{E}$ denote expectation, the probability mass (or density) function f, the cumulative distribution function F, and the probability generating function G are given by

$$
\begin{aligned}
f(w) &= \mathbb{P}(W = w) = \boldsymbol{\pi}\mathbf{T}^{w-1}\mathbf{t}, \text{ for } w \geq 1, \\
F(w) &= \mathbb{P}(W \leq w) = 1 - \boldsymbol{\pi}\mathbf{T}^{w}\mathbf{e}, \\
G(z) &= \mathbb{E}(z^{W}) = z\boldsymbol{\pi}(\mathbf{I} - z\mathbf{T})^{-1}\mathbf{t}, \text{ for } |z| \leq 1.
\end{aligned} \tag{2.2}
$$

The probability generating function can be written in the rational form

$$G(z) = \frac{U(z)}{V(z)} = \frac{u_p z^p + u_{p-1} z^{p-1} + \cdots + u_1 z}{v_p z^p + v_{p-1} z^{p-1} + \cdots + v_1 z + 1} \tag{2.3}$$

for some constants $u_i, v_i \in \Re$, having the constraint that $\frac{\sum_{i=1}^{p} u_i}{1 + \sum_{i=1}^{p} v_i} = 1$ as we have $\pi_{p+1} = 0$ [Neu81].

Taking the i-th derivative of $G(z)$ and evaluating at $z = 1$ we get the i-th factorial moment of W, which is given by

$$\mathbb{E}(W(W-1)\ldots(W-(i-1))) = i!\boldsymbol{\pi}(\mathbf{I} - \mathbf{T})^{-i}\mathbf{T}^{i-1}\mathbf{e}, \ i \geq 1. \tag{2.4}$$

The *order of a DPH representation* $(\boldsymbol{\pi}, \mathbf{T})$ is given by the dimension of $\mathbf{T}$. Unfortunately, the DPH representation of a given distribution is, in general, neither unique [O'C89] nor necessarily minimal [Van90, HZ07]. Thus, we define the *order of a DPH distribution* to be the minimal order of all possible DPH representations.

2.1.3 Discrete–Time Markovian Arrival Processes

Markovian arrival processes (MAPs) [Neu79, Luc91, Asm00, Blo93, AN95] have many explicit analytic properties and are well-suited for numerical investigations, e.g. as models for successive interarrival times of phase–type.

DEFINITION 2.3 (*Discrete–Time Markovian Arrival Process*) Let $\mathbf{D}$ be an transition matrix of a DTMC of dimension p, and let $\mathbf{D}_0$ and $\mathbf{D}_1$ be two substochastic matrices whose sum is $\mathbf{D}$ such that the matrix $\mathbf{I} - \mathbf{D}_0$ is non-singular. The point process defined by an initial probability vector $\boldsymbol{\pi}$ and matrices $\{\mathbf{D}_0, \mathbf{D}_1\}$ is called a discrete–time Markovian Arrival Process (MAP) if the initial distribution and the transition probability matrix of the underlying Markov chain are given by $\boldsymbol{\pi}$ and $\mathbf{D} = \mathbf{D}_0 + \mathbf{D}_1$ respectively. The matrix $\mathbf{D}_0$ gives the probabilities of state changes without arrivals, and the matrix $\mathbf{D}_1$ gives the probabilities of state changes with arrivals. We say that a discrete–time MAP has a representation $(\boldsymbol{\pi}, \mathbf{D}_0, \mathbf{D}_1)$ of order p.

An *embedded Markov chain* describes the sequence of states with arrivals in a discrete–time MAP. The transition probability matrix of the embedded Markov chain is $(\mathbf{I} - \mathbf{D}_0)^{-1}\mathbf{D}_1$. Let N_t be the accumulated number of arrivals at time epochs $1, 2, \ldots, t$, X_t be the state of the Markov chain at time t, and $\mathrm{P}(n,t)_{i,j} = \mathbb{P}(N_t = n, X_t = j | N_0 = 0, X_0 = i)$ be the (i,j)-th entry of a matrix $\mathbf{P}(n,t)$. The matrices $\mathbf{P}(n,t)$ satisfy the forward Chapman-Kolmogorov difference equations

$$
\begin{aligned}
\mathbf{P}(0,0) &= \mathbf{I}, \\
\mathbf{P}(0,t+1) &= \mathbf{P}(0,t)\mathbf{D}_0, \\
\mathbf{P}(n,t+1) &= \mathbf{P}(n,t)\mathbf{D}_0 + \mathbf{P}(n-1,t)\mathbf{D}_1, \ n \geq 1.
\end{aligned}
$$

PROPOSITION 2.4 *The generating function of N_t in a discrete–time Markovian arrival process is given by*

$$
\mathbb{E}(z^{N_t}) = \boldsymbol{\pi} \sum_{n=0}^{\infty} z^n \mathbf{P}(n,t)\mathbf{e} = \boldsymbol{\pi}(\mathbf{D}_0 + z\mathbf{D}_1)^t \mathbf{e}, \ |z| \leq 1.
$$

PROOF We show that $\sum_{n=0}^{\infty} z^n \mathbf{P}(n,t)$ equals $(\mathbf{D}_0 + z\mathbf{D}_1)^t$ by induction.

For base case, let $t = 0$, we have

$$
\sum_{n=0}^{\infty} z^n \mathbf{P}(n,0) = z^0 \mathbf{P}(0,0) + \sum_{n=1}^{\infty} z^n \mathbf{P}(n,0) = \mathbf{I}.
$$

When $t = 1$, we have

$$\sum_{n=0}^{\infty} z^n \mathbf{P}(n,1) = z^0 \mathbf{P}(0,1) + z^1 \mathbf{P}(1,1) + \sum_{n=2}^{\infty} z^n \mathbf{P}(n,1) = \mathbf{D}_0 + z\mathbf{D}_1.$$

For inductive step, let $s \geq 2$ and let $t = s + 1$, we have

$$
\begin{aligned}
\sum_{n=0}^{\infty} z^n \mathbf{P}(n, s+1) &= \sum_{n=0}^{\infty} z^n \left(\mathbf{P}(n,s)\mathbf{D}_0 + \mathbf{P}(n-1,s)\mathbf{D}_1 \right) \\
&= \sum_{n=0}^{\infty} z^n \mathbf{P}(n,s)\mathbf{D}_0 + \sum_{n=1}^{\infty} z^{n-1}\mathbf{P}(n-1,s)z\mathbf{D}_1 \\
&= \sum_{n=0}^{\infty} z^n \mathbf{P}(n,s)\mathbf{D}_0 + \sum_{n=0}^{\infty} z^n \mathbf{P}(n,s)z\mathbf{D}_1 \\
&= \left(\sum_{n=0}^{\infty} z^n \mathbf{P}(n,s) \right) (\mathbf{D}_0 + z\mathbf{D}_1).
\end{aligned}
$$

That produces

$$\sum_{n=0}^{\infty} z^n \mathbf{P}(n, s+1) = (\mathbf{D}_0 + z\mathbf{D}_1)^{s+1}.$$

$\square$

The joint transform $f(s,z)$ of N_t in the $[0,t]$ interval starting with an arrival right before 0 is

$$f(s,z) = \sum_{t=0}^{\infty} s^t \mathbb{E}(z^{N_t}) = \sum_{t=0}^{\infty} s^t \boldsymbol{\pi}(\mathbf{D}_0 + z\mathbf{D}_1)^t \mathbf{e} = \boldsymbol{\alpha} \left(\mathbf{I} - s(\mathbf{D}_0 + z\mathbf{D}_1) \right)^{-1} \mathbf{e}.$$

For a MAP $(\boldsymbol{\pi}, \mathbf{D}_0, \mathbf{D}_1)$, the probability mass function and the i-th factorial moment of the first arrival time W [Neu81] are

$$f(w) = \boldsymbol{\pi} \mathbf{D}_0^{w-1} \mathbf{D}_1 \mathbf{e}, \quad \text{for } w \geq 1,$$
$$\mathbb{E}(W(W-1)\ldots(W-(i-1))) = i! \boldsymbol{\pi}(\mathbf{I} - \mathbf{D}_0)^{-i}\mathbf{D}_0^{i-1}(\mathbf{I} - \mathbf{D}_0)^{-1}\mathbf{D}_1\mathbf{e}. \quad (2.5)$$

Note that $(\mathbf{I} - \mathbf{D}_0)^{-1}\mathbf{D}_1\mathbf{e} = \mathbf{e}$, thus the factorial moments of interarrival times of a discrete–time MAP is equivalent to the factorial moments of a DPH (cf. Formula (2.4)). Notice that by interpreting an arrival as a sort of absorption, the matrix $\mathbf{D}_0$ of a MAP is indeed the sub-stochastic matrix $\mathbf{T}$ of a DPH.

For $m \in \mathbb{N}_0$, the joint density function and the joint generating function of the first successive $m + 1$ inter-arrival times $W_0, W_1, \ldots, W_m$ are

$$f(w_0, w_1, \ldots, w_m) = \boldsymbol{\pi} \left(\prod_{i=0}^{m} \mathbf{D}_0^{w_i-1} \mathbf{D}_1 \right) \mathbf{e},$$

$$G(z_0, z_1, \ldots, z_m) = \boldsymbol{\pi} \left(\prod_{i=0}^{m} z_i (\mathbf{I} - z_i \mathbf{D}_0)^{-1} \mathbf{D}_1 \right) \mathbf{e}.$$

Let $t_0, t_1, \ldots, t_m$ be nonnegative integers, the joint factorial moments of the first $m + 1$ arrival times $W_0, W_1, \ldots, W_m$ are

$$\mathbb{E} \left(\prod_{i=0}^{m} W_i (W_i - 1) \ldots (W_i - (t_i - 1)) \right) = \boldsymbol{\pi} \left(\prod_{i=0}^{m} t_i! (\mathbf{I} - \mathbf{D}_0)^{-(t_i+1)} \mathbf{D}_0^{t_i-1} \mathbf{D}_1 \right) \mathbf{e}.$$
$$(2.6)$$

2.1.4   Discrete–Time Marked Markovian Arrival Processes

A discrete–time Markovian arrival process with marked transitions, named a Marked Markovian Arrival Process (MMAP), is a point process generated by marked transitions of a finite state DTMC with initial probability vector $\boldsymbol{\pi}$ and probability transition matrix $\mathbf{D}$.

Suppose that at time t, the chain is in some state i, i.e. $X_t = i$. At the next time instant, a transition occurs with or without an arrival. With probability $(D_0)_{i,j}$, there is a transition from state i to state j without an arrival. Assume we have K event types and let $h_1, \ldots, h_K$ be non-negative integers, then without loss of generality an arrival marked by $\boldsymbol{h} = (h_1, \ldots, h_K)$ counts h_k items of type k for $k = 1, \ldots, K$. Henceforth $(D_{\boldsymbol{h}})_{i,j}$ is the probability of a transition from state i to state j with the arrival marked by $\boldsymbol{h}$. Let $\mathcal{C}$ be a countable set of K-tuples of non-negative integers excluding $\boldsymbol{0}$, then we have for all i that $\sum_{j=1}^{p} \left((D_0)_{i,j} + \sum_{\boldsymbol{h} \in \mathcal{C}} (D_{\boldsymbol{h}})_{i,j} \right) = 1$.

Similarly to the continuous–time marked Markovian arrival process defined in [HN98], we have the following:

DEFINITION 2.5 (*Discrete–Time Marked Markovian Arrival Process*)
A discrete–time Marked Markovian Arrival Process (MMAP) is a point process defined by a vector $\boldsymbol{\pi}$ and matrices $\{\mathbf{D}_{\boldsymbol{h}}\}_{\boldsymbol{h} \in \mathcal{C} \cup \{0\}}$. The initial distribution and the transition probability matrix of the underlying Markov chain are given by $\boldsymbol{\pi}$ and $\mathbf{D} = \mathbf{D}_0 + \sum_{\boldsymbol{h} \in \mathcal{C}} \mathbf{D}_{\boldsymbol{h}}$. The matrix $\mathbf{D}_0$ gives the probabilities of state changes

without arrivals. For each vector of marks $\boldsymbol{h}$, the matrix $\mathbf{D_h}$ gives the probabilities of state changes with the arrival marked by $\boldsymbol{h}$. We say that an MMAP has a representation $(\boldsymbol{\pi}, \{\mathbf{D_h}\}_{\boldsymbol{h} \in \mathcal{C} \cup \{0\}})$ of order p, where p is the dimension of the matrix $\mathbf{D}_0$.

The transition probability matrix of the embedded Markov chain of discrete–time MMAP is $(\mathbf{I} - \mathbf{D}_0)^{-1} \sum_{\boldsymbol{h} \in \mathcal{C}} \mathbf{D_h}$. The counting process of the MMAP, $\boldsymbol{N}(t) = (N_1(t), \ldots, N_K(t))$, is a tuple of random variables such that $N_k(t)$ is the accumulated number of items of type k arriving up to and including time epoch t. Let $n_1, \ldots, n_K$ be nonnegative integers and $\boldsymbol{n} = (n_1, \ldots, n_K)$, we then define the conditional probability $P(\boldsymbol{n}, t)_{i,j}$ for the underlying Markov chain being in the state j at time instant t with $\boldsymbol{n}$ accumulated arrivals, given that the process started in state i at time 0, i.e. $P(\boldsymbol{n}, t)_{i,j} = \mathbb{P}(\boldsymbol{N}(t) = \boldsymbol{n}, X_t = j | \boldsymbol{N}(0) = \mathbf{0}, X_0 = i)$ for $1 \le i, j \le p$.

Let $\mathbf{P}(\boldsymbol{n}, t)$ be the matrix with the conditional probabilities $P(\boldsymbol{n}, t)_{i,j}$, which satisfies the discrete Chapman-Kolmogorov difference equations

$$
\begin{aligned}
\mathbf{P}(\mathbf{0}, 0) &= \mathbf{I}, \\
\mathbf{P}(\mathbf{0}, t+1) &= \mathbf{P}(\mathbf{0}, t)\mathbf{D}_0, \\
\mathbf{P}(\boldsymbol{n}, t+1) &= \mathbf{P}(\boldsymbol{n}, t)\mathbf{D}_0 + \sum_{\boldsymbol{h} \le \boldsymbol{n}, \boldsymbol{h} \in \mathcal{C}} \mathbf{P}(\boldsymbol{n} - \boldsymbol{h}, t)\mathbf{D_h}.
\end{aligned}
$$

PROPOSITION 2.6 *The generating function of $\boldsymbol{N}(t)$ in a discrete–time marked Markovian arrival process is given by*

$$
\mathbb{E}\left(\prod_{i=1}^{K} z_i^{N_i(t)}\right) = \boldsymbol{\pi} \sum_{\boldsymbol{n} \ge \mathbf{0}} \boldsymbol{z^n} \mathbf{P}(\boldsymbol{n}, t)\mathbf{e} = \boldsymbol{\pi} \left(\mathbf{D}_0 + \sum_{\boldsymbol{h} \in \mathcal{C}} \boldsymbol{z^h} \mathbf{D_h}\right)^t \mathbf{e}, \quad t \ge 0,
$$

where $\boldsymbol{z^h} = \prod_{i=1}^{K} z_i^{h_i}$.

PROOF We show that $\sum_{\boldsymbol{n} \ge \mathbf{0}} \boldsymbol{z^n} \mathbf{P}(\boldsymbol{n}, t)$ equals $\left(\mathbf{D}_0 + \sum_{\boldsymbol{h} \in \mathcal{C}} \boldsymbol{z^h} \mathbf{D_h}\right)^t$ by induction.

For base case, let $t = 0$, we have

$$
\sum_{\boldsymbol{n} \ge \mathbf{0}} \boldsymbol{z^n} \mathbf{P}(\boldsymbol{n}, 0) = \boldsymbol{z^0} \mathbf{P}(\mathbf{0}, 0) + \sum_{\boldsymbol{n} > \mathbf{0}} \boldsymbol{z^n} \mathbf{P}(\boldsymbol{n}, 0) = \mathbf{I}.
$$

When $t = 1$, we have

$$\sum_{n \geq 0} z^n \mathbf{P}(n, 1) = z^0 \mathbf{P}(0, 1) + \sum_{n > 0} z^n \mathbf{P}(n, 1)$$

$$= \mathbf{D}_0 + \sum_{n > 0} z^n \left(\mathbf{P}(n, 0)\mathbf{D}_0 + \mathbf{P}(0, 0)\mathbf{D}_n + \sum_{h < n, h \in \mathcal{C}} \mathbf{P}(n - h, 1)\mathbf{D}_h \right)$$

$$= \mathbf{D}_0 + \sum_{h \in \mathcal{C}} z^h \mathbf{D}_h,$$

such that $\sum_{h < n, h \in \mathcal{C}} \mathbf{P}(n - h, 1)\mathbf{D}_h = 0$, because strictly $n - h > 0$ then

$$\mathbf{P}(n - h, 1)\mathbf{D}_h = \mathbf{P}(n - h, 0)\mathbf{D}_0 + \sum_{h' \leq n - h, h' \in \mathcal{C}} \mathbf{P}(n - h - h', 0)\mathbf{D}_{h'} = 0.$$

For the inductive step, let $s \geq 2$ and let $t = s + 1$, we have

$$\sum_{n \geq 0} z^n \mathbf{P}(n, s + 1) = \sum_{n \geq 0} z^n \left(\mathbf{P}(n, s)\mathbf{D}_0 + \sum_{h \leq n, h \in \mathcal{C}} \mathbf{P}(n - h, s)\mathbf{D}_h \right)$$

$$= \sum_{n \geq 0} z^n \mathbf{P}(n, s)\mathbf{D}_0 + \sum_{n \geq 0} z^n \sum_{h \leq n, h \in \mathcal{C}} \mathbf{P}(n - h, s)\mathbf{D}_h,$$

such that the term $\sum_{n \geq 0} z^n \sum_{h \leq n, h \in \mathcal{C}} \mathbf{P}(n - h, s)\mathbf{D}_h$ includes finitely many terms. Without loss of generality, assume that the vector of marks of the MMAP are h_1, h_2, h_3 .... To expand the term we have the following:

$$\sum_{n \geq 0} z^n \sum_{h \leq n, h \in \mathcal{C}} \mathbf{P}(n - h, s)\mathbf{D}_h$$

$$= \sum_{n \geq 0} z^n \left(\mathbf{P}(n - h_1, s)\mathbf{D}_{h_1} + \mathbf{P}(n - h_2, s)\mathbf{D}_{h_2} + \dots \right)$$

$$= \sum_{n \geq h_1 \geq 0} z^{n - h_1} \mathbf{P}(n - h_1, s) z^{h_1} \mathbf{D}_{h_1} + \sum_{n \geq h_2 \geq 0} z^{n - h_2} \mathbf{P}(n - h_2, s) z^{h_2} \mathbf{D}_{h_2} + \dots$$

$$= \sum_{n \geq h_1 \geq 0} z^n \mathbf{P}(n, s) z^{h_1} \mathbf{D}_{h_1} + \sum_{n \geq h_2 \geq 0} z^n \mathbf{P}(n, s) z^{h_2} \mathbf{D}_{h_2} + \dots$$

$$= \left(\sum_{n \geq 0} z^n \mathbf{P}(n, s) \right) \left(\sum_{h \leq n, h \in \mathcal{C}} z^h \mathbf{D}_h \right).$$

Then, we have

$$\sum_{n \geq 0} z^n \mathbf{P}(n, s+1) = \sum_{n \geq 0} z^n \mathbf{P}(n, s) \mathbf{D}_0 + \left(\sum_{n \geq 0} z^n \mathbf{P}(n, s) \right) \left(\sum_{h \leq n, h \in \mathcal{C}} z^h \mathbf{D}_h \right)$$

$$= \left(\sum_{n \geq 0} z^n \mathbf{P}(n, s) \right) \left(\mathbf{D}_0 + \sum_{h \leq n, h \in \mathcal{C}} z^h \mathbf{D}_h \right).$$

That produces

$$\sum_{n \geq 0} z^n \mathbf{P}(n, s+1) = \left(\mathbf{D}_0 + \sum_{h \leq n, h \in \mathcal{C}} z^h \mathbf{D}_h \right)^{s+1}.$$

$\square$

The joint transform $f(s, \mathbf{z})$ of $\mathbf{N}(t)$ in the $[0, t]$ interval starting with an arrival right before 0 is

$$f(s, \mathbf{z}) = \sum_{t=0}^{\infty} s^t \mathbb{E} \left(\prod_{i=1}^{K} z_i^{N_i(t)} \right) = \boldsymbol{\pi} \left(\mathbf{I} - s \left(\mathbf{D}_0 + \sum_{h \in \mathcal{C}} z^h \mathbf{D}_h \right) \right)^{-1} \mathbf{e}, \quad (2.7)$$

where $z^h = \prod_{i=1}^{K} z_i^{h_i}$.

For an MMAP with representation $(\boldsymbol{\pi}, \{\mathbf{D}_h\}_{h \in \mathcal{C} \cup \{0\}})$, the joint density of the first successive $m + 1$ interarrival times $\Theta_0, \ldots, \Theta_m$ with arrivals marked by $\boldsymbol{H}_0, \ldots, \boldsymbol{H}_m$ is

$$f(\theta_0, \boldsymbol{h}_0, \ldots, \theta_m, \boldsymbol{h}_m) = \mathbb{P}(\Theta_0 = \theta_0, \boldsymbol{H}_0 = \boldsymbol{h}_0, \ldots, \Theta_m = \theta_m, \boldsymbol{H}_m = \boldsymbol{h}_m)$$

$$= \boldsymbol{\pi} \left(\prod_{i=0}^{m} \mathbf{D}_0^{\theta_i - 1} \mathbf{D}_{\boldsymbol{h}_i} \right) \mathbf{e}. \quad (2.8)$$

Let $t_0, \ldots, t_m$ be nonnegative integers, the joint factorial moments of the first successive $m + 1$ interarrival times $\Theta_0, \ldots, \Theta_m$ with the arrivals marked by $\boldsymbol{H}_0, \ldots, \boldsymbol{H}_m$ is

$$\mathbb{E} \left(\prod_{i=0}^{m} \Theta_i (\Theta_i - 1) \ldots (\Theta_i - (t_i - 1)) \mathbf{1}(\boldsymbol{H}_i = \boldsymbol{h}_i) \right)$$

$$= \boldsymbol{\pi} \left(\prod_{i=0}^{m} t_i! (\boldsymbol{I} - \mathbf{D}_0)^{-(t_i+1)} \mathbf{D}_0^{t_i - 1} \mathbf{D}_{\boldsymbol{h}_i} \right) \mathbf{e}, \quad (2.9)$$

such that the $\mathbf{1}(\boldsymbol{H}_i = \boldsymbol{h}_i)$ is an indicator function, which equals 1 when $\boldsymbol{H}_i = \boldsymbol{h}_i$.

2.2 Continuous–time Processes

2.2.1 Continuous–Time Markov Chains

DEFINITION 2.7 (*Continuous–Time Markov Chain*) A Continuous–Time Markov Chain (CTMC) is a process $\{X_t\}_{t \geq 0}$ that satisfies the Markov property $\mathbb{P}(X_{t_n} = j \mid X_{t_0} = i_0, \ldots, X_{t_{n-1}} = i_{n-1}) = \mathbb{P}(X_{t_n} = j \mid X_{t_{n-1}} = i_{n-1})$ for any sequence $t_0 < t_1 < \cdots < t_n$ of times.

A CTMC is called *homogeneous* if it does not depend on time, i.e. $\mathbb{P}(X_{t_n} = j \mid X_{t_{n-1}} = i) = \mathbb{P}(X_{t_1} = j \mid X_{t_0} = i)$ for all t_n, t_{n-1}, i, j. This thesis only considers homogeneous CTMCs.

To characterise a CTMC, we require an *initial probability vector* $\boldsymbol{\pi} = \{\pi_i\}$ and a *generator matrix* $\mathbf{Q} = \{Q_{i,j}\}$, such that the π_i is the probability that system starts at state i, and the $Q_{i,j}$ are parameters of (negative) Exponential (for short Exp.) distributions. A valid $\mathbf{Q}$ requires that $\forall i \neq j : Q_{i,j} \geq 0$, $\forall i : Q_{i,i} \leq 0$, $\sum_j Q_{i,j} = 0$. The *transient probability vector* of a CTMC shall be calculated as $\boldsymbol{\pi}(t) = \boldsymbol{\pi} e^{\mathbf{Q}t}$. Like DTMCs, a CTMC is called *irreducible* iff it is possible to reach all states with nonzero probabilities from any state in its embedded Markov chain [GS01]. For an irreducible CTMCs, the *steady-state or equilibrium probability vector* $\boldsymbol{\pi}(\infty)$ are given as $\mathbf{0} = \boldsymbol{\pi}(\infty)\mathbf{Q}$ and $\boldsymbol{\pi}(\infty)\mathbf{e} = 1$.

Like DTMCs, CTMCs have been well-studied for decades, we refer the other definitions and properties to [KT75, GS01].

2.2.2 Continuous Phase–type Distributions

Many definitions and results regarding DPH distributions (cf. Section 2.1.2) carry over verbatim to the continuous case, other need minor modifications.

DEFINITION 2.8 (*Continuous Phase–type Distribution*) A Continuous PHase–type (CPH) distribution is the distribution of the time until absorption in a finite continuous–time Markov chain with generator matrix $\mathbf{Q}$ of dimension $p + 1$ given as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{T} & \mathbf{t} \\ \mathbf{0} & 0 \end{bmatrix}, \tag{2.10}$$

such that the square matrix $\mathbf{T}$ is invertible, has nonpositive row sums and nonnegative off-diagonal entries. Let $(\boldsymbol{\pi}, \pi_{p+1})$ be the initial probability vector of the CTMC, the pair $(\boldsymbol{\pi}, \mathbf{T})$ is called a representation of the CPH distribution.

The probability density function f, the cumulative distribution function F, and the Laplace-Stieltjes Transform (LST) H of a CPH distributed random variable W are given by

$$
\begin{aligned}
f(w) &= \boldsymbol{\pi} e^{\mathbf{T}w} \mathbf{t}, \text{ for } w > 0, & (2.11) \\
F(w) &= 1 - \boldsymbol{\pi} e^{\mathbf{T}w} \mathbf{e}, & \\
H(s) &= \mathbb{E}(e^{-sW}) = \pi_{p+1} + \boldsymbol{\pi}(s\mathbf{I} - \mathbf{T})^{-1}\mathbf{t}. & (2.12)
\end{aligned}
$$

Let W be a CPH distributed random variable, the n-th moment $\mathbb{E}(W^n)$ is given by

$$
\mathbb{E}(W^n) = n! \boldsymbol{\pi}(-\mathbf{T})^{-n} \mathbf{e}. \tag{2.13}
$$

The *order of a CPH representation* $(\boldsymbol{\pi}, \mathbf{T})$ is given by the dimensions of $\mathbf{T}$. In analogy with DPH distributions, the CPH representation of a given distribution is, in general, neither unique nor necessarily minimal. Thus, we define the *order of a CPH distribution* to be the minimal order of all possible CPH representations.

Provided the LSTs of two independent continuous random variables exist, the distributions are equal if and only if their LSTs are equal (see the continuity theorem, page 429 in [Fel68]). By Formula (2.12), we observe that CPH distributions have a rational function. If the rational function has no common factor, the *degree of the phase–type distribution* is defined to be the degree of denominator. It has been proved in [Van90] that two CPH representations of order n and m respectively, describe the same CPH distribution if and only if their first $2 \cdot \max\{n, m\}$ moments agree.

2.2.3 Continuous–time Markovian Arrival Process

Continuous–time Markov arrival processes [Neu79, Asm00, Nie98] have many analogous results to the discrete–time case mentioned in Section 2.1.3. This thesis focuses on discrete–time MAPs, so we omit the detailed definitions and properties for continuous–time MAPs.

2.2.4 Continuous–time Marked Markovian Arrival Process

Like continuous–time MAPs, the original definitions of marked Markovian arrival processes and other results are in continuous–time [HN98, BKK10]. This

thesis focuses on discrete–time MMAPs to supplement the literatures, thus we omit the detailed definitions and properties for continuous–time MMAPs.

2.2.5   Markov Renewal Processes

A *Markov Renewal Process* (MRP) [Ros96] is a continuous–time stochastic process such that the exponentially distributed sojourn times in a CTMC are now generalised to follow arbitrary probability distributions. The probability distributions of the holding time of an MRP is referred as *the kernel* of an MRP.

DEFINITION 2.9 (***Markov Renewal Process***) Let $m \in \mathbb{N}$, $E = \{1, \dots, m\}$ be a discrete state space, and $\{H_n\}_{n=0,1,2,\dots}$ be a stochastic process assuming values in E. Let W_0, W_1, W_2, ... be the continuous–time transition epochs on $[0, \infty)$, such that $0 = W_0 \leq W_1 \leq W_2 \leq \cdots$. A two-dimensional process $(H, W) = \{(H_n, W_n)\}_{n=0,1,2,\dots}$ is a Markov Renewal Process (MRP) if it has the property

$$\mathbb{P}(H_{n+1} = j, W_{n+1} - W_n \leq t \mid H_0, H_1, \dots, H_n = i, W_0, W_1, \dots, W_n) =$$
$$\mathbb{P}(H_{n+1} = j, W_{n+1} - W_n \leq t \mid H_n = i), \quad i, j \in E \text{ and } t \geq 0.$$

That is the density of jumping to the future state j at time t depends only on the current state i.

An MRP represents both the transition epoch and the state of the process at that epoch. The stochastic process representing the states of an MRP at an arbitrary time is called a *Semi-Markov Process*.

Phase–type Distributions in Stochastic Analysis

Phase–type (PH) distributions were considered first in [Neu75, Neu81], and are defined as distributions of absorption times in a Markov chain with finite transient states and one absorbing state. There are several motivations for using PH distributions in stochastic modelling, such as well–established analytical results (cf. Section 2.1.2 and Section 2.2.2), compositionality [Neu75], to approximate any kind of probability distributions on $[0, \infty)$ [JT88], and so on. The applications of PHs can be found in various areas in stochastic modelling such as reliability analysis [Neu81, CP09], queueing theory [Asm92, Neu89], and networks [CNI04, EZN11].

This chapter presents our recent developments for both discrete and continuous PH distributions, which consists of two separate parts. In the first part, we report our recent studies on applying discrete PH distributions for evaluating performance of probabilistic concurrent wireless sensor network protocols. This work proposes a highly abstracted probabilistic model using combinatorial reasoning, such that discrete PH distributions are able to accelerate the performance analysis. This result has been published and presented in [EZN11]. The second part is about exploring compositionality results for continuous PH distributions by introducing a stochastic process algebra, named phase–type process algebra. Phase–type process algebra provides a way of modelling large-scale and

complicated systems using continuous PH distributions as building blocks.

3.1 A Probabilistic Model of the LMAC Protocol

In this section, we present a probabilistic model for the network setup phase
of the Lightweight Medium Access Protocol (LMAC) for concurrent wireless
sensor networks. In the network setup phase, time slots are allocated to the
individual sensors through resolution of successive collisions. The setup phase
involving collisions should preferably be as short as possible for efficiency and
energy consumption reasons. This concurrent stochastic process has inherent
internal nondeterminism, and we model it using combinatorics. The setup phase
is modelled by a discrete–time Markov chain such that we can apply results
from the theory of phase–type distributions. Having obtained our model we are
able to find optimal protocol parameters. We have simultaneously developed a
simulation model, partly to verify our analytical derivations and partly to be
able to deal with systems of excessively high order or stiff systems that might
cause numerical challenges. Our abstracted model has a state space of limited
size where the number of states are of the order $\binom{n+r+1}{n}$, where n is number of
sensors, and r is the maximum back-off time. We have developed a tool, named
LMAC analyser, on the MATLAB platform to assist automatic generation and
analysis of the model.

3.1.1 Introduction

Wireless Sensor Networks (WSN) consist of widely distributed sensors that co-
operatively monitor physical or environmental conditions, and have been used
in widespread applications. WSN are one of the prime examples of net-worked
embedded systems, where many modern computer science challenges exist, such
as the challenges in distributed computing, wireless communication, and system
integration. One major consideration in WSN is how to prolong the network
lifetime. The Lightweight Medium Access Protocol (LMAC) was introduced
in [vHH04], designed as a multi-hop and energy-efficient protocol for WSN at
the Medium Access Control (MAC) layer. In the LMAC protocol, the network
is self-organising in terms of time slot assignment and synchronisation. The
protocol uses Time Division Multiple Access (TDMA), where each time slot is
assigned to a sensor. In this manner, the nodes can communicate collision- free
after the network has stabilised. In this way, the protocol provides energy effi-
ciency. The LMAC protocol gives a significant lifetime improvement compared
to prior protocols, such as EMACs and SMAC. Thus, analysing and reducing

collisions in order to optimise the network have been some of the remaining challenges. As a consequence, we will concentrate on the part of the protocol that is responsible for the distributed and localised strategy of assigning time slots to sensors.

In previous works [FvHM07, vH07, VR09], the concurrent behaviour of the LMAC nodes was modelled using parallel decomposition. The complexity of the models rose drastically with the number of network nodes, limiting the analysis to small-sized networks far from reality. In addition, the random nature of the slot selection process required further analysis and modelling. In our approach, we will use a direct mathematical method with immediate abstraction of network details that does not influence the time for the network to stabilise.

Related Work   Although LMAC came into being just a few years ago, there are already some interesting studies.

In [FvHM07, vH07], formal verification of the LMAC protocol is investigated in the timed automaton model checker UPPAAL [BDL04]. The LMAC network is modelled by parallel composition of single node behaviours. The properties for model check- ing primarily focus on the fundamental mechanisms. For example, checking whether collisions can be detected or a new choice of slots is initiated after collision. The UPPAAL model has been used to systematically investigate all topologies with 4 and 5 nodes. Based on this work, the LMAC protocol has been updated by patching discovered bugs, and problematic topologies with possible scenarios of unsolved collision have been identified. However, the UPPAAL model has encountered serious state space explosion, if the model contains more than 5 nodes. Moreover, probabilistic aspects of LMAC, e.g. optimal parameters, have been mentioned as important future work. This inspired our work and has now become one of our key contributions.

A study on probabilistic aspects of LMAC is given in [VR09] relying on timed automata in the probabilistic version of UPPAAL (UPPAAL PRO 0.2). The probabilistic choice has been made by pre-assigned weights to all possible transitions when nodes select back-off time after collision. By changing the weights, various probability distributions represent different back-off strategies. It shows that if the back-off time before starting to pick a new time slot increases, the number of collisions will decrease. In this work [VR09], slot selection is modelled in a deterministic way. Each individual sensor keeps trying from the first time slot and then the second, until it finds a free one. Thereby, later coming sensors unavoidably have a number of collisions before they settle down. This has negative influences on the overhead of the protocol, therefore we suggest a probabilistic solution. At last, the probabilistic UPPAAL model encounters

even severer state explosion problem than the non-probabilistic version, which is capable of modelling a maximum of 4 nodes under the fully connected topology.

In summary, previous works in the slot selection phase of the LMAC protocol rely on parallel composition approach. Modelling the non-deterministic nature and solving the state space dilemma are very interesting topics. In our work, we propose a mathematical approach using a direct abstraction technique to solve these concurrent stochastic problems.

3.1.2   The LMAC Protocol
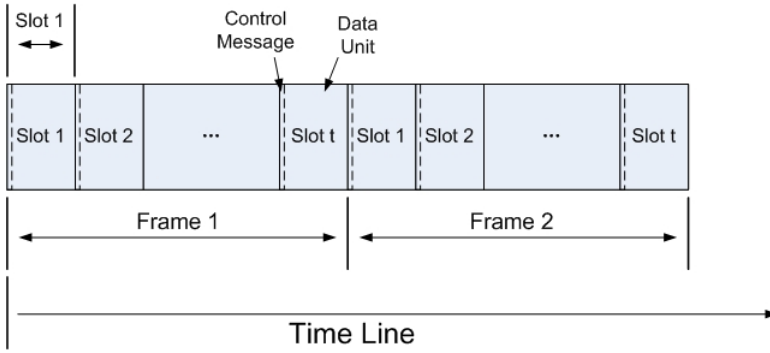
Figure 3.1: Time structure in the LMAC protocol.

As a schedule-based MAC protocol, the time in LMAC is organised in time slots, which are grouped into frames (see Fig. 3.1). For each time slot, the controlling node always transmits a fixed length (12 bytes, [vHH04]) control message in order to maintain synchronisation. The control message also carries a node ID of the time slot controller, the size of the data unit and the intended receiver. In particular, the control message is critical for broadcasting information regarding the occupied time slots. For this reason, late coming nodes can pick only free slots. The remaining part of a time slot is an optional data unit if there are any needs. The current maximum size of the data unit is 256 bytes, [vHH04]. During each frame, nodes can only transmit messages in their own time slot, for the rest of the time they can only receive messages. In this manner, energy consumption is minimised.

At the beginning of the network setup phase, all of the nodes are unsynchronised. In order to get synchronisation, one (or more) gateway node(s) will take initiative to start controlling the time slot(s), i.e. becoming the master node(s). Control messages from the gateway will be received by its one-hop neighbours.

Once these nodes get their time slots, they will start sending control messages to the other hops. The network will stabilise once all nodes get their reserved time slots. Thereafter, nodes can communicate with each other in a collision free manner.

Fig. 3.2 describes the behaviour in terms of the phases for an individual node in LMAC.
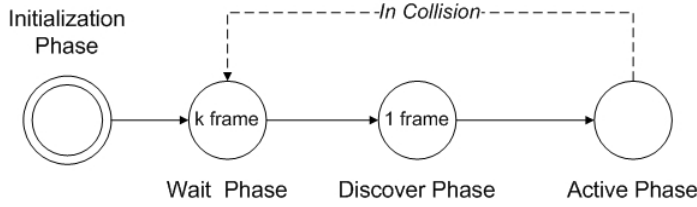
Figure 3.2: Phase diagram for an individual sensor.

Initialisation Phase: When a sensor node powers on, it is unsynchronised. In this phase, a node will try to detect its neighbouring nodes. As long as at least one neighbouring node is detected, the node will synchronise with it and go to the wait phase.

Wait Phase: The wait phase is designed with the purpose of reducing the number of nodes that pick slots at the same time, which helps to reduce the probability of collision. In this phase, a node waits for random k frames, where k is an integer number from the set $S = \{0, 1, \ldots, r\}$, where r is the maximum back-off time. After waiting k frames, the node will go to the discover phase.

Discover Phase: Before a node starts to pick a time slot, it registers all the currently available slots in order to pick only among those. This happens in the discover phase where nodes compute free slot information based on the control messages. Afterwards, it will randomly select one of the available slots and go to the active phase. A node stays one time frame in this phase.

Active Phase: After a node has picked a time slot in the discover phase, it will start to transmit a message and receive messages from neighbouring nodes. Here, if there are two or more nodes transmitting simultaneously, a collision occurs. Then neighbouring nodes will send control messages to ask them to give up their time slots and go to the wait phase.

In the LMAC specification, the back-off mechanism for collided nodes is under-specified. Thereby, it is possible for nodes to start back-off in either the current frame or the next. It depends on whether there are neighbouring nodes to register collisions for the discover identities at the remaining time of the frame.

This non-determinism can be interpreted as implementation freedom. Considering the worst case scenario, we resolve the non-determinism by assuming that back-off always starts in the successive frame.

Moreover, to limit the number of time slots necessary in the network, the LMAC protocol allows for time slots to be reused at non-interfering distances. In [FvHM07] is proved that it is safe to share the same time slot after at least three hops. The LMAC protocol uses a distributed algorithm described in [TSP⁺03, MW08] to manage the division of time slots.

3.1.3   A Probabilistic Model

The component-based approach to model the concurrency in the stabilisation process was applied in [FvHM07, VR09], where the behaviour of a single node is modelled as a basic component. Indeed, the system properties are represented in terms of parallel composition of individual nodes. The compositional way is inherently close to the protocol specification, which gives a detailed verification result. The state space explosion problem, however, restricts model checking experiments to 5 nodes. This is far from the WSN applications, where the number of sensors could be up to hundreds. Therefore, we will propose a lightweight model that will be valid for the verification task at hand.

In [FvHM07], the case with 5 nodes considering all the 61 topologies has been investigated. These different topologies can lead to dramatically varied verification results. Generally, it is hard to identify a representative one. Hence, throughout the paper, we will assume only the fully connected topology, which has been proven as one of the successful topologies.

3.1.3.1   System Abstraction

Related to parallel composition, the abstraction methods described in [CGL94, CC77] produce abstracted models to reduce the state space problem of model checking. But, in order to do so, they have to start with a detailed model. As an alternative, our mathematical approach will attack the highly abstracted model directly, thereby obtaining a huge reduction of the state space. The system is abstracted by the statistical collection of system level information based on the given LMAC specification. The statistical collection is represented by a data vector which also represents the state of the system. In our implementation, we will propose an injective function, which maps a set of data vectors to a set of positive integers in order to construct a one dimensional discrete state space.

In Section 3.1.2, we introduced the LMAC protocol, which uses a distributed algorithm to divide time slots in order to reuse them in more than 2 hops. Thereby, it is sufficient to analyse the worst case of at most 2 hops distance to characterise the overall network. Thus, we will only model the behaviour for the worst case, defined by $\max\{\frac{n}{t} \mid \frac{n}{t} \leq 1; n, t \text{ within 2 hops}\}$, where n and t are the system variables defined below.

In the following we will define the system variables.

- n is the number of sensors.

- t is the number of time slots, where we assume $t \geq n$, since the number of time slots in each frame is, at least, equal to the number of nodes in the network.

- r is the maximum back-off (waiting) time.

Now, assuming that the system is in frame $j \geq 0$, we define the following:

- $\mathbf{X}_j = (X_{j,0}, X_{d_j}, X_{j,1}, X_{j,2}, \ldots, X_{j,r})$ is the state vector which collects the system information, where

  $X_{j,0}$ is the number of sensors with a reserved slot.

  X_{d_j} is the number of sensors in the discover phase.

  $X_{j,s}$ is the number of sensors which will wait $s \in \{1, \ldots, r\}$ frames more.

- Y_j is the number of sensors that successfully get a slot.

- $Z_{j,s}$ is the number of sensors that collided in frame j and chose to wait $s \in \{0, 1, \ldots, r\}$ frames. The vector $\mathbf{Z}_{j+1} = (Z_{j+1,1}, \ldots, Z_{j+1,r})$ is used to record results of the random choice for back-off time from a multinomial distribution with parameters $X_{d_j} - Y_{j+1}$ and $\boldsymbol{\pi}$, where $\boldsymbol{\pi}$ is an r dimensional vector corresponding to a uniform distribution, i.e. $\boldsymbol{\pi} = (\frac{1}{r}, \ldots, \frac{1}{r})$. Based on various kinds of back-off time selection strategies, however, $\boldsymbol{\pi}$ can be an arbitrary probability vector.

Hence, we have the basic identity

$$X_{j,0} + X_{d_j} + \sum_{s=1}^{r} X_{j,s} = n, \text{ for } j \geq 0,$$

as the sensors can only be in the active phase ($X_{j,0}$), the discover phase (X_{d_j}) or the wait phase ($\sum_{s=1}^{r} X_{j,s}$). With the variables defined above, we are able to capture the dynamics of the process.

Algorithm 1 The LMAC simulation algorithm.

Require: $\mathbf{X}_0 := (0, n, 0, 0, \ldots, 0)$, $Y_0 := 0$, $j := 0$
  1: **repeat**
  2:     Generate $Y_{j+1} \leftarrow \mathbb{P}_{t-X_{j,0}, X_{d_j}} (Y = y)$
  3:     $X_{j+1,0} \leftarrow X_{j,0} + Y_{j+1}$
  4:     Generate $\mathbf{Z}_{j+1}$ from multinomial distribution
  5:     **for** $s = 1$ to $r - 1$ **do**
  6:         $X_{j+1,s} \leftarrow X_{j,s+1} + Z_{j+1,s}$
  7:     **end for**
  8:     $X_{j+1,r} \leftarrow Z_{j+1,r}$
  9:     $X_{d_j+1} \leftarrow X_{j,1}$
 10:     $j \leftarrow j + 1$
 11: **until** $X_{j+1,0} = n$

The dynamic is driven by frames as the basic time unit where the vector Xj is used to record the system information. The network starts when all the sensors are unsynchronised and are attempting to get (unreserved) slots. As time elapses, an increasing number of sensors get a reserved slot. Eventually, the system stabilises when all of the nodes have a reserved time slot. The whole process is modelled as a discrete–time Markov chain and the total time spent on the stabilisation process is discrete phase–type distributed [Neu75, Neu81]. The absorbing state of the underlying Markov chain is the state where all of the sensors have their reserved slots. Algorithm 1 illustrates the dynamics by pseudo code. Here, we identify the initial state of the system under the worst case, where all the sensors are trying to get their slots at the first frame. Indeed, the worst case gives the highest likelihood for the sensors to experience collisions at the beginning.

Table 3.1: Example for Algorithm 1 considering $r = 3$ and $n = 3$.

Frame j	Y_j	$Z_{j,1}$	$Z_{j,2}$	$Z_{j,3}$	$X_{j,0}$	X_{d_j}	$X_{j,1}$	$X_{j,2}$	$X_{j,3}$
0	0				0	(3)	0	0	0
1	0	0	2	1	0	0	0	2	1
2	0	0	0	0	0	0	2	1	0
3	0	0	0	0	0	2	1	0	0
4	0	2	0	0	0	1	2	0	0
5	1	0	0	0	1	2	0	0	0
6	0	1	1	0	1	0	1	1	0
7	0	0	0	0	1	1	1	0	0
8	1	0	0	0	2	1	0	0	0
9	1	0	0	0	3	0	0	0	0

Table 3.1 exemplifies one scenario of the network dynamics of our abstracted model following Algorithm 1, where $r = 3$ and $n = 3$. The initial data vector is $(0,0,0,0,0,3,0,0,0)$, where 3 sensors are in the discover phase, X_{d_0}. Unluckily, all of the three collide (2 sensors wait 2 frames and 1 sensor waits 3 frames), which is depicted by the second line of the table. Thereafter, time just passes and the sensors are waiting their turn to retry. In the end, all the sensors successfully get reserved time slots. Thereby, the number of sensors having reserved slots, $X_{j,0}$, becomes 3.

3.1.3.2 Analysis of randomness

Two probabilistic choices occur in step 2 and 4 of Algorithm 1. In the following, we shall formalise and calculate both assuming that the system is in arbitrary frame j.

Probability distribution of slot selection In step 2 of Algorithm 1, we need to find the random number Y_{j+1}, which is the number of new sensors with reserved slots. Note that Y_{j+1} could be any integer value in the interval $[0, X_{d_j}]$ except $X_{d_j} - 1$, which has a probability distribution described below.

To shorten notation, we define some new variables to assist us in calculating probabilities derived from the state vector $\mathbf{X}_j$.

- l is the number of current free slots, i.e. $l = t - X_{j,0}$,the number of current free slots is equal to the total available number of slots minus the number of reserved slots.

- k is the number of sensors in the discover phase, i.e. $k = X_{d_j}$ and $k \leq l$.

- y is the number of new sensors with reserved slots, i.e. $0 \leq y \leq k$.

- x is the number of sensors experiencing collision, i.e. $x = k - y$.

- h is the number of unreserved slots, i.e. $h = l - y$.

The probability distribution of slot selection (i.e. distribution of Y) depends only on l, k, and y. Let $\mathbb{P}_{l,k}(Y = y)$ denote the probability that y sensors successfully get reserved slots, given the condition that there are l free time slots and k attempting sensors. Based on combinatorial theory, $\mathbb{P}_{l,k}(Y = y)$ is calculated by a rational function with parameters l, k, and y. The numerator counts the

number of combinations where y sensors are reserved, and the denominator is the number of combinations for all possible values of Y.

When none of sensors collide, i.e $y = k$, we have that

$$\mathbb{P}_{l,k}(Y = k) = \frac{1}{l^k}\binom{l}{0}\frac{h!}{(h-0)!}\frac{k!}{x!} = \frac{1}{l^k}\binom{k}{x}.$$

Obviously, $\mathbb{P}_{l,k}(Y = k-1) = 0$ since it is impossible to have collision involving just one sensor.

If 2 or 3 sensors are in collision, they can only collide in a single slot. However, for 4 or more sensors the collision can happen in more than one time slot. To consider a general case, we have the formula for $y \geq 2$:

$$\mathbb{P}_{l,k}(Y = y) = \frac{1}{l^k}\binom{l}{y}\sum_{i=1}^{\lfloor\frac{x}{2}\rfloor}\frac{h!}{(h-i)!}k!p_i(x) = \frac{1}{l^k}\binom{k}{x}\sum_{i=1}^{\lfloor\frac{x}{2}\rfloor}\frac{l!}{(l-y-i)!}x!p_i(x),$$

$$(3.1)$$

where

- $\lfloor\cdot\rfloor$ is the floor function. The number of slots with collided sensors can vary from 1 up to $\lfloor\frac{x}{2}\rfloor$. E.g. if we have 7 collided sensors, these can be in at most $\lfloor\frac{7}{2}\rfloor$=3 slots.

- $p_i(x)$ is an iterative function that depends on x and i. The way of finding the explicit form of this function is described in Appendix A.1.

- $k!p_i(x)$ is the number of ways of putting x sensors into i slots, and these x sensors are under collision.

Thus, from Formula (3.1), we are able to compute the distribution of slot selection analytically.

Probability distribution of back-off time   Intuitively, to shorten the setup time we give high priority to small back-off times but only in the case where there are only few sensors in the network [VR09]. If there are a considerable number of sensors, it becomes necessary to have larger back-off time options to reduce the probability of collision. Therefore, there exists an optimistic strategy which chooses the probability distribution(s) of the back-off time such that the expected time till absorption is minimum.

In [vHH06] network latency is taken into account. Four types of strategy to ensure a low latency for the most common data traffic in WSN are proposed.

We chose a uniform distribution of the back-off times, which is one of the four classic strategies mentioned in [vHH06].

3.1.3.3 Discrete Time Markov Chain

A Markov chain is a discrete state space stochastic process with the Markov property (cf. Definition 2.1). Since the frames are taken as the time unit, the model of LMAC is a DTMC. Let $E = \{1, 2, \ldots, \binom{n+r+1}{n}\}$ denote the state space of the underlying DTMC with transition matrix defined as follows

$$\mathbf{P} = (p_{m,m'})_{m,m' \in E}. \tag{3.2}$$

Now, let M be an injective function which maps state vectors $\mathbf{X}_j$, $j \geq 0$, to E in the following way

$$M(\mathbf{X}_j) = \sum_{s=0}^{X_{d_j}} \binom{n+r-s}{s} - \sum_{s=1}^{r} \sum_{i=X_{j,s}+1}^{a} \binom{a+r-s-i}{r-s},$$

where $a = n - X_{d_j} - \sum_{q=1}^{s-1} X_{j,q}$. Note that other mappings could have been used.

Now, suppose that $M(\mathbf{X}_j) = m$ and $M(\mathbf{X}_{j+1}) = m'$, the (m, m')-th element of $\mathbf{P}$ is computed as follows

$$
\begin{aligned}
p_{m,m'} &= \mathbb{P}(\mathbf{X}_{j+1}|\mathbf{X}_j) \\
&= \mathbb{P}_{t-X_{j,0},X_{d_j}}(Y = X_{j+1,0} - X_{j,0}) \\
&\quad \cdot \binom{X_{d_j} - (X_{j+1,0} - X_{j,0})}{(X_{j+1,1} - X_{j,2}),(X_{j+1,2} - X_{j,3}),\ldots,(X_{j+1,r-1} - X_{j,r}), X_{j+1,r}} \\
&\quad \cdot \left(\frac{1}{r}\right)^{X_{d_j} - (X_{j+1,0} - X_{j,0})}.
\end{aligned}
$$

As we can see, the transition probability from the state m to the state m', consists of two parts. The first part, $\mathbb{P}_{t-X_{j,0},X_{d_j}}(Y = X_{j+1,0} - X_{j,0})$, is the probability that $X_{j+1,0} - X_{j,0}$ sensors get reserved slots. The remaining part of the formula calculates the probability that $X_{d_j} - (X_{j+1,0} - X_{j,0})$ collided sensors back-off, which is multinomially distributed. Since the discovering sensors can either become reserved or back-off, the multiplication of these two probabilities gives the overall transition probability.

Xdj	X1	X2	M()	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
		Xdj		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	2	2	2	3
		X1		0	0	0	0	1	1	1	2	2	3	0	0	0	1	1	2	0	0	1	0
		X2		0	1	2	3	0	1	2	0	1	0	1	2	0	1	0	0	1	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	2	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	3	4	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	2	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	2	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	2	1	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	3	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	12	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	2	13	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	14	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	1	1	15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	2	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
2	0	0	17	0.67	0	0.08	0	0	0.17	0	0.08	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	18	0	0	0	0	0.75	0	0.06	0	0.13	0.06	0	0	0	0	0	0	0	0	0	0
2	1	0	19	0	0	0	0	0	0	0	0	0	0	0.75	0	0.06	0	0.13	0.06	0	0	0	0
3	0	0	20	0.38	0	0.14	0.01	0	0.28	0.02	0.14	0.02	0.01	0	0	0	0	0	0	0	0	0	0

Figure 3.3: Example of the transition matrix $\mathbf{P}$ for $n = 3$, $r = 2$, and $t = 4$.

With the above formulae, we have built a DTMC analytically using combinatorics to represent the dynamics at the LMAC setup phase. Fig. 3.3 shows an example of a DTMC considering the parameters $n = 3$, $r = 2$, and $t = 4$. With our proposed mapping function M, the state vector is ordered with priority from X_{d_j} until X_r. The initial state of the network is defined with all sensors in the discover phase at the first frame, corresponding to the last row in the transition matrix. The absorbing state of the DTMC is state number 1 at the top row.

3.1.3.4 Simulation Study

In this part, we present the result of a simulation study using Algorithm 1. The network is configured with 4 sensors, 5 time slots, and 2 as maximum back-off time. We have made statistical inference on the example, where the standard derivations we obtained indicate the reliability of the estimation from our simulation.

Besides the purpose of verifying the analytical model, simulation also provides an optional way of computing probabilities in the case when the model has high order or stiff systems are hard to be computed analytically.

3.1.4   Optimisation Using Phase–type

In Section 3.1.3 we have modelled LMAC by a DTMC using a direct mathematical abstraction. Now, we will investigate optimal parameter settings of the system in order to minimise the time for stabilisation of the WSN. In particular, we will analyse the stabilisation time when adding a minimal amount of excess capacity.

The critical property of interest is the expected time to stabilisation. By definition, the expectation of a discrete random variable is calculated by

$$\mathbb{E}(W) = \sum_{w=1}^{\infty} w\mathbb{P}(W = w), \tag{3.3}$$

where in our case w represents the frame number, and $\mathbb{P}(W = w)$ is the probability of absorption occurring exactly at frame w. The formula contains an infinite sum, therefore it is hard to compute the true value without truncation. Thus, an iterative method is required to guarantee the convergence. However, it is very costly to obtain $\mathbb{P}(W = w)$ as many vector-matrix multiplications are required. We now present an alternative and simplified method to obtain (3.3).

It is easy to see that the time till absorption of the underlying DTMC follows a discrete phase–type distribution (cf. Section 2.1.2). Using DPH distributions in our analysis gives computational advantages. By assuming that there is at least the same number of time slots as the number of sensors, we are able to guarantee absorption. We rewrite the transition matrix $\mathbf{P}$ of Formula (3.2) into the DPH form

$$\mathbf{P} = \begin{bmatrix} \mathbf{T} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}.$$

Thus, the expectation can therefore be computed by

$$\mathbb{E}(W) = \boldsymbol{\pi}(\mathbf{I} - \mathbf{T})^{-1}\mathbf{e}, \tag{3.4}$$

where $\boldsymbol{\pi}$ is the initial probability vector (cf. Section 2.1.2, Formula 2.4). The complexity of computing Formula 3.4 can be bounded by $\mathcal{O}(n^2 + n^{2.376})$. Using the Coppersmith-Winograd algorithm [CW90], a matrix inversion has the complexity $\mathcal{O}(n^{2.376})$.

Moreover, the variance of a random variable is defined as

$$Var(W) = \mathbb{E}(W^2) - \mathbb{E}(W)^2, \tag{3.5}$$

where $\mathbb{E}(W^2)$ is computed by the second factorial moment of a DPH distributed random variable (cf. Section 2.1.2, Formula 2.4) given by $\mathbb{E}(W^2) = \mathbb{E}(W(W - 1)) + \mathbb{E}(W)$, such that $\mathbb{E}(W(W - 1)) = 2\boldsymbol{\pi}(\mathbf{I} - \mathbf{T})^{-2}\mathbf{e}$.

As we can see in this analysis, DPH distributions take the role as computational vehicle in solving optimisation problems.
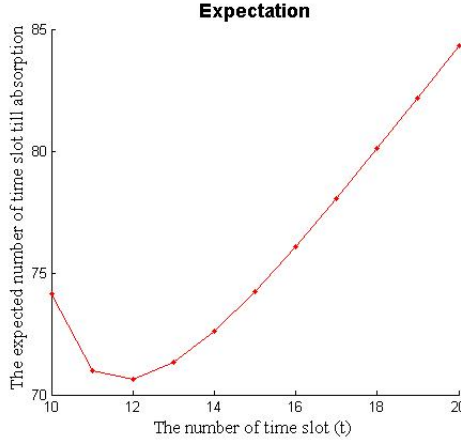
Figure 3.4: Choosing the optimal number of time slots for LMAC considering $n = 10$, $r = 2$.

Fig. 3.4 illustrates one optimisation problem regarding the number of time slots. Here, the network contains 10 sensors and the maximum back-off time is 2. In this case, 12 time slots provide the best possible stabilisation rate, as it has the lowest expected stabilisation time.

Fig. 3.4 describes how to pick the optimal number of time slots given the number of sensors and the maximum back-off time. By having this way of finding the optimal number of time slots, Fig. 3.5 depicts the growing trend by pairing the number of sensors and optimised number of time slots. Here, the maximum back-off time is also 2. The result can be divided in linear segments, where the gap between the sensor number and the slot number increases in larger networks (i.e. when the network has more sensors), because the probability of collision increases when more sensors are in the network. Therefore, Fig. 3.5 provides a guide for network designers to decide how to match the sensor number with the slot number in an optimal setting. For instance, 17 sensors should match 20 time slots to be optimal with maximum back-off time being 2.

On the way of computing the plot, the state space is growing by having an increasing number of sensors. To deal with higher order models, we switched the
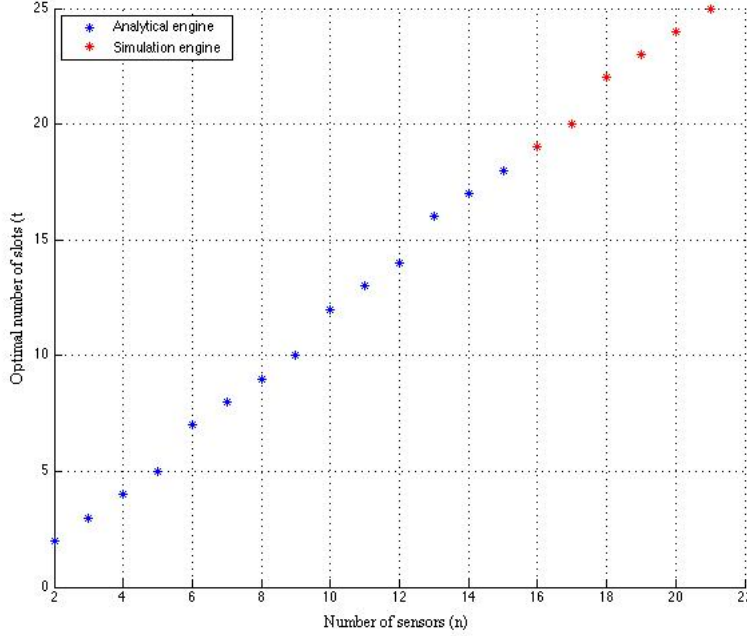
Figure 3.5: Matching the number of sensors with the optimal number of time slots considering $r = 2$.

computational engine from analytical to simulation at the point where we had around 1000 states (i.e. 16 sensors). In Fig. 3.5, we distinguish the two engines by colours. Note, here we only intend to show how we used the simulation engine to assist the analytical engine in deriving solutions. Therefore, the point of switching would depend on your local computing power.

Furthermore, our abstracted model is able to analyse modification issues. In a predefined LMAC network, the number of sensors, the number of time slots, and the maximum back-off are identified. For instance, sometimes it is necessary to add or remove a certain number of sensors in the current network. A question that may arise is how to adjust the parameters to keep the stabilisation at an optimal speed. Fig. 3.6 offers some recommendations regarding these issues.

It is clear that more time should be expected if there are more sensors in the network. However, the marginal cost varies due to the fluctuation of the probability of collision. Fig. 3.6 has been plotted given 20 time slots with maximum back-off time being 2. The vertical axis describes the incremental cost for plugging in
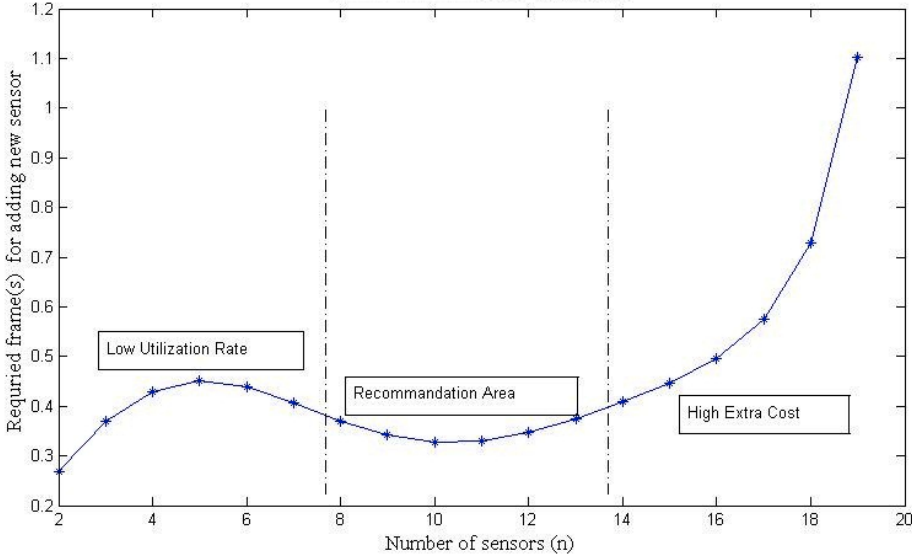
Figure 3.6: Matching the number of sensors with the optimal number of time slots considering $r = 2$.

one extra sensor. Because of low utilisation rate of the network, i.e. extremely low probability for collision, the price for incrementing sensors is higher at the beginning. For example, the probability for 2 sensors in collision is only 0.05. But the probabilities for 3, 4 and 5 sensors in collision increase to 0.15, 0.27 and 0.42 respectively.

For more than 14 sensors in the network, the cost rises dramatically. Therefore, the recommended number of sensors for the current configuration should remain in the middle region. If the required number of sensors has a high additional cost, it would be better to increase the number of time slots or the maximum back-off time in order to keep efficiency. In contrast, the number of time slots or the maximum back-off time should be reduced in order to raise the network utilisation rate. Evidently, there are many other questions about optimisation that can be solved. For instance, to find the optimal number of sensors given a fixed maximum back-off time and a fixed number of time slots.

3.1.5   State Space Explosion

In some experiments, state space explosion emerges considering above 10,000 states even though the sparseness of the transition matrix of the underlying

DTMC increases as well. With the MATLAB sparse matrix representation, the number of states in our model, given by $\binom{n+r+1}{n}$, crashed considering 10660 states (38 sensors and 2 maximum back-off) on 2GB RAM memory. For maximum 3 and 4 frames waiting time, we are up to 19 and 15 sensors, respectively. Note that since we work on a relatively old computing platform, we expect our approach can handle a larger number of states than our experimental data.
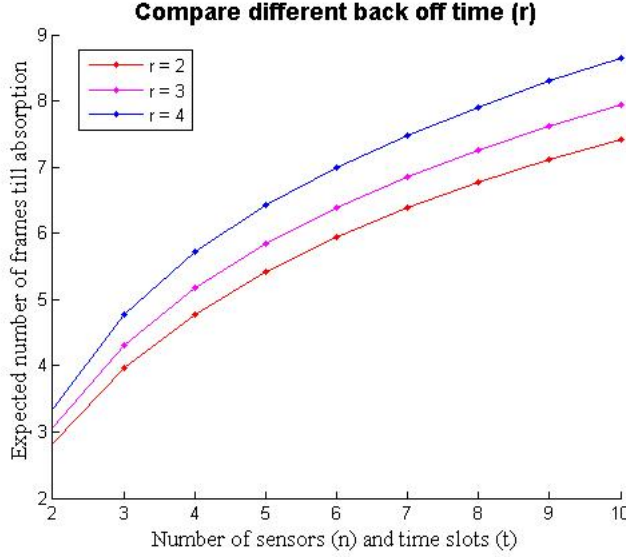
Figure 3.7: Measure different maximum back-off time.

Even though the current result is limited due to the state explosion problem, a pioneer study on different maximum waiting has been depicted in Fig. 3.7. In Fig. 3.7, the number of sensors and time slots is the same in all the sample points. It clarifies that maximum 2 frames waiting is the most favourable choice comparing with the cases of 3 frames and 4 frames given a number of sensors up to 10. It supposes that the favourable choice will switch to 4 frames after reaching a certain point because of the reduction of the probability of collision. Part of our future work will be to extend the current plot, i.e. to consider more sensors and time slots.

3.1.6   LMAC Analyser

Based on the results described in the previous sections, we have created a tool, called LMAC analyser, which is a prototype on the MATLAB platform. Fig. 3.8

is a screen shot of the running prototype. Users are required to input the desired LMAC parameters (the number of sensors, the number of time slots, and the maximum back-off time), and the tool will automatically output a probabilistic model, in the form of a probability transition matrix of a DTMC. The expected time to stabilisation is also calculated.

3.1.7   Summary

In this work, we have reported a probabilistic model for the analysis of a medium access protocol LMAC for WSN. The model abstracts the concurrent behaviour of the setup process in LMAC, where individual sensors are allocated to time slots, preferably in an efficient way to minimise the price. There are two sources of randomness in this concurrent stochastic process. First, when the nodes randomly pick time slots, and second when the collided nodes randomly pick back-off time. Considering the worst case scenario, we resolved the non-determinism, which comes from the underspecification of the protocol by probabilistic choices. The probabilities are calculated using combinatorial theory with a uniformity assumption. Moreover, our simulation engine verifies the analytical model, and it can optionally compute solutions whenever numerical challenges appear for the analytical model. After obtaining the model, we have analysed the performance of the process and we have calculated optimal parameters for network configuration.

Contrary to previous work [FvHM07, vH07, VR09], our contribution is an alternative approach of system abstraction, by which parallel composition can be avoided. The inherent advantage is that no detailed model is required in order to do the abstraction. Thereby, it is possible to handle larger systems directly. Note that our model has a moderate state space $\binom{n+r+1}{n}$, which depends only on the number of sensors n and the maximum back-off time r.

3.2   Phase–Type Process Algebra

3.2.1   Introduction

In recent years, there have been many works on model checking concurrent stochastic systems. The mathematical fundamentals for modelling stochastic aspects of system behaviour are based on Markov chains. In the context of composite stochastic modelling framework, the stochastic process algebras (SPAs)

```
Command Window                                                    →| □ ⤢ ×

  **************************************************************
  * LMAC Protocol Analyzer: version 1.0                        *
  *                                                            *
  * LMAC model generator: in discrete time Markov chain        *
  * Expectation caculator: expected time till absorption       *
  *                                                            *
  * Kebin Zeng, Luz Esparza, Bo Nielsen                        *
  * DTU Informatics, keze@imm.dtu.dk, Dec. 2010                *
  **************************************************************
  **************************************************************
  * Usage: MATLAB function: [P] LMAC(r,n,t)                    *
  * INPUT:                                                     *
  * r : maxmium back off time {2,3,4}                          *
  * n : number of sensors                                      *
  * t : number of time slots                                   *
  * OUTPUT:                                                    *
  * P : transition matrix of DTMC in file p.mat                *
  * Par : parameters of LMAC model in file p_par.mat           *
  **************************************************************
  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
  LMAC model has been successfully generated in file p.mat.

  Matrix size: [1001,1001]
  LMAC Parameters:
          maximum back off time (r) = 3
              number of sensors (n) = 10
            number of time slots (t) = 12
  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
  <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
  Expectation of time till absorbing of LMAC

  LMAC Parameters:
          maximum back off time (r) = 3
              number of sensors (n) = 10
            number of time slots (t) = 12

  The expectation (in number of time slot) is 78.359887
      The expectation (in number of frame) is 6.529991e+000
  <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
fx >> |
```

Figure 3.8: Screen shot of the LMAC Analyser.

have been a solid research field on formal modelling and analysis. The promi-
nent SPAs, such as TIPP [GHR92], stochastic π-calculus [Pri95], PEPA [Hil96],

EMPA [BG96] and IMC [Her02], are all based on the delay between states to
be exponential (EXP) distributed, which ensures that the underlying stochastic
process is a continuous–time Markov chain (cf. Section 2.2.1, Definition 2.7).
However, this restriction to the EXP duration is unrealistic when modelling
many phenomena such as traffic sources and system biology, where the system
may include deterministic delays or heavy-tail distributed quantities. Therefore,
we focus on generalising the EXP duration.

Continuous phase–type distributions are distributions of the time until absorp-
tion in a CTMC (cf. Section 2.2.2, Definition 2.8). EXP, Erlang, Hypo-EXP,
Hyper-EXP and Coxian distributions are examples of CPH distributions. CPH
distributions can be used to approximate any kind of probability distributions
on $[0, \infty)$ [JT88]. Statistical inference tools are available for fitting both DPH
and CPH distributions, e.g. EMPHT [ANO96] and PhFit [HT02]. In partic-
ular, well behaved distributions are easily fitted by CPH distributions with a
moderate number like 3-6 transient states [Asm00]. However, for complicated
systems, especially those with concurrency, often it is impossible to fit CPH dis-
tributions for the whole system directly. Therefore, we take the widely accepted
approach to formally model concurrent stochastic systems in SPAs. Using SPAs,
an overall CPH distributed delay could be constructed from the simple CPH dis-
tributed random variables as building blocks through different operators given
proper operational rules.

To serve a sound mathematical construction, we concentrate on the closure prop-
erties of CPH distributions, especially their algebraic laws. We then introduce
Phase–type Process Algebra (PHPA), which takes CPH distributed random vari-
ables as primitives, such that each random variable has its corresponding CPH
representation. According to the closure properties of CPH distributions, PHPA
has several operators to operate on CPH distributed random delays. In this
thesis we focus on the purely stochastic behaviour raised by CPH distributions
without considering nondeterminism.

Related Work Phase Type Processes (PTPs), introduced in [Wol08], are
defined as a generalisation of IMC, where the Markov transition relation in
IMC [Her02] is extended to have PH distributed random delay. The successive
states are decided by an *independent* discrete probability distribution, following
each PH distributed system delay. The main contribution in this development
is bisimulation equivalence and parallel composition for PTPs. Since PTPs take
PH distributed random variables to represent system delays, it maintains the
usual interleaving semantics in CTMCs.

El-Rayes et al. in [ERKN99] introduce $\text{PEPA}_{ph}^{\infty}$, a stochastic process algebra

extended from Hillston's PEPA. The activities of $\text{PEPA}_{ph}^{\infty}$ components have durations given by PH distributed random variables. The $\text{PEPA}_{ph}^{\infty}$ focuses on evaluating the performance of queueing systems, where the matrix-geometric method is employed to solve the steady state probabilities of an infinite state model.

A stochastic calculus, Cox & Convenience Calculus, is proposed by R. Pulungan in [Pul09] having Acyclic Phase–type (APH) distributed system delay. APH distributions constitute a subclass of PH distributions with triangular matrix representations, such that any APH has an equivalent Coxian representation. Applying the reduction algorithm for APH representations, the calculus is able to define the processes having almost surely minimal state space. The reason behind "almost surely minimal state space" comes from the awareness of potential smaller representation of matrix–exponential distributions [AB99]. We shall clarify this point when we introduce matrix–exponential distributions in Chapter 5.

3.2.2   Closure Properties

In contrast to EXP distributions (which are only closed under the minimum), the class of CPH distributions is closed under a number of operations [AL82]. The closure properties of CPH distributions guarantee that we stay in the class of CPH distributions with several types of operations.

Let $\text{CPH}_m(\boldsymbol{\pi},\mathbf{T})$ denote a CPH representation such that m is the order of the phase–type representation, and $\text{DPH}_m(\boldsymbol{\pi},\mathbf{T})$ denote a DPH representation with m is the order of the representation. To have great generality, in this work, i.e. Section 3.2, we consider the probability mass at zero does not have to be zero, i.e. $\pi^0 \neq 0$ is possible. Now, we introduce the closure properties of CPH distributions.

1. Sum of independent CPH variables
   Given two random variables W_1 and W_2 with representations $\text{CPH}_m(\boldsymbol{\pi}_1,\mathbf{T})$ and $\text{CPH}_n(\boldsymbol{\pi}_2,\mathbf{S})$ respectively, then the random variable $W = W_1 + W_2$ follows a phase-type distribution with the representation $\text{CPH}_{m+n}(\boldsymbol{\pi}_3, \mathbf{L})$ given by $\boldsymbol{\pi}_3 = (\boldsymbol{\pi}_1, \pi_1^0\boldsymbol{\pi}_2)$, and

$$\mathbf{L} = \begin{bmatrix} \mathbf{T} & \mathbf{t}\boldsymbol{\pi}_2 \\ \mathbf{0} & \mathbf{S} \end{bmatrix}. \tag{3.6}$$

PROPOSITION 3.1 *The convolution of a finite number of continuous phase–type distributed random variables is itself continuous phase–type distributed.*

2. Finite order statistics (min, max)

   Given two CPH distributed random variables W_1 and W_2 with representations $\text{CPH}_m(\boldsymbol{\pi}_1, \mathbf{T})$ and $\text{CPH}_n(\boldsymbol{\pi}_2, \mathbf{S})$ respectively, we have $W_{min} = \min(W_1, W_2)$ is CPH distributed with the representation $\text{CPH}_{mn}(\boldsymbol{\pi}_3, \mathbf{L})$ given by $\boldsymbol{\pi}_3 = \boldsymbol{\pi}_1 \otimes \boldsymbol{\pi}_2$, and

   $$\mathbf{L} = \mathbf{T} \otimes \mathbf{I}_n + \mathbf{I}_m \otimes \mathbf{S} = \mathbf{T} \oplus \mathbf{S}. \tag{3.7}$$

   We have $W_{max} = \max(W_1, W_2)$ is CPH distributed with the representation $\text{CPH}_{mn+m+n}(\boldsymbol{\pi}_3, \mathbf{L})$ given by $\boldsymbol{\pi}_3 = (\boldsymbol{\pi}_1 \otimes \boldsymbol{\pi}_2,\ \boldsymbol{\pi}_1 \pi_2^0,\ \pi_1^0 \boldsymbol{\pi}_2)$, and

   $$\mathbf{L} = \begin{bmatrix} \mathbf{T} \oplus \mathbf{S} & \mathbf{I}_m \otimes \mathbf{s} & \mathbf{t} \otimes \mathbf{I}_n \\ \mathbf{0} & \mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S} \end{bmatrix}. \tag{3.8}$$

PROPOSITION 3.2 *The finite minimum and maximum of continuous phase–type distributed random variables is itself continuous phase–type distributed.*

3. Finite mixture of CPH distributions
   Let k be a finite integer, W_i be a CPH distributed random variable with representation $\text{CPH}_{m_i}(\boldsymbol{\pi}_i, \mathbf{T}_i)$ respectively, and let $\mathrm{W} = I_i W_i$ such that $\sum_{i=1}^{k} I_i = 1$ and $\mathrm{P}(I_i = 1) = p_i$. The random variable W is itself phase-type distributed with the representation $\text{CPH}_{m_1+m_2+\cdots+m_k}(\boldsymbol{\pi}, \mathbf{L})$ given by $\boldsymbol{\pi} = (p_1 \boldsymbol{\pi}_1, p_2 \boldsymbol{\pi}_2, \ldots, p_k \boldsymbol{\pi}_k)$, and

   $$\mathbf{L} = \begin{bmatrix} \mathbf{T}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{T}_k \end{bmatrix}. \tag{3.9}$$

PROPOSITION 3.3 *Any finite convex mixture of continuous phase–type distributed random variables is itself continuous phase-type distributed.*

4. Random sum
   Let W_1 be a discrete PH distributed random variable with representation $\text{DPH}_m(\boldsymbol{\pi}_1, \mathbf{T})$ and W_2 be CPH distributed random variable with representation $\text{CPH}_n(\boldsymbol{\pi}_2, \mathbf{S})$, the random variable $W = \underbrace{W_2 + \cdots + W_2}_{W_1}$ follows

a CPH distribution with the representation $\text{CPH}_{mn}(\boldsymbol{\pi}_3, \mathbf{L})$ given by $\boldsymbol{\pi}_3 = (\boldsymbol{\pi}_1 \otimes \boldsymbol{\pi}_2)$, and

$$\mathbf{L} = \left[\mathbf{I}_n \otimes \mathbf{S} + \mathbf{T} \otimes \mathbf{s}\boldsymbol{\pi}_2 \right]. \tag{3.10}$$

PROPOSITION 3.4 *The continuous phase–type distributed random variable compounding with discrete phase-type distributed random variable is itself continuous phase–type distributed.*

It is worth mentioning that discrete phase–type distributions have the most closure properties with minor modifications, see e.g. [Neu75].

3.2.3   Race Condition for Phase–type Distributions

In CTMCs, the *race condition* is known as for multiple outgoing transitions the fastest one will determine the completion time. Hence, the overall delay caused by race condition is determined by the minimum of all participating EXP distributed random variables, which is still EXP distributed with the parameter as the sum of all the outgoing rates.
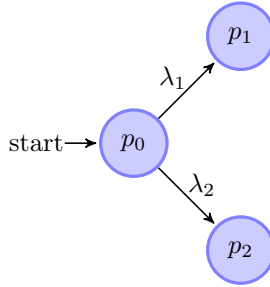
Figure 3.9: Race condition in CTMCs.

Given the probability density function of an EXP distribution $f(w) = \lambda e^{-\lambda w}$; let W_1 and W_2 be two EXP distributed random variables with parameters λ_1 and λ_2 respectively, and let $\mathbf{1}(W_1 < W_2)$ is an indicator function, which equals 1 when $W_1 < W_2$. We have the basic formula

$$\mathbf{1}(W_1 < W_2)\mathbb{P}(W_1 < W_2) + \mathbf{1}(W_1 > W_2)\mathbb{P}(W_1 > W_2) = 1,$$

such that $\mathbb{P}(W_1 = W_2) = 0$ in continuous–time, and the probability of $W_1 <$

W_2 is given as

$$
\begin{aligned}
\mathbb{P}(W_1 < W_2) &= \int_0^\infty \mathbb{P}(W_2 > W_1 | W_1 = w) f_1(w) dw \\
&= \int_0^\infty \mathbb{P}(W_2 > w) \lambda_1 e^{-\lambda_1 w} dw = \int_0^\infty e^{-\lambda_2 w} \lambda_1 e^{-\lambda_1 w} dw \\
&= \lambda_1 \left[-\frac{1}{\lambda_1 + \lambda_2} e^{-(\lambda_1 + \lambda_2)w} \right]_0^\infty = \frac{\lambda_1}{\lambda_1 + \lambda_2}
\end{aligned}
\tag{3.11}
$$

The probability stated in Formula (3.11) is known as the probability that the transition from p_0 to p_1 in Fig. 3.9 to win the race. In race condition for EXP distributions, the random delays, i.e. W_1 and W_2, and the winning variable, i.e. $\mathbf{1}(W_1 < W_2) = 1$ or $\mathbf{1}(W_1 > W_2) = 1$, are independent.

Turning our attention to CPH distributions, let a *CPH transition* be a transition labelled by a CPH distributed random variable, the race condition carries over for multiple outgoing CPH transitions. In analogy with EXP distributions, the overall delay of the race condition is given as the minimum of the participating CPH distributions, which follows Formula (3.7).
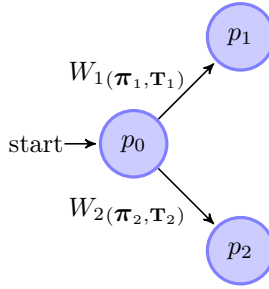
Figure 3.10: Race condition for phase–type distributed random variables.

Given the probability density function of CPH (cf. Formula (2.11)), and let $W_{1(\boldsymbol{\pi}_1, \mathbf{T}_1)}$ (W_1 for short), $W_{2(\boldsymbol{\pi}_2, \mathbf{T}_2)}$ (W_2 for short) be independent CPH distributed random variables with representations $(\boldsymbol{\pi}_1, \mathbf{T}_1)$ and $(\boldsymbol{\pi}_2, \mathbf{T}_2)$ respectively, we have again the basic formula

$$
\mathbf{1}(W_1 < W_2) \mathbb{P}(W_1 < W_2) + \mathbf{1}(W_1 > W_2) \mathbb{P}(W_1 > W_2) = 1,
$$

such that we shall show how to compute $\mathbb{P}(W_1 < W_2)$. To clarify that, we assume readers are familiar with the standard properties of Kronecker operations (see

Appendix A.2 for reference).

$$
\begin{aligned}
\mathbb{P}(W_1 < W_2) &= \int_0^\infty \mathbb{P}(W_2 > W_1 | W_1 = w) f_1(w) dw \\
&= \int_0^\infty \mathbb{P}(W_2 > w) \boldsymbol{\pi}_1 e^{\mathbf{T}_1 w} \mathbf{t}_1 dw \\
&= \int_0^\infty \boldsymbol{\pi}_2 e^{\mathbf{T}_2 w} \mathbf{e} \boldsymbol{\pi}_1 e^{\mathbf{T}_1 w} \mathbf{t}_1 dw \\
&= \int_0^\infty \left(\boldsymbol{\pi}_2 e^{\mathbf{T}_2 w} \mathbf{e} \right) \otimes \left(\boldsymbol{\pi}_1 e^{\mathbf{T}_1 w} \mathbf{t}_1 \right) dw \\
By \quad & (A_1 A_2 A_3) \otimes (B_1 B_2 B_3) = (A_1 \otimes B_1)(A_2 \otimes B_2)(A_3 \otimes B_3) \\
&= \int_0^\infty (\boldsymbol{\pi}_2 \otimes \boldsymbol{\pi}_1)(e^{\mathbf{T}_2 w} \otimes e^{\mathbf{T}_1 w})(\mathbf{e} \otimes \mathbf{t}_1) dw \\
&= (\boldsymbol{\pi}_2 \otimes \boldsymbol{\pi}_1) \int_0^\infty (e^{\mathbf{T}_2 w} \otimes e^{\mathbf{T}_1 w}) dw \ (\mathbf{e} \otimes \mathbf{t}_1) \\
By \quad & e^{\mathbf{A}_1} \otimes e^{\mathbf{A}_2} = e^{\mathbf{A}_1 \oplus \mathbf{A}_2} \\
&= (\boldsymbol{\pi}_2 \otimes \boldsymbol{\pi}_1) \int_0^\infty e^{(\mathbf{T}_2 \oplus \mathbf{T}_1) w} dw \ (\mathbf{e} \otimes \mathbf{t}_1) \\
By \quad & \int_0^\infty e^{\mathbf{A} w} dw = (-\mathbf{A})^{-1} \\
&= (\boldsymbol{\pi}_2 \otimes \boldsymbol{\pi}_1)(-\mathbf{T}_2 \oplus \mathbf{T}_1)^{-1}(\mathbf{e} \otimes \mathbf{t}_1) \quad\quad (3.12)
\end{aligned}
$$

The closed form stated in Formula (3.12) computes the probability that the transition from p_0 to p_1 in Fig. 3.10 to win such a race between CPH transitions. Since EXP distribution is a subclass of CPH distribution. It is no surprise that Formula (3.12) contains Formula (3.11) for a special case. However, rather than the race condition for EXP distributions, for CPH distributions the random delays, i.e. $W_{1(\boldsymbol{\pi}_1, \mathbf{T}_1)}$ and $W_{2(\boldsymbol{\pi}_2, \mathbf{T}_2)}$, and the winning variable, i.e. $\mathbf{1}(W_1 < W_2) = 1$ or $\mathbf{1}(W_1 > W_2) = 1$, are dependent.

3.2.4 Compositionality of Phase–type Distributions

3.2.4.1 Composite Operators and Algebraic Properties

Having the closure properties of CPH distributions (cf. Section 3.2.2), we are ready to introduce *composite operators for continuous phase–type distributions* and their *algebraic properties*. The composite operators operate on CPH distributed random variables, which are *convolution* ($\sum$) by Proposition 3.1, *minimum* (min) and *maximum* (max) by Proposition 3.2, and *convex mixture* ($+$) by

Proposition 3.3. For these composite operators, we study their algebraic properties, those of commutativity, associativity, idempotence and distributivity.

For the remaining part of this section, let $W_{(\boldsymbol{\pi}, \mathbf{T})}$, $W_{1(\boldsymbol{\pi}_1, \mathbf{T}_1)}$, $W_{2(\boldsymbol{\pi}_2, \mathbf{T}_2)}$, and $W_{3(\boldsymbol{\pi}_3, \mathbf{T}_3)}$ be CPH distributed random variables with representations $(\boldsymbol{\pi}, \mathbf{T})$, $(\boldsymbol{\pi}_1, \mathbf{T}_1)$, $(\boldsymbol{\pi}_2, \mathbf{T}_2)$, and $(\boldsymbol{\pi}_3, \mathbf{T}_3)$ respectively.

THEOREM 3.5 *The convolution $\sum$ of a finite number of continuous phase–type distributed random variables is commutative and associative, but not idempotent.*

PROOF <u>For commutativity.</u> We have

$$W_{1(\boldsymbol{\pi}_1, \mathbf{T}_1)} \sum W_{2(\boldsymbol{\pi}_2, \mathbf{T}_2)} \quad = \quad \bar{W}_{\left((\boldsymbol{\pi}_1, \pi_1^0 \boldsymbol{\pi}_2), \begin{bmatrix} \mathbf{T}_1 & \mathbf{t}_1 \boldsymbol{\pi}_2 \\ \mathbf{0} & \mathbf{T}_2 \end{bmatrix} \right)}.$$

The Laplace–Stieltjes Transform (LST) of above CPH distributed random variable (cf. Formula (2.12)) is

$$
\begin{aligned}
H(s) \quad = \quad & \pi_1^0 \pi_2^0 + (\boldsymbol{\pi}_1, \pi_1^0 \boldsymbol{\pi}_2) \left(s\mathbf{I} - \begin{bmatrix} \mathbf{T}_1 & \mathbf{t}_1 \boldsymbol{\pi}_2 \\ \mathbf{0} & \mathbf{T}_2 \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{t}_1 \pi_2^0 \\ \mathbf{t}_2 \end{bmatrix} \\
= \quad & \pi_1^0 \pi_2^0 + \boldsymbol{\pi}_1 (s\mathbf{I} - \mathbf{T}_1)^{-1} \mathbf{t}_1 \pi_2^0 - \boldsymbol{\pi}_1 (s\mathbf{I} - \mathbf{T}_1)^{-1} \mathbf{t}_1 \boldsymbol{\pi}_2 (s\mathbf{I} - \mathbf{T}_2)^{-1} \mathbf{t}_2 \\
& + \boldsymbol{\pi}_2 (s\mathbf{I} - \mathbf{T}_2)^{-1} \mathbf{t}_2 \pi_1^0.
\end{aligned}
$$

On the opposite direction, we get

$$W_{2(\boldsymbol{\pi}_2, \mathbf{T}_2)} \sum W_{1(\boldsymbol{\pi}_1, \mathbf{T}_1)} \quad = \quad \bar{W}_{\left((\boldsymbol{\pi}_2, \pi_2^0 \boldsymbol{\pi}_1), \begin{bmatrix} \mathbf{T}_2 & \mathbf{t}_2 \boldsymbol{\pi}_1 \\ \mathbf{0} & \mathbf{T}_1 \end{bmatrix} \right)}.$$

The LST of above CPH distributed random variable is

$$
\begin{aligned}
H(s) \quad = \quad & \pi_1^0 \pi_2^0 + (\boldsymbol{\pi}_2, \pi_2^0 \boldsymbol{\pi}_1) \left(s\mathbf{I} - \begin{bmatrix} \mathbf{T}_2 & \mathbf{t}_2 \boldsymbol{\pi}_1 \\ \mathbf{0} & \mathbf{T}_1 \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{t}_2 \pi_1^0 \\ \mathbf{t}_1 \end{bmatrix} \\
= \quad & \pi_1^0 \pi_2^0 + \boldsymbol{\pi}_2 (s\mathbf{I} - \mathbf{T}_2)^{-1} \mathbf{t}_2 \pi_1^0 - \boldsymbol{\pi}_2 (s\mathbf{I} - \mathbf{T}_2)^{-1} \mathbf{t}_2 \boldsymbol{\pi}_1 (s\mathbf{I} - \mathbf{T}_1)^{-1} \mathbf{t}_1 \\
& + \boldsymbol{\pi}_1 (s\mathbf{I} - \mathbf{T}_1)^{-1} \mathbf{t}_1 \pi_2^0 \\
= \quad & \pi_1^0 \pi_2^0 + \boldsymbol{\pi}_1 (s\mathbf{I} - \mathbf{T}_1)^{-1} \mathbf{t}_1 \pi_2^0 - \boldsymbol{\pi}_1 (s\mathbf{I} - \mathbf{T}_1)^{-1} \mathbf{t}_1 \boldsymbol{\pi}_2 (s\mathbf{I} - \mathbf{T}_2)^{-1} \mathbf{t}_2 \\
& + \boldsymbol{\pi}_2 (s\mathbf{I} - \mathbf{T}_2)^{-1} \mathbf{t}_2 \pi_1^0.
\end{aligned}
$$

By random variables having the same LST, we proof that commutativity holds.

For associativity. We consider

$$
\left(W_{1(\boldsymbol{\pi}_1, \mathbf{T}_1)} \sum W_{2(\boldsymbol{\pi}_2, \mathbf{T}_2)} \right) \sum W_{3(\boldsymbol{\pi}_3, \mathbf{T}_3)} = W'_{\left((\boldsymbol{\pi}_1, \pi_1^0 \boldsymbol{\pi}_2), \begin{bmatrix} \mathbf{T}_1 & \mathbf{t}_1 \boldsymbol{\pi}_2 \\ \mathbf{0} & \mathbf{T}_2 \end{bmatrix} \right)} \sum W_{3(\boldsymbol{\pi}_3, \mathbf{T}_3)}
$$

$$
= \bar{W}_{\left((\boldsymbol{\pi}_1, \pi_1^0 \boldsymbol{\pi}_2, \pi_1^0 \pi_2^0 \boldsymbol{\pi}_3), \begin{bmatrix} \mathbf{T}_1 & \mathbf{t}_1 \boldsymbol{\pi}_2 & \mathbf{t}_1 \pi_2^0 \boldsymbol{\pi}_3 \\ \mathbf{0} & \mathbf{T}_2 & \mathbf{t}_2 \boldsymbol{\pi}_3 \\ \mathbf{0} & \mathbf{0} & \mathbf{T}_3 \end{bmatrix} \right)},
$$

and

$$
W_{1(\boldsymbol{\pi}_1, \mathbf{T}_1)} \sum \left(W_{2(\boldsymbol{\pi}_2, \mathbf{T}_2)} \sum W_{3(\boldsymbol{\pi}_3, \mathbf{T}_3)} \right) = W_{1(\boldsymbol{\pi}_1, \mathbf{T}_1)} \sum W'_{\left((\boldsymbol{\pi}_2, \pi_2^0 \boldsymbol{\pi}_3), \begin{bmatrix} \mathbf{T}_2 & \mathbf{t}_2 \boldsymbol{\pi}_3 \\ \mathbf{0} & \mathbf{T}_3 \end{bmatrix} \right)}
$$

$$
= \bar{W}_{\left((\boldsymbol{\pi}_1, \pi_1^0 (\boldsymbol{\pi}_2, \pi_2^0 \boldsymbol{\pi}_3)), \begin{bmatrix} \mathbf{T}_1 & & \mathbf{t}_1 (\boldsymbol{\pi}_2, \pi_2^0 \boldsymbol{\pi}_3) \\ \mathbf{0} & \mathbf{T}_2 & \mathbf{t}_2 \boldsymbol{\pi}_3 \\ \mathbf{0} & \mathbf{0} & \mathbf{T}_3 \end{bmatrix} \right)}
$$

$$
= \bar{W}_{\left((\boldsymbol{\pi}_1, \pi_1^0 \boldsymbol{\pi}_2, \pi_1^0 \pi_2^0 \boldsymbol{\pi}_3), \begin{bmatrix} \mathbf{T}_1 & \mathbf{t}_1 \boldsymbol{\pi}_2 & \mathbf{t}_1 \pi_2^0 \boldsymbol{\pi}_3 \\ \mathbf{0} & \mathbf{T}_2 & \mathbf{t}_2 \boldsymbol{\pi}_3 \\ \mathbf{0} & \mathbf{0} & \mathbf{T}_3 \end{bmatrix} \right)}.
$$

By random variables having the same CPH representation, we proof that associativity holds.

For idempotence. The convolution of a CPH distribution with itself is equivalent to visit the same absorbing Markov chain twice with the same initial states, which is given as

$$
W_{(\boldsymbol{\pi}, \mathbf{T})} \sum W_{(\boldsymbol{\pi}, \mathbf{T})} \quad = \quad \bar{W}_{\left((\boldsymbol{\pi}, \pi^0 \boldsymbol{\pi}), \begin{bmatrix} \mathbf{T} & \mathbf{t}\boldsymbol{\pi} \\ \mathbf{0} & \mathbf{T} \end{bmatrix} \right)},
$$

such that the block matrix $\mathbf{t}\boldsymbol{\pi}$ contains the probability densities that the process exits the transient phases by $\mathbf{t}$ and immediately restarts with the same initial distribution $\boldsymbol{\pi}$. Thus idempotence does not hold.   □

THEOREM 3.6 *The finite minimum of continuous phase–type distributed random variables is commutative and associative, but not idempotent.*

PROOF For commutativity. We have

$$
W_{1(\boldsymbol{\pi}_1, \mathbf{T}_1)} \min W_{2(\boldsymbol{\pi}_2, \mathbf{T}_2)} = \bar{W}_{(\boldsymbol{\pi}_1 \otimes \boldsymbol{\pi}_2, \mathbf{T}_1 \oplus \mathbf{T}_2)},
$$

and
$$W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)} \min W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} = \bar{W}_{(\boldsymbol{\pi}_2\otimes\boldsymbol{\pi}_1,\mathbf{T}_2\oplus\mathbf{T}_1)},$$

have the same structure. By reordering the states, we are able to proof that the representations are indeed identical, thus commutativity holds.

For associativity. We have that the representations

$$
\begin{aligned}
(W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} \min W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)}) \min W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)} &= W'_{(\boldsymbol{\pi}_1\otimes\boldsymbol{\pi}_2,\mathbf{T}_1\oplus\mathbf{T}_2)} \min W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)} \\
&= \bar{W}_{(\boldsymbol{\pi}_1\otimes\boldsymbol{\pi}_2\otimes\boldsymbol{\pi}_3,\mathbf{T}_1\oplus\mathbf{T}_2\oplus\mathbf{T}_3)},
\end{aligned}
$$

and

$$
\begin{aligned}
W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} \min(W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)} \min W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}) &= W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} \min W'_{(\boldsymbol{\pi}_2\otimes\boldsymbol{\pi}_3,\mathbf{T}_2\oplus\mathbf{T}_3)} \\
&= \bar{W}_{(\boldsymbol{\pi}_1\otimes\boldsymbol{\pi}_2\otimes\boldsymbol{\pi}_3,\mathbf{T}_1\oplus\mathbf{T}_2\oplus\mathbf{T}_3)}
\end{aligned}
$$

are identical, thus associativity holds.

For idempotence. We show the property does not hold through a counterexample. Consider the EXP distributions with parameter λ, i.e. the CPH representation $(\boldsymbol{\pi}, \mathbf{T})$ is now $(1, [-\lambda])$, we get

$$W_{(1,[-\lambda])} \min W_{(1,[-\lambda])} = \bar{W}_{(1,[-2\lambda])},$$

which is clearly not same to $W_{(1,[-\lambda])}$. Thus, idempotent property doesn't hold.
□

THEOREM 3.7 *The finite maximum of continuous phase–type distributed random variables is commutative and associative, but not idempotent.*

PROOF For commutativity. We have

$$W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} \max W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)} = \bar{W}\left((\boldsymbol{\pi}_1\otimes\boldsymbol{\pi}_2,\boldsymbol{\pi}_1\pi_2^0,\pi_1^0\boldsymbol{\pi}_2), \begin{bmatrix} \mathbf{T}_1\oplus\mathbf{T}_2 & \mathbf{I}_{\mathbf{T}_1}\otimes\mathbf{t}_2 & \mathbf{t}_1\otimes\mathbf{I}_{\mathbf{T}_2} \\ 0 & \mathbf{T}_1 & 0 \\ 0 & 0 & \mathbf{T}_2 \end{bmatrix} \right),$$

and

$$W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)} \max W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} = \bar{W}\left((\boldsymbol{\pi}_2\otimes\boldsymbol{\pi}_1,\boldsymbol{\pi}_2\pi_1^0,\pi_1^0\boldsymbol{\pi}_1), \begin{bmatrix} \mathbf{T}_2\oplus\mathbf{T}_1 & \mathbf{I}_{\mathbf{T}_2}\otimes\mathbf{t}_1 & \mathbf{t}_2\otimes\mathbf{I}_{\mathbf{T}_1} \\ 0 & \mathbf{T}_2 & 0 \\ 0 & 0 & \mathbf{T}_1 \end{bmatrix} \right).$$

By reordering the states, we are able to prove that two random variables have the same CPH representation.

For associativity. Applying the same reordering strategy, we are able to proof associativity holds. We omit the expansion since the presentation will be very large for the maximum of three CPH distributed random variables.

For idempotence. We show idempotent property does not hold by considering EXP distribution with parameter λ, i.e. let $W_{(\boldsymbol{\pi},\mathbf{T})}$ be $W_{(1,[-\lambda])}$, as counterexample.

$$W_{(\boldsymbol{\pi},\mathbf{T})} \max W_{(\boldsymbol{\pi},\mathbf{T})} = \bar{W}_{\left((\boldsymbol{\pi}\otimes\boldsymbol{\pi},\boldsymbol{\pi}\pi^0,\pi^0\boldsymbol{\pi}), \begin{bmatrix} \mathbf{T}\oplus\mathbf{T} & \mathbf{I_T}\otimes\mathbf{t} & \mathbf{t}\otimes\mathbf{I_T} \\ 0 & \mathbf{T} & 0 \\ 0 & 0 & \mathbf{T} \end{bmatrix}\right)}$$

$$= \bar{W}_{\left((1,0,0), \begin{bmatrix} -2\lambda & \lambda & \lambda \\ 0 & -\lambda & 0 \\ 0 & 0 & -\lambda \end{bmatrix}\right)},$$

which is not equal to $W_{(1,[-\lambda])}$.   □

Convex mixture operator is a multinary operator such that for a finite integer k, and some probabilities $p_1, p_2, \ldots, p_k$ such that $\sum_{i=1}^{k} p_i = 1$ operated by Formula (3.9). However, considering the process algebra which comes later, we prefer binary operation. Therefore, we define the binary version of convex mixture operator for CPH distributions.

DEFINITION 3.8 (*Binary Convex Mixture*) Let p and q be some probabilities such that $p+q = 1$, the binary convex mixture of two continuous phase–type distributed random variables with representations CPH_{m_1} $(\boldsymbol{\pi}_1, \mathbf{T}_1)$ and CPH_{m_2} $(\boldsymbol{\pi}_2, \mathbf{T}_2)$ respectively, is itself continuous phase–type distributed with representation $\mathrm{CPH}_{m_1+m_2}(\boldsymbol{\pi}_3, \mathbf{L})$ given by $\boldsymbol{\pi}_3 = (p\boldsymbol{\pi}_1, q\boldsymbol{\pi}_2)$, and

$$\mathbf{L} = \begin{bmatrix} \mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 \end{bmatrix}. \tag{3.13}$$

Unless otherwise specified, the left side of binary operator $+$ is always assigned with probability p and the right side has the complement probability of q. Thus, it immediately yields that commutativity does not hold in general. Without loss of expressiveness, one could derive any finite convex mixture by successively applying binary convex mixture.

THEOREM 3.9 *The binary convex mixture of continuous phase–type distributed random variables is idempotent but not commutative and associative.*

PROOF For commutativity. Commutativity does not hold, because in general

$p \neq q$:

$$W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} + W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)} = \bar{W}_{\left((p\boldsymbol{\pi}_1,q\boldsymbol{\pi}_2),\begin{bmatrix}\mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2\end{bmatrix}\right)},$$

$$W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)} + W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} = \bar{W}_{\left((p\boldsymbol{\pi}_2,q\boldsymbol{\pi}_1),\begin{bmatrix}\mathbf{T}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_1\end{bmatrix}\right)} = \bar{W}_{\left((q\boldsymbol{\pi}_1,p\boldsymbol{\pi}_2),\begin{bmatrix}\mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2\end{bmatrix}\right)}.$$

For associativity. Let probabilities p_1 and q_1 for binary convex mixture between $\bar{W}_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)}$ and $W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)}$, and probabilities p_2 and q_2 for binary convex mixture between $W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)}$ and $W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}$, we have

$$\left(W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} + W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)}\right) + W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)} = W'_{\left((p_1\boldsymbol{\pi}_1,q_1\boldsymbol{\pi}_2),\begin{bmatrix}\mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2\end{bmatrix}\right)} + W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}$$

$$= \bar{W}_{\left((p_1 p_2\boldsymbol{\pi}_1,q_1 p_2\boldsymbol{\pi}_2,q_2\boldsymbol{\pi}_3),\begin{bmatrix}\mathbf{T}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{T}_3\end{bmatrix}\right)},$$

and

$$W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} + \left(W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)} + W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}\right) = W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} + W'_{\left((p_2\boldsymbol{\pi}_2,q_2\boldsymbol{\pi}_3),\begin{bmatrix}\mathbf{T}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_3\end{bmatrix}\right)}$$

$$= \bar{W}_{\left((p_1\boldsymbol{\pi}_1,q_1 p_2\boldsymbol{\pi}_2,q_1 q_2\boldsymbol{\pi}_3),\begin{bmatrix}\mathbf{T}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{T}_3\end{bmatrix}\right)}.$$

Generally, $p_1 p_2 \neq p_1$ and $q_2 \neq q_1 q_2$, thus associativity doesn't hold.

For Idempotence. Idempotence holds, because

$$W_{(\boldsymbol{\pi},\mathbf{T})} + W_{(\boldsymbol{\pi},\mathbf{T})} = \bar{W}_{\left((p\boldsymbol{\pi},q\boldsymbol{\pi}),\begin{bmatrix}\mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}\end{bmatrix}\right)},$$

which is clearly identical to $W_{(\boldsymbol{\pi},\mathbf{T})}$. $\qquad\qquad\square$

THEOREM 3.10 *The binary convex mixture of continuous phase–type distributed random variables is distributive over convolution, minimum and maximum.*

PROOF From Theorem 3.5, Theorem 3.6, and Theorem 3.7, convolution operator $\sum$, minimum operator min and maximum operator max are commutative, thus it is sufficient to show just right-distributivity.

For distributivity over convex mixture $+$, we get the following identical random variables:

$$\left(W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} + W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)}\right) \sum W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}$$

$$= W'_{\left((p\boldsymbol{\pi}_1,q\boldsymbol{\pi}_2),\begin{bmatrix} \mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 \end{bmatrix}\right)} \sum W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}$$

$$= \bar{W}_{\left((p\boldsymbol{\pi}_1,q\boldsymbol{\pi}_2,(p\pi_1^0+q\pi_2^0)\boldsymbol{\pi}_3),\begin{bmatrix} \mathbf{T}_1 & \mathbf{0} & \mathbf{t}_1\boldsymbol{\pi}_3 \\ \mathbf{0} & \mathbf{T}_2 & \mathbf{t}_2\boldsymbol{\pi}_3 \\ \mathbf{0} & \mathbf{0} & \mathbf{T}_3 \end{bmatrix}\right)},$$

and

$$\left(W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} \sum W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}\right) + \left(W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)} \sum W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}\right)$$

$$= W'_{\left((\boldsymbol{\pi}_1,\pi_1^0\boldsymbol{\pi}_3),\begin{bmatrix} \mathbf{T}_1 & \mathbf{t}_1\boldsymbol{\pi}_3 \\ \mathbf{0} & \mathbf{T}_3 \end{bmatrix}\right)} + W''_{\left((\boldsymbol{\pi}_2,\pi_2^0\boldsymbol{\pi}_3),\begin{bmatrix} \mathbf{T}_2 & \mathbf{t}_2\boldsymbol{\pi}_3 \\ \mathbf{0} & \mathbf{T}_3 \end{bmatrix}\right)}$$

$$= \bar{W}_{\left((p\boldsymbol{\pi}_1,p\pi_1^0\boldsymbol{\pi}_3,q\boldsymbol{\pi}_2,q\pi_2^0\boldsymbol{\pi}_3),\begin{bmatrix} \mathbf{T}_1 & \mathbf{t}_1\boldsymbol{\pi}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{T}_2 & \mathbf{t}_2\boldsymbol{\pi}_3 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{T}_3 \end{bmatrix}\right)}$$

$$= \bar{W}_{\left((p\boldsymbol{\pi}_1,q\boldsymbol{\pi}_2,(p\pi_1^0+q\pi_2^0)\boldsymbol{\pi}_3),\begin{bmatrix} \mathbf{T}_1 & \mathbf{0} & \mathbf{t}_1\boldsymbol{\pi}_3 \\ \mathbf{0} & \mathbf{T}_2 & \mathbf{t}_2\boldsymbol{\pi}_3 \\ \mathbf{0} & \mathbf{0} & \mathbf{T}_3 \end{bmatrix}\right)}.$$

For distributivity over minimum min, we get the following identical random variables:

$$\left(W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} + W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)}\right) \min W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}$$

$$= W'_{\left((p\boldsymbol{\pi}_1,q\boldsymbol{\pi}_2),\begin{bmatrix} \mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 \end{bmatrix}\right)} \min W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}$$

$$= \bar{W}_{\left((p\boldsymbol{\pi}_1\otimes\boldsymbol{\pi}_3,q\boldsymbol{\pi}_2\otimes\boldsymbol{\pi}_3),\begin{bmatrix} \mathbf{T}_1 \oplus \mathbf{T}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 \oplus \mathbf{T}_3 \end{bmatrix}\right)},$$

and

$$\left(W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} \min W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}\right) + \left(W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)} \min W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}\right)$$

$$= W'_{((\boldsymbol{\pi}_1\otimes\boldsymbol{\pi}_3),\mathbf{T}_1\oplus\mathbf{T}_3)} + W''_{((\boldsymbol{\pi}_2\otimes\boldsymbol{\pi}_3),\mathbf{T}_2\oplus\mathbf{T}_3)}$$

$$= \bar{W}_{\left((p\boldsymbol{\pi}_1\otimes\boldsymbol{\pi}_3,q\boldsymbol{\pi}_2\otimes\boldsymbol{\pi}_3),\begin{bmatrix} \mathbf{T}_1 \oplus \mathbf{T}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 \oplus \mathbf{T}_3 \end{bmatrix}\right)}.$$

For distributivity over maximum max, we get the following identical random variables:

$$\left(W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} + W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)}\right) \max W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}$$

$$= W'_{\left((p\boldsymbol{\pi}_1,q\boldsymbol{\pi}_2),\begin{bmatrix}\mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2\end{bmatrix}\right)} \max W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}$$

$$= \bar{W}_{((p\boldsymbol{\pi}_1\otimes\boldsymbol{\pi}_3,q\boldsymbol{\pi}_2\otimes\boldsymbol{\pi}_3,p\boldsymbol{\pi}_1\pi_3^0,q\boldsymbol{\pi}_2\pi_3^0,p\pi_1^0\boldsymbol{\pi}_3,q\pi_2^0\boldsymbol{\pi}_3),}$$

$$\begin{bmatrix}
\mathbf{T}_1 \oplus \mathbf{T}_3 & \mathbf{0} & \mathbf{I}_{\mathbf{T}_1}\otimes\mathbf{t}_3 & \mathbf{0} & \mathbf{t}_1\otimes\mathbf{I}_{\mathbf{T}_3} & \mathbf{0} \\
\mathbf{0} & \mathbf{T}_2 \oplus \mathbf{T}_3 & \mathbf{0} & \mathbf{I}_{\mathbf{T}_2}\otimes\mathbf{t}_3 & \mathbf{0} & \mathbf{t}_2\otimes\mathbf{I}_{\mathbf{T}_3} \\
\mathbf{0} & \mathbf{0} & \mathbf{T}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{T}_2 & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{T}_3 & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{T}_3
\end{bmatrix}),$$

and

$$W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} \max W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)} + W_{2(\boldsymbol{\pi}_2,\mathbf{T}_2)} \max W_{3(\boldsymbol{\pi}_3,\mathbf{T}_3)}$$

$$= W'_{\left((\boldsymbol{\pi}_1\otimes\boldsymbol{\pi}_3,\boldsymbol{\pi}_1\pi_3^0,\pi_1^0\boldsymbol{\pi}_3),\begin{bmatrix}\mathbf{T}_1 \oplus \mathbf{T}_3 & \mathbf{I}_{\mathbf{T}_1}\otimes\mathbf{t}_3 & \mathbf{t}_1\otimes\mathbf{I}_{\mathbf{T}_3} \\ \mathbf{0} & \mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{T}_3\end{bmatrix}\right)} +$$

$$W''_{\left((\boldsymbol{\pi}_2\otimes\boldsymbol{\pi}_3,\boldsymbol{\pi}_2\pi_3^0,\pi_2^0\boldsymbol{\pi}_3),\begin{bmatrix}\mathbf{T}_2 \oplus \mathbf{T}_3 & \mathbf{I}_{\mathbf{T}_2}\otimes\mathbf{t}_3 & \mathbf{t}_2\otimes\mathbf{I}_{\mathbf{T}_3} \\ \mathbf{0} & \mathbf{T}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{T}_3\end{bmatrix}\right)}$$

$$= \bar{W}_{((p\boldsymbol{\pi}_1\otimes\boldsymbol{\pi}_3,p\boldsymbol{\pi}_1\pi_3^0,p\pi_1^0\boldsymbol{\pi}_3,q\boldsymbol{\pi}_2\otimes\boldsymbol{\pi}_3,q\boldsymbol{\pi}_2\pi_3^0,q\pi_2^0\boldsymbol{\pi}_3),}$$

$$\begin{bmatrix}
\mathbf{T}_1 \oplus \mathbf{T}_3 & \mathbf{I}_{\mathbf{T}_1}\otimes\mathbf{t}_3 & \mathbf{t}_1\otimes\mathbf{I}_{\mathbf{T}_3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{T}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{T}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{T}_2 \oplus T_3 & \mathbf{I}_{\mathbf{T}_2}\otimes\mathbf{t}_3 & \mathbf{t}_2\otimes\mathbf{I}_{\mathbf{T}_3} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{T}_2 & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{T}_3
\end{bmatrix}),$$

such that both representations are indeed the same by reordering the states. $\square$

THEOREM 3.11 *There is no other distributive property among convolution, minimum and maximum operators for continuous phase–type distributions.*

PROOF This theorem shows three sub-results: the convolution of continuous phase–type distributed random variables is not distributive over minimum and maximum; the minimum of continuous phase–type distributed random variables is not distributive over convolution and maximum; the maximum of continuous

phase–type distributed random variables is not distributive over convolution and minimum. We prove them using EXP distributions as counterexamples, such that the complete proof is given in Appendix A.3. □

3.2.4.2 Empty PH distributions

A CPH distribution is the distribution of the time until absorption in an absorbing CTMC. By Definition 2.8 in Section 2.2.2, infinite absorption time is strictly excluded from CPH distributions (occurs with probability 0), because infinite absorption time is equivalent to that the CTMC will never reach absorption, which contradicts the definition. On the other hand, we can consider absorption times of zero, where the distribution is known as *empty phase–type distribution* [O'C90]. An empty CPH distribution has a representation $(\mathbf{0}, \star)$, such that $\star$ is an irrelevant matrix. Empty PH distributions have special properties when they operate with other CPH distributions.

Neutral element and absorbing element In mathematics, a *neutral element (or identity element)* is a special element in a set, which leaves other elements unchanged when operated with them. An *annihilating element (or absorbing element)* is a special element such that operating an annihilating element with any element of the set is the annihilating element itself. For example, let $\Re$ be the set, 0 is a neutral element under addition, and an annihilating element under multiplication. This example also shows that, even with the same set, the same element could be a different kind of element depending on the type of operation under consideration.

PROPOSITION 3.12 *The empty phase–type distributions are neutral elements of the convolution operator and maximum operator for the class of continuous phase–type distributions.*

PROOF For convolution: Theorem 3.5 shows the convolution operator is commutative, thus we just show one direction:

$$W_{(\boldsymbol{\pi},\mathbf{T})} \sum W^*_{(\mathbf{0},\star)} \quad = \quad \bar{W}_{\left((\boldsymbol{\pi},\mathbf{0}),\begin{bmatrix}\mathbf{T} & \mathbf{0} \\ \mathbf{0} & \star\end{bmatrix}\right)}.$$

Since the matrix block $\star$ will never be reached, the random variable $\bar{W}$ is equivalent to $W_{(\boldsymbol{\pi},\mathbf{T})}$.

For maximum: Theorem 3.7 shows the maximum operator is commutative, thus we just show one direction. Let $\hat{\star} = \mathbf{e} - \star\mathbf{e}$, we have

$$W_{(\boldsymbol{\pi},\mathbf{T})} \max W^*_{(\mathbf{0},\star)} = \bar{W}_{\left((\boldsymbol{\pi}\otimes\mathbf{0},\boldsymbol{\pi},\mathbf{0}), \begin{bmatrix} \mathbf{T}\oplus\star & \mathbf{I}\otimes\hat{\star} & \mathbf{t}\otimes\mathbf{I} \\ \mathbf{0} & \mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \star \end{bmatrix}\right)}$$

$$= \bar{W}_{\left((\mathbf{0},\boldsymbol{\pi},\mathbf{0}), \begin{bmatrix} \mathbf{T}\oplus\star & \mathbf{I}\otimes\hat{\star} & \mathbf{t}\otimes\mathbf{I} \\ \mathbf{0} & \mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \star \end{bmatrix}\right)}.$$

Since all matrix blocks will never be reached except $\mathbf{T}$, the random variable $\bar{W}$ is equivalent to $W_{(\boldsymbol{\pi},\mathbf{T})}$. $\qquad\square$

PROPOSITION 3.13 *The empty PH distributions are absorbing elements of the minimum operator for the class of continuous phase–type distributions.*

PROOF Theorem 3.6 shows the minimum operator is commutative, thus we just show

$$W_{(\boldsymbol{\pi},\mathbf{T})} \min W^*_{(\mathbf{0},\star)} = W^*_{\left((\boldsymbol{\pi}\otimes\mathbf{0}), [\mathbf{T}\oplus\star]\right)},$$

which is again an empty phase–type distribution. $\qquad\square$

Now, we conclude our findings on the compositionality of continuous phase–type distributions in Table 3.2.

	Composite Operators			
Properties	$\sum$	min	max	+
Commutativity	✓	✓	✓	
Associativity	✓	✓	✓	
Idempotence				✓
Neutral Element	✓		✓	
Absorbing Element		✓		
Distributive Over	+	+	+	

• $\sum$– Convolution • min– Minimum • max– Maximum • +– Convex Mixture

Table 3.2: Algebraic properties for the composite operators of continuous phase–type distributions.

3.2.5   Phase–type Process Algebra

Process algebras are high-level specification languages to formally specify concurrent systems, which provide a mechanism to capture the important aspects

and behaviours of the systems. In this part, we introduce a process algebra for continuous phase–type distributions, namely *Phase–type Process Algebra (*PHPA*)*, to model concurrent stochastic systems with CPH distributed process durations.

Let a *random delay* of the PHPA be a CPH distributed random variable, which can be formalised as an absorbing CTMC. Recall that CPH distributions can be versatilely used to approximate any distributions on $[0, \infty)$ [JT88]. Having CPH distributed random delays as primitives, we could straightforwardly encode the composite operators with their algebraic properties for CPH representations (cf. Section 3.2.4) into the language. In order to express recursive behaviour, we also define a finite set $\mathcal{V}$ of process variables. Now, we are ready to introduce the formal syntax of the PHPA.

DEFINITION 3.14 (PHPA ***Syntax***) Let $W_{(\boldsymbol{\pi}, \mathbf{T})} \in \Re_{\geq 0}$ be a continuous phase–type distributed random variable represented by $(\boldsymbol{\pi}, \mathbf{T})$, and let $X \in \mathcal{V}$ be a process variable, the language PHPA is defined using phase–type processes P by the following grammar.

$$P ::= 0 \mid W_{(\boldsymbol{\pi}, \mathbf{T})}.P \mid P : P \mid P + P \mid P \nabla P \mid P \triangle P \mid X \mid [X := P]_i$$

$[X := P]$ is a shorthand notation for an arbitrary (finite) set of definitions of the form $[X_1 := P_1, X_2 := P_2, \ldots, X_n := P_n]$.

Process 0 simply expresses a termination.

Process $W_{(\boldsymbol{\pi}, \mathbf{T})}.P$ expresses a *phase–type delay prefix*, which means that the process has to delay for a CPH distributed random time W according to the representation $(\boldsymbol{\pi}, \mathbf{T})$, and then turns to the process P.

Let P, $\mathcal{Q}$ and $\mathcal{R}$ range over phase–type processes, the language PHPA includes four fundamental composite scenarios.

- Process $P : \mathcal{Q}$ stands for *sequential composition*, such that the process $\mathcal{Q}$ behaves after the process P finishes. The sequential composition is operated as the sum of the CPH distributed delays from P and $\mathcal{Q}$, which follows Rule (3.6) and holds the algebraic properties in Theorem 3.5.

- Process $P + \mathcal{Q}$ stands for *probabilistic composition*, such that one of the processes P and $\mathcal{Q}$ is chosen probabilistically to execute. The probabilistic composition is operated as the binary convex mixture of the CPH distributed delays from P and $\mathcal{Q}$, which follows Rule (3.13) and holds the

algebraic properties in Theorem 3.9. Recall that Rule (3.13) ensures the process P is assigned with probability p and the process Q is assigned with probability q, such that $p + q = 1$.

- Process $P \nabla Q$ stands for *synchronic composition*, such that two processes P and Q synchronise with each other, that the one finished later corresponds to the continuation. The synchronic composition is operated as the maximum of the CPH distributed delays from P and Q, which follows Rule (3.8) and holds the algebraic properties in Theorem 3.7.

- Process $P \triangle Q$ stands for *racing composition*, such that two processes P and Q compete with each other, that the one finished earlier corresponds to the continuation. The racing composition is operated as the minimum of the CPH distributed delays from P and Q, which follows Rule (3.7) and holds the algebraic properties in Theorem 3.6.

The expression $[X := P]_i$ defines the behaviour of the i-th process variable of the mutually recursive behaviour definition set $[X := P]$. It means $[X := P]_i$ behaves like P_i when the recursive variable X_i is reached from somewhere else in the definition set.

The formal semantics of PHPA are mapped on phase–type labelled transition systems, which is defined as:

DEFINITION 3.15 (*PH Labelled Transition System*) A Phase–type Labelled Transition System is given by the triple ($\mathcal{C}$, $\mathcal{C}_0$, $\mathbf{W}_{(\boldsymbol{\pi},\mathbf{T})}$, $\rightarrow$), where

- $\mathcal{C}$ is the set of all PHPA processes,

- $\mathcal{C}_0$ is the initial process,

- $\mathbf{W}_{(\boldsymbol{\pi},\mathbf{T})}$ is the set of continuous phase–type distributed random variables, such that a random variable $W_{1(\boldsymbol{\pi}_1,\mathbf{T}_1)} \in \mathbf{W}_{(\boldsymbol{\pi},\mathbf{T})}$ is distributed according to the representation $(\boldsymbol{\pi}_1, \mathbf{T}_1)$,

- $\rightarrow$ is the *PH transition relation* such that $\rightarrow \subset \mathcal{C} \times W_{(\boldsymbol{\pi},\mathbf{T})} \times \mathcal{C}$.

Now, we give the formal semantics of PHPA by *Structural Operational Semantics (SOS)* [Plo81]. In general, a SOS derivation rule is of the form

$$\frac{Premises}{Conclusions} \quad Conditions$$

to express that *if Conditions hold, then Premises imply Conclusions.* The SOS Derivation Rules for PHPA are given in Table 3.3, where the probabilities of

Table 3.3: The SOS derivation rules for PHPA.

$$[\text{PREFIX}] \quad \frac{P := W_{(\boldsymbol{\pi},\mathbf{T})}.P'}{P \stackrel{W_{(\boldsymbol{\pi},\mathbf{T})}}{\rightarrow} P'}$$

$$[\text{PROB1}] \quad \frac{P \stackrel{W_1(\boldsymbol{\pi}_1,\mathbf{T}_1)}{\rightarrow} P', \quad Q \stackrel{W_2(\boldsymbol{\pi}_2,\mathbf{T}_2)}{\rightarrow} Q'}{P + Q \stackrel{W_{(\boldsymbol{\pi}_1,\mathbf{T}_1)+(\boldsymbol{\pi}_2,\mathbf{T}_2)}}{\rightarrow} P'}$$

$$[\text{PROB2}] \quad \frac{P \stackrel{W_1(\boldsymbol{\pi}_1,\mathbf{T}_1)}{\rightarrow} P', \quad Q \stackrel{W_2(\boldsymbol{\pi}_2,\mathbf{T}_2)}{\rightarrow} Q'}{P + Q \stackrel{W_{(\boldsymbol{\pi}_1,\mathbf{T}_1)+(\boldsymbol{\pi}_2,\mathbf{T}_2)}}{\rightarrow} Q'}$$

$$[\text{CONV}] \quad \frac{P \stackrel{W_1(\boldsymbol{\pi}_1,\mathbf{T}_1)}{\rightarrow} 0, \quad Q \stackrel{W_2(\boldsymbol{\pi}_2,\mathbf{T}_2)}{\rightarrow} Q'}{P : Q \stackrel{W_{(\boldsymbol{\pi}_1,\mathbf{T}_1)\sum(\boldsymbol{\pi}_2,\mathbf{T}_2)}}{\rightarrow} Q'}$$

$$[\text{MAX1}] \quad \frac{P \stackrel{W_1(\boldsymbol{\pi}_1,\mathbf{T}_1)}{\rightarrow} P', \quad Q \stackrel{W_2(\boldsymbol{\pi}_2,\mathbf{T}_2)}{\rightarrow} Q'}{P \nabla Q \stackrel{W_{(\boldsymbol{\pi}_1,\mathbf{T}_1)\max(\boldsymbol{\pi}_2,\mathbf{T}_2)}}{\rightarrow} P'} \quad W_1 > W_2$$

$$[\text{MAX2}] \quad \frac{P \stackrel{W_1(\boldsymbol{\pi}_1,\mathbf{T}_1)}{\rightarrow} P', \quad Q \stackrel{W_2(\boldsymbol{\pi}_2,\mathbf{T}_2)}{\rightarrow} Q'}{P \nabla Q \stackrel{W_{(\boldsymbol{\pi}_1,\mathbf{T}_1)\max(\boldsymbol{\pi}_2,\mathbf{T}_2)}}{\rightarrow} Q'} \quad W_1 < W_2$$

$$[\text{MIN1}] \quad \frac{P \stackrel{W_1(\boldsymbol{\pi}_1,\mathbf{T}_1)}{\rightarrow} P', \quad Q \stackrel{W_2(\boldsymbol{\pi}_2,\mathbf{T}_2)}{\rightarrow} Q'}{P \triangle Q \stackrel{W_{(\boldsymbol{\pi}_1,\mathbf{T}_1)\min(\boldsymbol{\pi}_2,\mathbf{T}_2)}}{\rightarrow} P'} \quad W_1 < W_2$$

$$[\text{MIN2}] \quad \frac{P \stackrel{W_1(\boldsymbol{\pi}_1,\mathbf{T}_1)}{\rightarrow} P', \quad Q \stackrel{W_2(\boldsymbol{\pi}_2,\mathbf{T}_2)}{\rightarrow} Q'}{P \triangle Q \stackrel{W_{(\boldsymbol{\pi}_1,\mathbf{T}_1)\min(\boldsymbol{\pi}_2,\mathbf{T}_2)}}{\rightarrow} Q'} \quad W_1 > W_2$$

$$[\text{REC}] \quad \frac{P_i \stackrel{W_{(\boldsymbol{\pi},\mathbf{T})}}{\rightarrow} P'_i}{[X := P]_i \stackrel{W_{(\boldsymbol{\pi},\mathbf{T})}}{\rightarrow} P'_i}$$

the conditions, i.e. $W_1 < W_2$ and $W_1 > W_2$, can be calculated according to Formula 3.12 in Section 3.2.3.

Now, let $P \neq Q \neq R$, according to the algebraic properties for the composite operators of CPH distributions (cf. Table 3.2), the PHPA has the following algebraic operational rules.

$$P : Q = Q : P \quad (P : Q) : R = Q : (P : R) \quad (\mathbf{0}, \star) : P = P$$
$$P + P = P$$
$$P \nabla Q = Q \nabla P \quad (P \nabla Q) \nabla R = Q \nabla (P \nabla R) \quad (\mathbf{0}, \star) \nabla P = P$$
$$P \triangle Q = Q \triangle P \quad (P \triangle Q) \triangle R = Q \triangle (P \triangle R) \quad (\mathbf{0}, \star) \triangle P = (\mathbf{0}, \star)$$
$$P : (Q + R) = (P : Q) + (P : R)$$
$$P \nabla (Q + R) = (P \nabla Q) + (P \nabla R)$$
$$P \triangle (Q + R) = (P \triangle Q) + (P \triangle R)$$

Table 3.4: PHPA algebraic rules.

3.2.6 Summary

Process algebra emerged as a formal modelling technique for the qualitative analysis of concurrent systems for many years. Over the last 20 years, there have been many attempts to take process algebras to model the field of performance evaluation. A process algebra allows one to model systems consisting of interacting components to be constructed systematically. In this work, we take the advantage of the attractive features of stochastic process algebras to offer the compositionality of continuous phase–type distributions in the stochastic system modelling.

Phase–type process algebra is a stochastic process algebra having CPH distributed random delays, such that one could model large-scale and complicated system using CPH distributions as building blocks. From sound mathematical operations on CPH distributions together with their proved algebraic properties, one could then express the systems using our language with fully respect to the underlying stochastic processes. It is important that the stochastic systems obtained from our language can always be formulated as a huge absorbing Markov chain, which can be analysed using standard techniques for evaluating continuous–time Markov chains.

CHAPTER 4

Stochastic Equivalence and Time-lapse Bisimulation

A *discrete phase–type distribution* is the distribution of the time until absorption of a Markov chain with a finite number of transient states and one absorbing state. A *discrete–time marked Markovian arrival process* is a marked point process with phase–type distributed intervals. In this chapter, we give the definitions of *stochastic equivalence* for discrete phase–type distributions and discrete–time marked Markovian arrival processes. Derived from stochastic equivalence, we define a new bisimulation relation, named *time-lapse bisimulation*, for labelled discrete–time Markov chains to capture probabilistic timed behaviour. We show that time-lapse bisimulation counts the number of internal actions on traces, which is coarser than strong bisimulation and not comparable with weak bisimulation. A Hennessy-Milner style logic for characterising time-lapse bisimulation is presented. Last but not least, we show that bisimulation characterisation of stochastic equivalence are sound for some bisimulations, but none of the bisimulations are complete to stochastic equivalence.

4.1 Introduction

There are many approaches to define relations intending to capture when two processes have the same behaviour. In the case of non-stochastic process algebras there exist whole families of trace-based equivalences, bisimulations, and other forms of equivalence. One point of view is that processes have the same behaviour if there is no way that a series of experiments upon the two processes can demonstrate their difference. In the case of non-stochastic process algebras this has given rise to notions of testing equivalence [NH83]. Another point of view is to give a co-inductive characterisation of when processes can mimic each other. This has given rise to a variety of bisimulation relations. Comparing the two notions, the bisimulation characterisation is sound in that two bisimilar processes are also testing equivalent, whereas we cannot expect completeness, that two processes that are testing equivalent can necessarily be shown to be bisimilar. This apparent weakness of bisimulations is compensated by the many algorithmic techniques that allow demonstrating whether or not two processes are bisimilar.

Turning our attention to stochastic systems there are notions of stochastic equivalence [LN91] as well as notions of probabilistic bisimulation [JS90, LS91]. Carrying over from labelled transition systems, probabilistic versions of strong and weak bisimulations characterise the probabilistic behaviour of the stochastic systems. In this work, we integrate the notion of stochastic equivalence from stochastic processes into probabilistic transition systems to define a new type of bisimulation relation. First, we introduce the notion of stochastic equivalence for two classes of stochastic systems, those of discrete phase–type distributions and discrete–time marked Markovian arrival processes. Discrete phase–type distributions [Neu75] are distributions of the time until absorption of a Markov chain. A discrete–time Markovian arrival processes [Neu79] is a point process with DPH distributed sojourn times between arrivals. These processes are further extended to allow marked transitions, thus defining discrete–time Marked Markovian Arrival Processes (MMAPs) [HN98, BN10].

To express MMAPs with probabilistic transition systems, we introduce labelled discrete–time Markov chains, which are also called *fully probabilistic transition systems* [vGSST90, BH97]. Having introduced stochastic equivalence, we propose a new bisimulation relation for labelled DTMCs, named time-lapse bisimulation, such that two time-lapse bisimilar states shall have the same probabilistic behaviour over time. The contributions of this work are

- to give notions of stochastic equivalence to discrete stochastic systems,

- to define time-lapse bisimulation for labelled DTMCs,

- to clarify the relationships between time-lapse bisimulation, strong bisimulation and weak bisimulation,

- to characterise time-lapse bisimulation by a Hennessy-Milner style logic,

- to discuss bisimulation characterisation of stochastic equivalence.

The structure of the chapter is as follows. We introduce the notion of stochastic equivalence and some decidability results in Section 4.2. In Section 4.3, we develop the definition of time-lapse bisimulation for labelled DTMCs and illustrate the relationship with strong and weak bisimulations. Section 4.4 clarifies bisimulation characterisation of stochastic equivalence covering time-lapse bisimulation, strong bisimulation and weak bisimulation. We conclude in Section 4.5.

4.2   Stochastic Equivalence

Stochastic equivalence is a fundamental principle of probability distributions and stochastic processes, see e.g. [LN91, Ros96]. Algebraic equivalence as introduced in [BT11b] is identical to stochastic equivalence. In this section, we introduce the notion of stochastic equivalence for discrete phase–type distributions (cf. Section 2.1.2, Definition 2.2) and discrete–time marked Markovian arrival processes (cf. Section 2.1.4, Definition 2.5).

4.2.1   Stochastic Equivalence for Phase–type Distributions

Stochastic equivalence for general probability distributions is defined as:

DEFINITION 4.1 (*Stochastic Equivalence for Probability Distributions*)
Two discrete random variables X and Y are stochastically equivalent if their cumulative distribution functions $F_X(x) = \mathbb{P}(X \leq x)$ and $F_Y(y) = \mathbb{P}(Y \leq y)$ are identical, i.e. $\mathbb{P}(X \leq z) = \mathbb{P}(Y \leq z)$ for all z.

Stochastic equivalence for two DPH distributed random variables with representations $\mathrm{DPH}(\boldsymbol{\alpha}, \mathbf{T})$ and $\mathrm{DPH}(\boldsymbol{\beta}, \mathbf{S})$ respectively implies that for all $k \geq 0$ we must have $\boldsymbol{\alpha}\mathbf{T}^k\mathbf{e} = \boldsymbol{\beta}\mathbf{S}^k\mathbf{e}$. It is known from [O'C90] that a DPH distribution is characterised by its first $2p - 1$ factorial moments (cf. Section 2.1.2, Formula (2.4)). Using the Cayley-Hamilton Theorem [AM69], the equality of the first $2p - 1$ values of the cumulative distribution function ensures the equality for all

$k \geq 0$. We show the decidability of stochastic equivalence for DPH distributions in Proposition 4.2.

PROPOSITION 4.2 *Given two discrete phase–type representations DPH($\boldsymbol{\alpha}$, $\mathbf{T}$) and DPH($\boldsymbol{\beta}$, $\mathbf{S}$) of order n and m respectively, they are stochastically equivalent iff $\boldsymbol{\alpha}\mathbf{T}^k\mathbf{e} = \boldsymbol{\beta}\mathbf{S}^k\mathbf{e}$ for $0 \leq k \leq 2\max\{n, m\}$-1.*

PROOF To prove the proposition, we show that for a representation DPH($\boldsymbol{\alpha}$, $\mathbf{T}$) of order n, the $\{\boldsymbol{\alpha}\mathbf{T}^k\mathbf{e}\}_{0\leq k\leq 2n-1}$ completely characterises the distribution.

The Cayley-Hamilton Theorem states every square matrix over a real field satisfies its own characteristic equation. Let $\mathbf{A}$ be a square matrix of dimension p, we have that $\mathbf{A}^p = a_0\mathbf{I} + a_1\mathbf{A} + \cdots + a_{p-1}\mathbf{A}^{p-1}$. That is $\{\mathbf{A}^k\}_{k\geq p}$ can be decided by the linear combination of the lower power terms $\mathbf{I}$, $\mathbf{A}$, ..., $\mathbf{A}^{p-1}$ with the p coefficients, $a_0, a_1, \ldots, a_{p-1}$. Using the Cayley-Hamilton Theorem for the DPH($\boldsymbol{\alpha}$, $\mathbf{T}$), we have $\forall k \geq n$: $\boldsymbol{\alpha}\mathbf{T}^k\mathbf{e} = a_0\boldsymbol{\alpha}\mathbf{T}^0\mathbf{e} + a_1\boldsymbol{\alpha}\mathbf{T}\mathbf{e} + \cdots + a_{n-1}\boldsymbol{\alpha}\mathbf{T}^{n-1}\mathbf{e}$. To determine $\forall k \geq 0$: $\boldsymbol{\alpha}\mathbf{T}^k\mathbf{e}$, we require first n basic scalars, i.e. $\boldsymbol{\alpha}\mathbf{T}^0\mathbf{e}$, ..., $\boldsymbol{\alpha}\mathbf{T}^{n-1}\mathbf{e}$, and then the successive n scalars, i.e. $\boldsymbol{\alpha}\mathbf{T}^n\mathbf{e}$, ..., $\boldsymbol{\alpha}\mathbf{T}^{2n-1}\mathbf{e}$, shall determine the n coefficients, i.e. $a_0, a_1, \ldots, a_{n-1}$, because

$$\boldsymbol{\alpha}\mathbf{T}^n\mathbf{e} = a_0\boldsymbol{\alpha}\mathbf{T}^0\mathbf{e} + a_1\boldsymbol{\alpha}\mathbf{T}\mathbf{e} + \cdots + a_{n-1}\boldsymbol{\alpha}\mathbf{T}^{n-1}\mathbf{e}$$

$$\boldsymbol{\alpha}\mathbf{T}^{n+1}\mathbf{e} = a_0\boldsymbol{\alpha}\mathbf{T}\mathbf{e} + a_1\boldsymbol{\alpha}\mathbf{T}^2\mathbf{e} + \cdots + a_{n-1}\boldsymbol{\alpha}\mathbf{T}^n\mathbf{e}$$

$$= a_0\boldsymbol{\alpha}\mathbf{T}\mathbf{e} + a_1\boldsymbol{\alpha}\mathbf{T}^2\mathbf{e} + \cdots + a_{n-1}(a_0\boldsymbol{\alpha}\mathbf{T}^0\mathbf{e} + a_1\boldsymbol{\alpha}\mathbf{T}\mathbf{e} + \cdots + a_{n-1}\boldsymbol{\alpha}\mathbf{T}^{n-1}\mathbf{e})$$

$$\vdots$$

$$\boldsymbol{\alpha}\mathbf{T}^{2n-1}\mathbf{e} = \ldots$$

is a system of linear equations with n equations and n unknowns. Therefore, the first $2n$ power terms shall completely characterise the DPH distribution. $\square$

4.2.2 Stochastic Equivalence for Marked Markovian Arrival Processes

For a stochastic process, stochastic equivalence is defined through the finite dimensional distributions [LN91]. For discrete–time MMAPs, we consider a sequence of the pairs of random variables such that the first random variable gives the interarrival time and the second random variable gives the type of the arrival mark.

DEFINITION 4.3 (*Stochastic Equivalence for MMAPs*) Let Q_n and R_n be the random variables of the interarrival times, Y_n and Z_n be the random

variables of the arrival marks of two marked Markovian arrival processes. The two processes $\{(Q_n, Y_n)\}_{n \in \mathbb{N}}$ and $\{(R_n, Z_n)\}_{n \in \mathbb{N}}$ are stochastically equivalent if for all $n \in \mathbb{N}$ and $m_n, l_n \in \mathbb{N}_0$: $\mathbb{P}((Q_1 = m_1, Y_1 = l_1), \ldots, (Q_n = m_n, Y_n = l_n)) = \mathbb{P}((R_1 = m_1, Z_1 = l_1), \ldots, (R_n = m_n, Z_n = l_n))$.

By definition, we have that two MMAPs are stochastically equivalent iff all their corresponding joint density functions (cf. Section 2.1.4, Formula (2.8)) are identical. The characterisation of continuous–time MMAPs through moments is studied in [BKK10, Tel11], we adapt their results to the discrete–time setting, with extra considerations of arbitrary initial distributions.

DEFINITION 4.4 (***Non-redundant MMAP representation***) A discrete–time MMAP representation is called *non-redundant* if the order of representation equals the maximum degree of the numerator and the denominator of $f(s, \boldsymbol{z})$ (cf. Section 2.1.4, Formula (2.7)), and *redundant* otherwise.

A non-redundant discrete–time MMAP representation has independent joint moments. Therefore, one could decide stochastic equivalence through joint moments.

THEOREM 4.5 *Consider two non-redundant discrete–time MMAP representations with k marks of order n and m, and let $p = \max\{n, m\}$. They are stochastically equivalent iff the first $kp^2 + p$ joint factorial moments of successive interarrival times agree.*

PROOF To prove the theorem, we show that a discrete–time MMAP with representation $(\boldsymbol{\pi}, \mathbf{D}_0, \mathbf{D}_{\boldsymbol{h}_0}, \ldots, \mathbf{D}_{\boldsymbol{h}_{k-1}})$ of order n and k marks is completely characterised by $kp^2 + p$ joint factorial moments. Our proof follows the idea from Theorem 2 in [BKK10] and Theorem 4 in [TH07].

Recall that the joint factorial moments of the interracial times and arrival marks (cf. Section 2.1.4, Formula (2.9)) are given as

$$\mathbb{E}\left(\prod_{i=0}^m \Theta_i \ldots (\Theta_i - (t_i - 1))\mathbf{1}(\boldsymbol{H}_i = \boldsymbol{h}_i)\right) = \boldsymbol{\pi}\left(\prod_{i=0}^m t_i!(\mathbf{I} - \mathbf{D}_0)^{-(t_i+1)}\mathbf{D}_0^{t_i-1}\mathbf{D}_{\boldsymbol{h}_i}\right)\mathbf{e}.$$

Consider a base case that there exists only one arrival mark $\boldsymbol{h}$, let $\mathbf{P} = (\mathbf{I} - \mathbf{D}_0)^{-1}\mathbf{D}_{\boldsymbol{h}}$ be the transition matrix of embedded Markov chain and let the Jordan decomposition [Str88] $\mathbf{D}_0 = \Gamma^{-1}\mathbf{E}\Gamma$, such that the $\mathbf{E}$ is a diagonal matrix and the normalised matrix Γ satisfies $\Gamma\mathbf{e} = \mathbf{e}$. We rewrite the joint factorial moments

as

$$\mathbb{E}\left(\prod_{i=0}^{m}\Theta_i\ldots(\Theta_i-(t_i-1))\right)$$

$$= \boldsymbol{\pi}t_0!(\mathbf{I}-\mathbf{D}_0)^{-(t_0+1)}\mathbf{D}_0^{t_0-1}\mathbf{D}_{\boldsymbol{h}}\cdots t_m!(\mathbf{I}-\mathbf{D}_0)^{-(t_m+1)}\mathbf{D}_0^{t_m-1}\mathbf{D}_{\boldsymbol{h}}\mathbf{e}$$

$$= \boldsymbol{\pi}t_0!(\mathbf{I}-\mathbf{D}_0)^{-t_0}\mathbf{D}_0^{t_0-1}\underbrace{(\mathbf{I}-\mathbf{D}_0)^{-1}\mathbf{D}_{\boldsymbol{h}}}\cdots t_m!(\mathbf{I}-\mathbf{D}_0)^{-t_m}\mathbf{D}_0^{t_m-1}\underbrace{(\mathbf{I}-\mathbf{D}_0)^{-1}\mathbf{D}_{\boldsymbol{h}}}\mathbf{e}$$

$$= \boldsymbol{\pi}t_0!(\Gamma^{-1}\Gamma-\Gamma^{-1}\mathbf{E}\Gamma)^{-t_0}(\Gamma^{-1}\mathbf{E}\Gamma)^{t_0-1}\mathbf{P}\cdots t_m!(\Gamma^{-1}\Gamma-\Gamma^{-1}\mathbf{E}\Gamma)^{-t_m}(\Gamma^{-1}\mathbf{E}\Gamma)^{t_m-1}$$
$$\mathbf{P}\mathbf{e}$$

$$= \boldsymbol{\pi}t_0!(\Gamma^{-1}(\mathbf{I}-\mathbf{E})\Gamma)^{-t_0}\Gamma^{-1}\mathbf{E}^{t_0-1}\Gamma\mathbf{P}\cdots t_m!(\Gamma^{-1}(\mathbf{I}-\mathbf{E})\Gamma)^{-t_m}\Gamma^{-1}\mathbf{E}^{t_0-1}\Gamma\mathbf{P}\Gamma^{-1}\Gamma\mathbf{e}$$

$$= t_0!\underbrace{\boldsymbol{\pi}\Gamma^{-1}}(\mathbf{I}-\mathbf{E})^{-t_0}\mathbf{E}^{t_0-1}\underbrace{\Gamma\mathbf{P}\Gamma^{-1}}t_1!\cdots t_m!\Gamma^{-1}(\mathbf{I}-\mathbf{E})^{-t_m}\mathbf{E}^{t_0-1}\underbrace{\Gamma\mathbf{P}\Gamma^{-1}}\Gamma\mathbf{e},$$

such that $\boldsymbol{\pi}\Gamma^{-1}\mathbf{e}=\mathbf{e}$, because $\boldsymbol{\pi}\Gamma^{-1}\Gamma\mathbf{e}=\mathbf{e}$. Notice that $\boldsymbol{\pi}\Gamma^{-1}\Gamma\mathbf{P}\Gamma^{-1}\neq\boldsymbol{\pi}\Gamma^{-1}$, because we consider arbitrary initial distribution, i.e. $\boldsymbol{\pi}\mathbf{P}\neq\boldsymbol{\pi}$ in general. However, Theorem 2 in [BKK10] and Theorem 4 in [TH07] both consider the stationary processes of MMAPs, such that $\boldsymbol{\pi}\mathbf{P}=\boldsymbol{\pi}$. Thus, in their setting the vector $\boldsymbol{\pi}\Gamma^{-1}$ is determined by the matrix $\Gamma\mathbf{P}\Gamma^{-1}$.

From the above decomposition, all joint factorial moments are determined by the vector $\boldsymbol{\pi}\Gamma^{-1}$, the matrix $\mathbf{E}$ and the matrix $\Gamma\mathbf{P}\Gamma^{-1}$, such that the vector $\boldsymbol{\pi}\Gamma^{-1}$ is determined by n elements, the diagonal matrix $\mathbf{E}$ is determined by n diagonal elements, and the matrix $\Gamma\mathbf{P}\Gamma^{-1}$ is determined by $n(n-1)$ elements, because $\Gamma\mathbf{P}\Gamma^{-1}\mathbf{e}=\Gamma\mathbf{P}\mathbf{e}=\Gamma\mathbf{e}=\mathbf{e}$. Thus, totally n^2+n factorial moments shall decide all joint factorial moments in the base case, which is identical to the characterisation of a discrete–time MAP.

In analogy with the base case, we consider joint factorial moments with generally k arrival marks. Let $\mathbf{P}_0=(\mathbf{I}-\mathbf{D}_0)^{-1}\mathbf{D}_{\boldsymbol{h}_0}$, $\cdots$, $\mathbf{P}_{k-1}=(\mathbf{I}-\mathbf{D}_0)^{-1}\mathbf{D}_{\boldsymbol{h}_{k-1}}$, without loss of generality, we consider

$$\mathbb{E}\left(\Theta_0\ldots(\Theta_0-(t_0-1))\mathbf{1}(\boldsymbol{H}_0=\boldsymbol{h}_0)\cdots\Theta_m\ldots(\Theta_m-(t_m-1))\mathbf{1}(\boldsymbol{H}_m=\boldsymbol{h}_{k-1})\right)$$

$$= \boldsymbol{\pi}t_0!(\mathbf{I}-\mathbf{D}_0)^{-(t_0+1)}\mathbf{D}_0^{t_0-1}\mathbf{D}_{\boldsymbol{h}_0}\cdots t_m!(\mathbf{I}-\mathbf{D}_0)^{-(t_m+1)}\mathbf{D}_0^{t_m-1}\mathbf{D}_{\boldsymbol{h}_{k-1}}\mathbf{e}$$

$$= \boldsymbol{\pi}t_0!(\mathbf{I}-\mathbf{D}_0)^{-t_0}\mathbf{D}_0^{t_0-1}\underbrace{(\mathbf{I}-\mathbf{D}_0)^{-1}\mathbf{D}_{\boldsymbol{h}_0}}\cdots t_m!(\mathbf{I}-\mathbf{D}_0)^{-t_m}\mathbf{D}_0^{t_m-1}\underbrace{(\mathbf{I}-\mathbf{D}_0)^{-1}\mathbf{D}_{\boldsymbol{h}_{k-1}}}\mathbf{e}$$

$$= \boldsymbol{\pi}t_0!(\Gamma^{-1}\Gamma-\Gamma^{-1}\mathbf{E}\Gamma)^{-t_0}(\Gamma^{-1}\mathbf{E}\Gamma)^{t_0-1}\mathbf{P}_0\cdots t_m!(\Gamma^{-1}\Gamma-\Gamma^{-1}\mathbf{E}\Gamma)^{-t_m}(\Gamma^{-1}\mathbf{E}\Gamma)^{t_m-1}$$
$$\mathbf{P}_{k-1}\mathbf{e}$$

$$= \boldsymbol{\pi}t_0!(\Gamma^{-1}(\mathbf{I}-\mathbf{E})\Gamma)^{-t_0}\Gamma^{-1}\mathbf{E}^{t_0-1}\Gamma\mathbf{P}_0\cdots t_m!(\Gamma^{-1}(\mathbf{I}-\mathbf{E})\Gamma)^{-t_m}\Gamma^{-1}\mathbf{E}^{t_0-1}\Gamma\mathbf{P}_{k-1}\Gamma^{-1}\Gamma\mathbf{e}$$

$$= t_0!\underbrace{\boldsymbol{\pi}\Gamma^{-1}}(\mathbf{I}-\mathbf{E})^{-t_0}\mathbf{E}^{t_0-1}\underbrace{\Gamma\mathbf{P}_0\Gamma^{-1}}t_1!\cdots t_m!\Gamma^{-1}(\mathbf{I}-\mathbf{E})^{-t_m}\mathbf{E}^{t_0-1}\underbrace{\Gamma\mathbf{P}_{k-1}\Gamma^{-1}}\Gamma\mathbf{e}.$$

Therefore, all joint factorial moments are decided by the vector $\boldsymbol{\pi}\Gamma^{-1}$, the matrix $\mathbf{E}$ and the matrices $\Gamma\mathbf{P}_0\Gamma^{-1}$, ..., $\Gamma\mathbf{P}_{k-1}\Gamma^{-1}$. Such that the vector $\boldsymbol{\pi}\Gamma^{-1}$ is determined by n elements, the diagonal matrix $\mathbf{E}$ is determined by n diagonal elements, and the matrices $\Gamma\mathbf{P}_0\Gamma^{-1}$, ..., $\Gamma\mathbf{P}_{k-1}\Gamma^{-1}$ are determined by $kn^2 - n$ elements, because $\Gamma\sum_{i=0}^{k-1}\mathbf{P}_i\Gamma^{-1}\mathbf{e} = \Gamma\sum_{i=0}^{k-1}\mathbf{P}_i\mathbf{e} = \Gamma\mathbf{e} = \mathbf{e}$. Summing up, we have the first $kn^2 + n$ factorial moments are sufficient to characterise a discrete–time MMAP. $\qquad\square$

The decidability for redundant MMAPs is more complicated. The work in [Tel11] states that fewer independent joint factorial moments are required than in the non-redundant case. However one shall be especially careful on identifying independent joint factorial moments. It turns out that the complexity is too high for practical purposes. We leave this problem as an open question, which requires further investigations.

4.3   Time-lapse Bisimulation

A discrete–time marked Markovian arrival process is a stochastic process involving successive interarrival times of phase–type with marks on arrivals. The stochastic equivalence for MMAPs is an equivalence relation for the distributions of interarrival times and the distributions of arrival marks. In discrete–time, an interarrival time is the number of jumps in the Markov chain before the next arrival. In this section, we show how to formalise the interarrival time of MMAPs to explore a new bisimulation relation for capturing probabilistic behaviour over time.

4.3.1   Labelled DTMC

A labelled DTMC is a discrete–time Markov chain with action labels on the transitions, which is purely probabilistic without nondeterminism.

DEFINITION 4.6 (*Labelled DTMC*) A labelled DTMC is a tuple $(S, Act_\tau, \mathbb{P})$, where

- S is a finite set of states,

- Act_τ is a finite set of actions that contains the internal action τ and the external actions $Act = Act_\tau \setminus \{\tau\}$,

- $\mathbb{P}$ is a function: $S \times Act_\tau \times S \longrightarrow [0, 1]$ such that $\Sigma_{(a,s') \in Act_\tau \times S} \mathbb{P}(s, a, s')$ = 1 for all $s \in S$.

An *execution* ϖ in $(S, Act_\tau, \mathbb{P})$ is an infinite sequence involving states and actions. Let $s_0, s_1, \cdots \in S$ and $a_1, a_2, \cdots \in Act_\tau$, then an execution $\varpi = s_0 \overset{a_1}{\rightarrow} s_1 \overset{a_2}{\rightarrow} \ldots$ must satisfy that $\mathbb{P}(s_{i-1}, a_i, s_i) > 0$ for $i = 1, 2, \ldots$. Let k be a finite index number, we express the *k-th prefix of ϖ* as $\varpi^k = s_0 \overset{a_1}{\rightarrow} s_1 \overset{a_2}{\rightarrow} \ldots \overset{a_k}{\rightarrow} s_k$. We define $first(\varpi) = first(\varpi^k) = s_0$, $last(\varpi^k) = s_k$, $length(\varpi^k) = k$, $trace(\varpi^k) = a_1 a_2 \ldots a_k$. A *probability measure μ on ϖ^k* is uniquely defined by setting $\mu(\varpi^k) = \mathbb{P}(s_0, a_1, s_1) \times \mathbb{P}(s_1, a_2, s_2) \times \cdots \times \mathbb{P}(s_{k-1}, a_k, s_k)$.

A discrete–time MMAP and a labelled DTMC.   To transform an MMAP representation into a labelled DTMC (the other direction reverses), we map each element of matrices from an MMAP representation to a transition function of a labelled DTMC. Without loss of generality, assuming mark vector $\boldsymbol{h}_1$ is replaced by external action a_1, mark vector $\boldsymbol{h}_2$ is replaced by external action a_2, and so on, an MMAP representation with finite marks $(\boldsymbol{\pi}, \mathbf{D}_0, \mathbf{D}_{a_1}, \mathbf{D}_{a_2}, \ldots)$ describes a $(S, Act_\tau, \mathbb{P})$, such that $\mathbb{P}(s_i, \tau, s_j) = (D_0)_{i,j}$, $\mathbb{P}(s_i, a_1, s_j) = (D_{a_1})_{i,j}$, $\mathbb{P}(s_i, a_2, s_j) = (D_{a_2})_{i,j}$, $\ldots$. Therefore, an MMAP representation and a labelled DTMC are inter-definable.

Let $\mathbb{P}(s, a, C) = \sum_{s' \in C} \mathbb{P}(s, a, s')$ be the probability from state s to reach a set C via an a transition, and let $S/\mathcal{R}$ denote the set of equivalence classes under the equivalent relation $\mathcal{R}$, and let us next recall the definitions of strong and weak bisimulations [JS90, BH97].

DEFINITION 4.7 (***Strong Bisimulation***) Given a labelled DTMC $(S, Act_\tau, \mathbb{P})$, a strong bisimulation is an equivalence relation $\mathcal{R}$ on S such that for all $(p, q) \in \mathcal{R}$, all actions $a \in Act_\tau$ and all $C \in S/\mathcal{R}$: $\mathbb{P}(p, a, C) = \mathbb{P}(q, a, C)$. Two states p, q are called strong bisimilar, denoted $p \sim q$, iff $(p, q) \in \mathcal{R}$ for some strong bisimulation $\mathcal{R}$.

It has been proved in [JS90] that the relation $\sim = \bigcup\{\mathcal{R} | \mathcal{R}$ is a strong bisimulation$\}$ is an equivalence relation and a strong bisimulation relation, and hence is the largest strong bisimulation relation.

In strong bisimulation, internal actions, denoted τ, have to be simulated step by step as is also the case for external actions. However, since internal actions may be regarded as being invisible, there is no direct need to observe each step of internal transitions. This motivates defining weak bisimulation by abstracting

a sequence of internal transitions. The regular expression $\tau^*a\tau^*$ defines the language $\{\tau^n a\tau^m | n, m \geq 0\}$, which identifies the set of traces for weak transitions. Let ϵ denote the empty trace, the probability for a state s to reach a set C via a weak transition $\mathbb{P}(s, \tau^*a\tau^*, C)$ is to solve the equation system:

$$\mathbb{P}(s, \tau^*a\tau^*, C) = \begin{cases} 1, \text{ if } s \in C \text{ and } \epsilon \in \tau^*a\tau^*, \\ \sum_{(a',s') \in Act_\tau \times S} \mathbb{P}(s, a', s') \times \mathbb{P}(s', \tau^*a\tau^*/a', C), \text{ otherwise,} \end{cases}$$

such that $\tau^*a\tau^*/a' = \{\lambda \mid a'\lambda \in \tau^*a\tau^*\}$.

DEFINITION 4.8 (*Weak Bisimulation*) Given a labelled DTMC $(S, Act_\tau, \mathbb{P})$, a weak bisimulation is an equivalence relation $\mathcal{R}$ on S such that for all $(p, q) \in \mathcal{R}$, all actions $a \in Act_\tau$ and all $C \in S/\mathcal{R}$: $\mathbb{P}(p, \tau^*a\tau^*, C) = \mathbb{P}(q, \tau^*a\tau^*, C)$. Two states p, q are called weak bisimilar, denoted $p \approx q$, iff $(p, q) \in \mathcal{R}$ for some weak bisimulation $\mathcal{R}$.

It has been proved in [JS90] that the relation $\approx = \bigcup\{\mathcal{R} | \mathcal{R} \text{ is a weak bisimulation}\}$ is an equivalence relation and a weak bisimulation relation, and hence is the largest weak bisimulation relation.

4.3.2 Time-lapse Bisimulation

Stochastic equivalence for MMAPs characterises the complete sequence of arrivals associated with their interarrival times, arrival marks and probabilities. Consider two executions on labelled DTMCs, they are stochastically equivalent iff they not only agree on what sequence of actions they can perform, but also agree on all the time intervals between actions and all the probabilities of performing each action. To transform stochastic equivalence into a state based bisimulation relation, the behaviour of bisimilar states shall be that each external action performs with the same probability after the same period.

At first, we define an interarrival time to be the accumulated number of internal transitions between two successive external actions on labelled DTMCs. Therefore, for each external action we shall count the number of internal actions on its prefix to determine the distribution of an interarrival time. Recall Act is the set of external actions, we then define a new transition relation on labelled DTMCs, named *timed transition*.

DEFINITION 4.9 (*Timed Transition*) Let $s, s' \in S$, $t \in \mathbb{N}_0$ and $a \in Act$, a timed transition, $s \xrightarrow{t,a} s'$, is defined as a sequence of ordinary transitions such

that for some states $s_1, \ldots, s_t$

$$s \overset{t,a}{\longmapsto} s' \quad = \quad \begin{cases} s \overset{a}{\to} s', \text{ if } t = 0, \\ s \overset{\tau}{\to} s_1 \cdots \overset{\tau}{\to} s_t \overset{a}{\to} s', \text{ if } t > 0. \end{cases}$$

The probability of a timed transition is given as $\mathbb{P}(s, \tau^t a, s') = \mu(s \overset{t,a}{\longmapsto} s') = \sum_{s_1, \ldots, s_t} \mathbb{P}(s, \tau, s_1) \times \cdots \times \mathbb{P}(s_t, a, s')$.

A timed transition $s \overset{t,a}{\longmapsto} s'$ includes the time t, the external action type a and implicitly the probability $\mathbb{P}(s, \tau^t a, s')$. Two timed transitions are identical iff t, a and $\mathbb{P}(s, \tau^t a, s')$ all agree. The regular expression $\tau^t a$ identifies the corresponding set of traces for a timed transition. Thus, the probability for s to reach a set C via a timed transition is $\mathbb{P}(s, \tau^t a, C) = \sum_{s' \in C} \mathbb{P}(s, \tau^t a, s')$. We are now ready to introduce time-lapse bisimulation.

DEFINITION 4.10 (***Time-lapse Bisimulation***) Given a labelled DTMC (S, $Act_\tau, \mathbb{P}$), a time-lapse bisimulation is an equivalence relation $\mathcal{R}$ on S such that for all $(p, q) \in \mathcal{R}$, all external actions $a \in Act$ and all $C \in S/\mathcal{R}$: $\mathbb{P}(p, \tau^t a, C)$ $= \mathbb{P}(q, \tau^t a, C)$ for all t. Two states p, q are called time-lapse bisimilar, denoted $p \simeq q$, iff $(p, q) \in \mathcal{R}$ for some time-lapse bisimulation $\mathcal{R}$.

THEOREM 4.11 *For all labelled DTMCs, the relation $\simeq = \bigcup \{ \mathcal{R} | \mathcal{R}$ is a time-lapse bisimulation$\}$ is*

1. *an equivalence relation*

2. *a time-lapse bisimulation*

3. *the largest time-lapse bisimulation relation.*

PROOF Consider a labelled DTMC ($S, Act_\tau, \mathbb{P}$), some states $s_1, s_1', s_2, s_2', s_3, s_3' \in S$, and some time-lapse bisimulations $\mathcal{R}, \mathcal{R}', \mathcal{R}''$. We prove the statements in turn.

1. To show $\simeq$ is an equivalence relation, we argue that $\simeq$ is reflexive, symmetric and transitive. *Reflexivity*: the identity relation $\mathcal{I} = \{(s, s) \mid s \in S\}$ is clearly a time-lapse bisimulation. *Symmetry*: consider the relation $\mathcal{R}^{-1} = \{(s_2, s_1) \mid (s_1, s_2) \in \mathcal{R}\}$ that is obtained by swapping any pair of states in $\mathcal{R}$. By the symmetry of time-lapse bisimulation $\mathcal{R}$, we know $\mathcal{R}^{-1}$ is a time-lapse bisimulation. *Transitivity*: consider the relation $\mathcal{R}^{tr} = \{(s_1, s_3) \mid (s_1, s_2) \in \mathcal{R} \land (s_2, s_3) \in \mathcal{R}'\}$. By $(s_1, s_2) \in \mathcal{R}$, it follows that

$\mathbb{P}(s_1, \tau^t a, C) = \mathbb{P}(s_2, \tau^t a, C)$ for all timed transitions $\overset{t,a}{\longmapsto}$ to all equivalence classes C. By $(s_2, s_3) \in \mathcal{R}'$, it follows that $\mathbb{P}(s_2, \tau^t a, C) = \mathbb{P}(s_3, \tau^t a, C)$ for all timed transitions $\overset{t,a}{\longmapsto}$ to all C. That implies $\mathbb{P}(s_1, \tau^t a, C) = \mathbb{P}(s_3, \tau^t a, C)$ for all timed transitions $\overset{t,a}{\longmapsto}$ to all C. Therefore, $(s_1, s_3) \in \mathcal{R}''$, and the relation $\mathcal{R}^{tr}$ is a time-lapse bisimulation.

2. To show that $\simeq$ is itself a time-lapse bisimulation, assume $s_1 \simeq s_2$. Then, there exists a $\mathcal{R}$ that $(s_1, s_2) \in \mathcal{R}$. It follows that for any timed transition $s_1 \overset{t,a}{\longmapsto} s_1'$, there is a timed transition $s_2 \overset{t,a}{\longmapsto} s_2'$ with $(s_1', s_2') \in \mathcal{R}'$. Since $\simeq$ is a union of all time-lapse bisimulations, we have $s_1' \simeq s_2'$.

3. Because $\simeq$ is a time-lapse bisimulation, the definition $\bigcup\{\mathcal{R}|\mathcal{R}$ is a time-lapse bisimulation$\}$ immediately yields that all time-lapse bisimulation are included in $\simeq$. That is $\simeq$ is the largest time-lapse bisimulation relation.

$\square$

Similar to strong bisimulation and weak bisimulation, one could lift time-lapse bisimulation to system level. Therefore, we have two labelled DTMCs are *time-lapse bisimilar* iff for each initial state of one system there always exist a corresponding time-lapse bisimilar initial state of the other, such that they have the same initial probability.
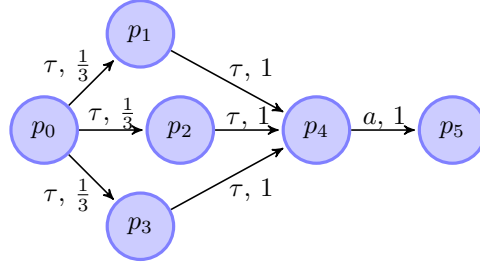
Figure 4.1: An example of time-lapse bisimulation: $p_1 \simeq p_2 \simeq p_3$.

EXAMPLE 4.1 *The labelled DTMC in Fig. 4.1 has $p_1 \simeq p_2 \simeq p_3$, because they have the same probabilistic timed behaviour, i.e. to perform an external action a after 1 time unit.*

Example 4.1 illustrates that probabilistic timed properties are captured by time-lapse bisimulation relations on labelled DTMCs, such that, for two time-lapse bisimilar states, one would expect that same external actions will be performed with the same probability after waiting the same amount of time. It would

be interesting to see how time-lapse bisimulation used in practice to improve verification. However, this is outside the scope of this work.

4.3.3   Comparative Study of Bisimulation Relations

As a new member of the bisimulation family, in this part, we compare time-lapse bisimulation with strong bisimulation and weak bisimulation.

Time-lapse Bisimulation meets Weak Bisimulation   Internal actions represent invisible computations, which have been studied and proved useful in modelling concurrent and hierarchical systems. In terms of the power of observation, in weak bisimulation arbitrarily many internal actions may be removed such that the probabilistic behaviour of the system is still preserved. With respect to stochastic equivalence, time-lapse bisimulation counts the number of internal actions to express how a stochastic system behaves probabilistically when time elapses. We clarify that neither time-lapse bisimulation nor weak bisimulation implies the other.

THEOREM 4.12 *Time-lapse bisimulation is not comparable with weak bisimulation.*

PROOF We show two scenarios that the first one holds $p \simeq q$ but $p \not\approx q$, and the second holds $p \approx q$ but $p \not\simeq q$.
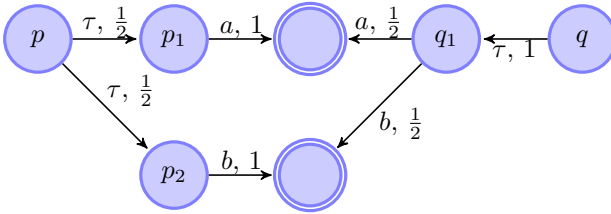
Figure 4.2: Time-lapse bisimilar states are neither necessarily weak bisimilar nor necessarily strong bisimilar: $p \simeq q$ but $p \not\approx q$ and $p \not\sim q$.

The labelled DTMC in Fig. 4.2 holds $p \simeq q$ but $p \not\approx q$. The $p \simeq q$ is because they have identical timed transitions $\overset{1,a}{\longmapsto}$ and $\overset{1,b}{\longmapsto}$. The $p \not\approx q$ is because $p_1 \not\approx p_2 \not\approx q_1$, however this fact does not affect time-lapse bisimulation. Recall the definition of $p \simeq q$, we have for all external actions $a \in Act$ and $C \in S/\mathcal{R}$ such that $\mathbb{P}(p, \tau^t a, C) = \mathbb{P}(q, \tau^t a, C)$ for all t. This requires that the equivalence classes

shall be reached only by external actions. However, in weak bisimulation, the action a in $\tau^* a \tau^*$ can be an internal action, so a sequence of τ actions could reach new equivalence classes. In Fig. 4.2, p_1, p_2 and q_1 are different equivalence classes of weak bisimulation reached by τ action.
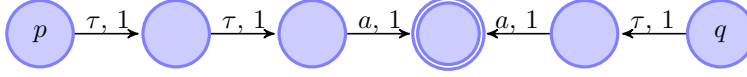
Figure 4.3: Weak bisimilar states are not necessarily time-lapse bisimilar: $p \approx q$ but $p \not\simeq q$.

The labelled DTMC in Fig. 4.3 holds $p \approx q$ but $p \not\simeq q$. Because weak bisimulation disregards of the time, i.e. the number of τ actions, but time-lapse bisimulation regards. $\qquad\square$

Time-lapse Bisimulation meets Strong Bisimulation Strong bisimulation is finer than weak bisimulation, such that internal actions are simulated stepwise. We clarify that strong bisimulation is also finer than time-lapse bisimulation.

THEOREM 4.13 *Strong bisimulation implies time-lapse bisimulation but not vice versa.*

PROOF Let p and q be two states, we first show that $p \sim q$ implies $p \simeq q$. Recall the definition of $p \sim q$, we have $\mathbb{P}(p, a, C) = \mathbb{P}(q, a, C)$ for all actions $a \in Act_\tau$ and $C \in S/\mathcal{R}$. In general let $\tau^t a'$ be the label of a timed transition for $t \in \mathbb{N}_0$ and action $a' \in Act$. For some C and some states $s_1, \ldots, s_t$, we have $\mathbb{P}(p, \tau^t a', C) = \mathbb{P}(p, \tau, s_1) \times \ldots \times \mathbb{P}(s_t, a', C)$. By the stepwise definition of strong bisimulation, $p \sim q$ and $p \xrightarrow{\tau} s_1$ implies $q \xrightarrow{\tau} s'_1$ and $s_1 \sim s'_1$. We then use $s_1 \sim s'_1$ and $s_1 \xrightarrow{\tau} s_2$ to imply further strong bisimilar states until some $s_t \sim s'_t$. Therefore, we have $\mathbb{P}(p, \tau, s_1) \times \ldots \times \mathbb{P}(s_t, a', C) = \mathbb{P}(q, \tau, s'_1) \times \ldots \times \mathbb{P}(s'_t, a', C)$, and that is $\mathbb{P}(p, \tau^t a', C) = \mathbb{P}(q, \tau^t a', C)$. Thus, $p \sim q$ implies $p \simeq q$.

Now we show $p \simeq q$ does not necessarily imply $p \sim q$ by considering the previous example in Fig. 4.2. We have argued that $p \simeq q$ and $p_1 \not\approx p_2 \not\approx q_1$. Since weak bisimulation is coarser than strong bisimulation [AILS07], $p_1 \not\approx p_2 \not\approx q_1$ implies $p_1 \not\sim p_2 \not\sim q_1$. Thus, $p \not\sim q$. $\qquad\square$

From Theorem 4.12 and Theorem 4.13, we conclude that time-lapse bisimulation is a new bisimulation relation, which is coarser than strong bisimulation and is not comparable to weak bisimulation.

4.3.4 Logical Characterisation of Time-lapse Bisimulation

Logical characterisations [LS91, DEP98, PS07] permit to understand what properties are preserved under different bisimulation relations, and are also very useful for comparative analysis. In analogy with the results of logical characterisation for deterministic probabilistic systems [DEP98, PS07], we give a Hennessy-Milner style logic for characterising time-lapse bisimulation on labelled DTMCs.

DEFINITION 4.14 (*Logic $\mathcal{L}$*) Let p range over the rational numbers in $[0,1]$, then the logic $\mathcal{L}$ has as syntax the following formulas:

$$\varphi ::= \top \mid \varphi_1 \wedge \varphi_2 \mid \Diamond_p^t a\varphi,$$

The satisfaction relation, $s \vDash \varphi$, between states of a labelled DTMC $(S, Act_\tau, \mathbb{P})$ and formulas of $\mathcal{L}$ is given as follows:

$s \vDash \top$ holds always true.

$s \vDash \varphi_1 \wedge \varphi_2$ holds iff $s \vDash \varphi_1$ and $s \vDash \varphi_2$.

$s \vDash \Diamond_p^t a\varphi$ holds iff state s can make a timed transition $s \overset{t,a}{\longmapsto} s'$ such that $s' \vDash \varphi$ and $\mathbb{P}(s, \tau^t a, s') \geq p$.

The rest part of this section is dedicated to the proofs of soundness and completeness of Hennessy-Milner style logic $\mathcal{L}$. We shall now introduce the notations to be used. Let φ, ψ range over formulas of the logic $\mathcal{L}$. We define the *depth*, denoted n, of a formula φ as the maximum number of nested diamond operators that occur in φ. Let $s \simeq_n r$ denote that $s \simeq r$ holds at the formula of depth n, $\mathcal{F}_n$ be the sets of the formulas of depth at most n, $\mathcal{F}(s)$ be the sets of the formulas which are satisfied by the state s and $\mathcal{F}_n(s)$ be the sets of the formulas of depth at most n which are satisfied by the state s.

LEMMA 4.15 *Given a $(S, Act_\tau, \mathbb{P})$ and $s, r \in S$, $\mathcal{F}_0(s) = \mathcal{F}_0(r)$ for each pair of time-lapse bisimilar states s and r.*

This preliminary lemma shows that the time-lapse bisimilar states of a labelled DTMC satisfy the same sets of formulas of depth zero.

THEOREM 4.16 *Given a $(S, Act_\tau, \mathbb{P})$ and $s, r \in S$, then $s \simeq r$ iff s and r obey the same formulae of $\mathcal{L}$.*

PROOF By induction on n, we show that $s \simeq_{n+1} r$ iff $\mathcal{F}_{n+1}(s) = \mathcal{F}_{n+1}(r)$ for each $n \geq 0$. The base case $n = 0$ holds trivially by Lemma 4.15 and definition of $\simeq_0$. For the inductive step we prove two directions of our claim separately.

For soundness ($\Rightarrow$). Let $s \simeq_{n+1} r$, we show $s \vDash \varphi \Leftrightarrow r \vDash \varphi$ by structural induction on $\varphi \in \mathcal{F}_{n+1}$ that $s \vDash \varphi$ iff $r \vDash \varphi$. Assume $s \vDash \varphi$ (the case for $r \vDash \varphi$ is symmetric). If $\varphi = \top$, then $r \vDash \varphi$ trivially. If $\varphi = \psi_1 \wedge \psi_2$, then $s \vDash \psi_1$ and $s \vDash \psi_2$. By structural induction, $r \vDash \psi_1$ and $r \vDash \psi_2$, and thus $r \vDash \varphi$. If $\varphi = \Diamond_p^t a\psi$, then $\psi \in \mathcal{F}_n$. By definition, there exists a timed transition $s \overset{t,a}{\longmapsto} s'$ such that $s' \vDash \psi$ and $\mathbb{P}(s, \tau^t a, s') \geq p$. From $s \simeq_{n+1} r$, there exists an identical timed transition $r \overset{t,a}{\longmapsto} r'$ such that $r' \vDash \psi$, $\mathbb{P}(r, \tau^t a, r') \geq p$ and $s' \simeq_n r'$. That is, $r \vDash \Diamond_p^t a\psi$.

For completeness ($\Leftarrow$). We show that $s \not\simeq_{n+1} r$ implies $\mathcal{F}_{n+1}(s) \neq \mathcal{F}_{n+1}(r)$. Assume $s \not\simeq_{n+1} r$ and for the sake of contradiction assume that $\mathcal{F}_{n+1}(s) = \mathcal{F}_{n+1}(r)$. Without loss of generality, assume there exits a timed transition $s \overset{t,a}{\longmapsto} s'$, which r doesn't have. The timed transition implies $s \vDash \Diamond_p^t a\psi$ and $s' \vDash \psi$. For the state r, there must not exist r' that $r \overset{t,a}{\longmapsto} r'$ and $s' \simeq_n r'$. That is $r \not\vDash \Diamond_p^t a\psi$ and $r' \vDash \psi$. That is the formula holds only for $\mathcal{F}_{n+1}(s)$ but not $\mathcal{F}_{n+1}(r)$, a contradiction appears as needed. $\square$

4.3.5   A Case Study

In this part, we show the novelty of time-lapse bisimulation through a dependable model of 2-stage safety device in Fig. 4.4.

Let us assume a safety device is designed to have two engines, such that the device can automatically switch to the second engine when the first failed. The device is broken once both engines failed. Assume now, one of two engines is reported to have defect, engineers would like to know how likely the device breaks after a certain period. Given that $1 - p_1$ is the probability of failure per day for regular engines, and $1 - p_2$ is the probability of failure per day for defective engines, we model this device as a labelled DTMC in Fig. 4.4-(a). The failure action f is the only observable/external action in the model. Since we don't know either normal or defective engine operates first, we use a uniform probability distribution to enclose both cases.

Now we illustrate how bisimulations perform in the case. Unluckily, it is impossible to reduce state space using strong bisimulation. On the other hand, with no respect of time, state reduction using weak bisimulation shall aggregate all τ-enabled states, which is useless if one is interested at time-bounded reacha-
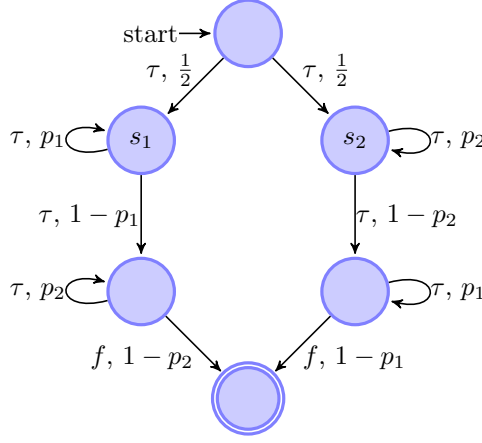
(a) Original Device.
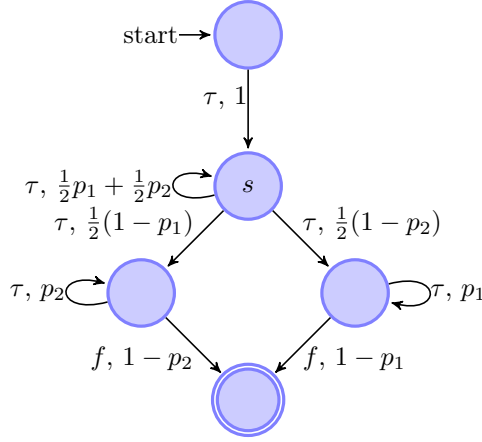
(b) Reduced Device.

Figure 4.4: A probabilistic model of 2-stage safety device: $s_1 \simeq s_2 \simeq s$.

bility properties. E.g. what is the probability that the device is broken after 40 days? However, time-lapse bisimulation successfully aggregates states s_1, s_2 into state s as showed in Fig. 4.4-(b).

This case study sheds some light on the novelty of time-lapse bisimulation, one of our future aims is to apply this new bisimulation relation on practical benchmark cases.

4.4   Bisimulation Characterisation of Stochastic Equivalence

We have presented time-lapse bisimulation on labelled DTMCs in order to characterise stochastic equivalence for discrete–time marked Markovian arrival process. In this section, we show that bisimulation characterisation of stochastic equivalence for some bisimulation relations are sound in that two bisimilar states are also stochastically equivalent, but for all bisimulation relations that two stochastically equivalent states are not necessarily bisimilar.

We have shown that a representation of an MMAP and a labelled DTMC are inter-definable, therefore for labelled DTMCs we obtain the following observation on stochastic equivalence for MMAPs, which is identical to Definition 4.3.

OBSERVATION 1 *Given a labelled DTMC $(S, Act_\tau, \mathbb{P})$ and some $s, r, \bar{s}, \bar{r} \in S$, s and r are stochastically equivalent if for all $n \in \mathbb{N}$, $t_1, \dots, t_n \in \mathbb{N}_0$ and external actions $a_1, \dots, a_n \in Act$: $\mathbb{P}(s, \tau^{t_1} a_1 \dots \tau^{t_n} a_n, \bar{s}) = \mathbb{P}(r, \tau^{t_1} a_1 \dots \tau^{t_n} a_n, \bar{r})$, where $\tau^{t_1} a_1 \dots \tau^{t_n} a_n$ is a sequence of timed transitions.*

Having Observation 1 for labelled DTMCs, we are now ready to clarify the theorem of time-lapse bisimulation characterisation.

THEOREM 4.17 *Time-lapse bisimulation is a sound but not complete characterisation of stochastic equivalence.*

PROOF Given a labelled DTMC $(S, Act_\tau, \mathbb{P})$ and let $s, \bar{s}, r, \bar{r} \in S$, we first show the soundness of time-lapse bisimulation characterisation. Because $s \simeq r$, states s and r have the same set of timed transitions. That is $s \simeq r$ implies $\mathbb{P}(s, \tau^{t_1} a_1, s') = \mathbb{P}(r, \tau^{t_1} a_1, r')$ for all $t_1 \in \mathbb{N}_0$, $a_1 \in Act$. By time-lapse bisimilarity $\simeq$ being a time-lapse bisimulation (cf. Theorem 4.11), we have that $s \simeq r$ also implies $\mathbb{P}(s, \tau^{t_1} a_1 \tau^{t_2} a_2, s'') = \mathbb{P}(r, \tau^{t_1} a_1 \tau^{t_2} a_2, r'')$ for all $t_1, t_2 \in \mathbb{N}_0$ and $a_1, a_2 \in Act$. The same step replays for any $n \in \mathbb{N}$, that is $s \simeq r$ implies $\mathbb{P}(s, \tau^{t_1} a_1 \dots \tau^{t_n} a_n, \bar{s}) = \mathbb{P}(r, \tau^{t_1} a_1 \dots \tau^{t_n} a_n, \bar{r})$ for all $t_1, \dots, t_n \in \mathbb{N}_0$ and external actions $a_1, \dots, a_n \in Act$. That is Observation 1 holds. Therefore, time-lapse bisimulation implies stochastic equivalence.

Now we show that stochastic equivalence does not necessarily imply time-lapse bisimulation by considering the τ-free example in Fig. 4.5. For τ-free labelled DTMCs, time-lapse bisimulation, strong bisimulation and weak bisimulation all coincide. Thus, we get $p \not\simeq q$, because $p' \not\simeq q' \not\simeq q''$. However, apparently the probabilities of all the sequences of timed transitions, i.e. a, ab and ac,
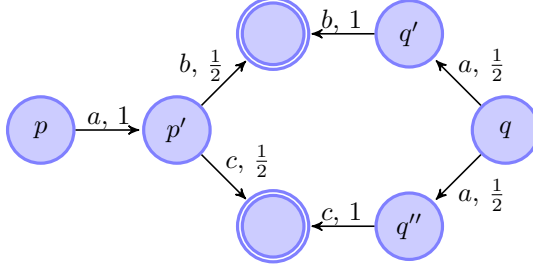
Figure 4.5: Stochastic equivalence does not necessarily imply the bisimulations: $p \not\sim q$, $p \not\simeq q$, $p \not\approx q$.

are the same. That's Observation 1 holds, therefore p and q are stochastically equivalent. $\square$

THEOREM 4.18 *Strong bisimulation is a sound but not complete characterisation of stochastic equivalence.*

PROOF For soundness, Theorem 4.17 states that time-lapse bisimulation is a sound characterisation of stochastic equivalence. Given Theorem 4.13 that strong bisimulation is finer than time-lapse bisimulation, we have that strong bisimilar states are time-lapse bisimilar, and are also stochastically equivalent.

For non-completeness, the τ-free example in Fig. 4.5 shows two stochastic equivalent states are not strong bisimilar. $\square$

THEOREM 4.19 *Weak bisimulation is neither sound nor complete characterisation of stochastic equivalence.*

PROOF For soundness, Theorem 4.17 states that time-lapse bisimulation is sound to stochastic equivalence. Given Theorem 4.12 that weak bisimilar states are not necessarily time-lapse bisimilar, we have weak bisimulation is not sound to stochastic equivalence.

For non-competeness, the τ-free example in Fig. 4.5 shows two stochastic equivalent states are not weak bisimilar. $\square$

4.5   Summary

There have been many papers on the stochastic extension of labelled transition systems for analysing quantitative properties, such as performance, reliability, timeliness, and efficiency. Through stochastic models and measurements, quantitative properties are evaluated and verified for computer systems and networks. Among them, equivalence relations are defined to capture when two systems have the same behaviour. As a co-inductive characterisation of equivalence relations, bisimulation relations allow one to reduce a system to an equivalent but smaller system, which is obtained by replacing each state in a system by its bisimulation equivalence class. Such a state aggregation technique is commonly used as a preprocessing step for model checking.

In this work, we studied the stochastic equivalence for discrete phase–type distributions and discrete–time marked Markovian arrival processes. Stochastically equivalent systems have the same probability distributions of the interarrival times and the arrival marks. Transforming discrete–time MMAPs to labelled DTMCs, we defined the time-lapse bisimulation to characterise stochastic equivalence. As a new bisimulation relation, time-lapse bisimilar states have the same probabilistic behaviour of performing external actions after waiting the same amount of time. We clarified that time-lapse bisimulation is coarser than strong bisimulation, and is not comparable to weak bisimulation. A Hennessy-Milner style logic was identified to characterise time-lapse bisimulation. As a co-inductive characterisation of stochastic equivalence, we showed that time-lapse bisimulation and strong bisimulation is a sound but not complete characterisation of stochastic equivalence, and weak bisimulation is neither sound nor complete to stochastic equivalence.

Based on the current development, we would like to see a practical case study, where the advantages or drawbacks of the time-lapse bisimulation are discussed. It would be very interesting to see how the time-lapse bisimulation improves the verification of stochastic systems. For this, an implementation and experimental results would be necessary. On the other hand, naturally, we would like to define time-lapse bisimulation characterisation of stochastic equivalence for continuous–time MMAPs, where the corresponding transition systems are labelled continuous time Markov chains.

CHAPTER 5

Stochastic Model Checking without Markov Chains

In recent years, stochastic model checking on continuous–time Markov chains has been widely studied and applied to analyse quantitative properties of stochastic systems, such as performance and reliability. The main obstacle to stochastic model checking in practice is the *state space explosion* problem. In this chapter, we pursue the *minimality* of a stochastic system to attack the state space explosion problem, which, however, takes us outside the world of Markov chains. We propose a *process algebra* MEME, which takes *multi-exits matrix–exponential distributions* as primitives. In our language, all the components before and after compositions are secured to have a minimal state space representation, and the overall system is still compatible with the standard Markov chain model checking techniques.

5.1 Introduction

Many researchers have studied different ways of expressing continuous time stochastic systems and how to argue that they fulfil certain properties often expressed in logical form. Among the various approaches, stochastic process algebras are widely studied and applied to characterise the systems. The promi-

nent ones, such as IMC [Her02], PEPA [Hil96], TIPP [GHR92], EMPA [BG96], and stochastic π-calculus [Pri95], all produce classic Markovian models in the form of Continuous Time Markov Chains (CTMCs) with various considerations of non-determinism. To express different properties, the logics for CTMCs are Continuous Stochastic Logic (CSL) [ASSB96, BHHK03] with the possibility of including rewards. The stochastic model checking against CSL formulae on CTMCs follow standard techniques in [BHHK03, KNP07].

A key consideration of stochastic process algebras is that of compositionality. Phase–type distributions were considered first in [Neu75, Neu81], and are defined as the distributions of the absorption times in Markov chains with a finite number of transient states and one absorbing state. While CPH distributions go a long way in providing the framework for a compositional construction of stochastic systems (cf. PHPA in Section 3.2), they lack the algorithmic support of finding the minimal representation that would enable efficient model checking against a suitable logic. At the same time even a closed system would have components that would be construed to be open. This calls for extending considered distributions along two directions: one is to allow for multiple exits corresponding to different absorbing states, the other is to extend our considerations from CPH distributions to the larger class of Matrix–Exponential (ME) distributions exactly because one shall always find a ME representation of minimal size. We name this generalised class of distributions as Multi-Exits Matrix–Exponential (MEME) distributions. In this work, we introduce a stochastic process algebra MEME to characterise the systems having MEME distributed durations, and we study the corresponding model checking techniques.

We solve the problem in the following manner. In Section 5.2, we introduce some interesting families of probability distributions to illustrate our idea behind MEME distributions. In Section 5.3, we present the formal development of MEME distributions and their compositionality results. A polynomial time algorithm shall be presented and summarised in the MATLAB language to obtain the minimality of a MEME representation. In Section 5.4, we introduce a MEME calculus to model the stochastic systems having MEME distributions, together with CSL formulae to express properties. In Section 5.5, we study transient and limiting probabilities through proper expansions and aggregations for a Markov renewal process with matrix–exponential kernels, which is compatible with standard CTMC solvers. We report our experimental results with a numerical example using MATLAB. In Section 5.6, we discuss model checking algorithms for the MEME system against CSL formulae. We also clarify how to transform a Markov renewal process to a semi-Markov chain to adapt model checking techniques of semi-Markov chains. We conclude our work in Section 5.7.

5.2   Background

Stochastic process algebras enrich process languages with random variables to model action durations (or delays before instantaneous actions). The majority of stochastic process algebras have random variables that are Exponentially (EXP) distributed, which ensures that the underlying stochastic process is a CTMC. However, this restriction to exponential durations is unrealistic when modelling many phenomena in areas such as road traffic and system biology, where the systems contain deterministic delays or highly variable distributed durations.

Continuous phase–type distributions can be used to approximate any kind of probability distributions with support on $[0, \infty)$ [JT88]. EXP, Erlang, Hypo-EXP, Hyper-EXP and Coxian distributions are examples of CPH distributions. Henceforth, some stochastic process algebras have gone one step further than EXP durations and dealt with CPH distributions, including MRP [NNN10], PTP [Wol08] and $\text{PEPA}_{ph}^{\infty}$ [ERKN99]. Multivariate PH distributions have been applied to calculate multivariate state rewards over CTMCs [NNN10]. The class of matrix–exponential distributions [AB99, BN03] includes and generalises CPH distributions, with minimal dimensionality as a crucial merit. A ME representation of a CPH distribution will be of lower order, or the same order, than the corresponding CPH representations [BN03]. An efficient algorithm exists to reduce a ME representation into its minimal order [AB99]. There exist examples [Bot86] that a ME distribution does not have a CPH representation of the same order, however it may be possible to embed the ME distribution into a higher dimensional space such that a CPH representation is obtained. The point processes as a natural extension of ME distributions are Rational Arrival Processes (RAPs) [AB99]. Recently, the equivalence relations and compositionality of marked RAPs have been studied in [BB12].

A *Markov Renewal Process* (MRP) is a continuous–time stochastic process such that the exponentially distributed sojourn times in a CTMC are now generalised to follow arbitrary probability distributions (cf. Section 2.2.5, Definition 2.9). If we let the arbitrary distributions be ME distributions, we get an MRP with matrix–exponential kernels. The stochastic process representing the states of an MRP at an arbitrary time is called a *Semi-Markov Process* (SMP).

Alternatively one can view the kernel of an MRP as a sequence of distributions with multiple exits, such that a distribution is extended from having one dimension to two dimensions: the first random variable expresses the delays that are EXP/CPH/ME distributed, while the second random variable expresses a random exit out of multiple candidates. With the second dimension, a stochastic delay is now able to trigger different successive routines with different probabilities. In terms of modelling power, this increases the flexibility of expressing

different scenarios in which a system might continue after a random holding time.

The relationships of our interesting distributions are summarised in the following table. The distribution on the destination side of an arrow is a generalisation of the originating side.

EXP Dist.	$\rightarrow$	CPH Dist.	$\rightarrow$	ME Dist.
$\downarrow$		$\downarrow$		$\downarrow$
Multi-Ex. EXP Dist.	$\rightarrow$	Multi-Ex. CPH Dist.	$\rightarrow$	Multi-Ex. ME Dist.

Markov Chain	**Markov Chain**	**Non Markov Chain**

Table 5.1: Relationships of Probability Distributions.

A stochastic process having (Multi-Exits) exponentially distributed holding times is clearly a Markov chain by definition. A stochastic process having (Multi-Exits) continuous phase–type distributed holding times is an MRP with CPH kernels, which could still be a Markov chain if one expands the CPH kernels into Markov chains. However, a stochastic process having (Multi-Exits) matrix–exponential distributed holding times is not a continuous–time Markov chain. Henceforth, in this work we deal with stochastic systems that in general are outside the world of Markov chains. We secure the minimality through compositional constructions, which shall be a nice feature for deploying stochastic model checking in practice.

5.3   Multi-Exits Matrix–Exponential Distributions

In this section, we introduce the formal definition of multi-exits matrix exponential distributions and their properties. We show that it is possible to construct a valid MEME distribution via mixing several independent matrix–exponential distributions. Then, we clarify the compositionality by introducing a number of operators that are closed within the class of MEME distributions. We propose an extension of the minimisation approach of [AB99] for MEME distributions, and summarise the minimality result in a polynomial time algorithm.

5.3.1 Definitions

DEFINITION 5.1 (*Matrix–Exponential Distribution*) A distribution on the positive real axis is called a Matrix–Exponential (ME) distribution if it has a possible atom at zero, and an absolutely continuous part represented by a density $f(\cdot)$ which can be written on the form

$$f(w) = \boldsymbol{\alpha} e^{\mathbf{S}w} \mathbf{s}^0, \text{ for } w > 0,$$

where $\boldsymbol{\alpha}$ is a $1 \times p$ row vector, $\mathbf{S}$ is a $p \times p$ matrix and $\mathbf{s}^0$ is a $p \times 1$ vector (complex entries are allowed).

For simplicity, throughout this chapter, we do not consider distributions with a point mass at zero, i.e. $\boldsymbol{\alpha}(-\mathbf{S})^{-1}\mathbf{s}^0 = 1$. As shown in [AB99], the $\boldsymbol{\alpha}$, $\mathbf{S}$ and $\mathbf{s}^0$ can always be chosen to have non-complex numbers, henceforth we shall only deal with ME distributions with real entries. Let W be a ME distributed random variable, the cumulative distribution function $F(\cdot)$, the moment generating function $M_W(\cdot)$ and n'th moment $\mathbb{E}(W^n)$ are given as: $F(w) = 1 + \boldsymbol{\alpha} e^{\mathbf{S}w} \mathbf{S}^{-1}\mathbf{s}^0$, $M_W(s) = \boldsymbol{\alpha}(-s\mathbf{I} - \mathbf{S})^{-1}\mathbf{s}^0$, and $\mathbb{E}(W^n) = n!\boldsymbol{\alpha}(-\mathbf{S})^{-n-1}\mathbf{s}^0$.

From [AB99] we know that, for any ME distribution, one can always choose a representation such that $\boldsymbol{\alpha}\boldsymbol{e} = 1$ and $\mathbf{s}^0 = -\mathbf{S}\boldsymbol{e}$; we shall do so in this part of the thesis to make $\mathbf{s}^0$ implicit. Henceforth, we refer to *the representation of ME distributions* as $\text{ME}_p(\boldsymbol{\alpha}, \mathbf{S})$, such that p is called *the order of the ME representation*. From ME distributions, we get the following observation, which is identical to the definition of CPH distributions (cf. Section 2.2.2, Definition 2.8).

OBSERVATION 2 *Let $W \sim ME_p(\boldsymbol{\alpha}, \mathbf{S})$, such that $\boldsymbol{\alpha}$ and $\mathbf{S}$ satisfy the following properties:*

- $\alpha_i \geq 0$ *and* $\boldsymbol{\alpha}\boldsymbol{e} = 1$,

- $\mathbf{S}_{ii} < 0$, $\mathbf{S}_{ij} \geq 0$ *for* $i \neq j$, *and* $\mathbf{S}\boldsymbol{e} \leq 0$,

- $\mathbf{S}$ *is non-singular,*

then W is continuous phase–type distributed with representation $CPH_p(\boldsymbol{\alpha}, \mathbf{S})$.

Both CPH distributions and ME distributions are probability distributions with support on $[0, \infty)$. In general, ME distributions contain and are stochastic equivalent (cf. Definition 4.1) with CPH distributions [AB99], therefore many results valid for CPH distributions also hold for ME distributions. However, the

probabilistic interpretation of CPH distributions on an absorbing CTMC is not valid in the general ME case. Instead, Bladt and Neuts in [BN03] have provided the interpretation as deterministic flows between containers. The advantage of ME distributions over CPH distributions primarily comes from dimensionality: in general, a ME representation of a CPH distribution will be of lower order than the corresponding CPH representation. We shall highlight the dimensionality advantage of ME distributions through an example.

EXAMPLE 5.1 *In stochastic modelling, Erlang distributions (an example of CPH distributions) are commonly used to model the random delay having low variation, e.g. approximating a deterministic delay. A family of ME distributions of order 3 governed by the parameters a and $\epsilon \geq 0$ can be made to have smaller coefficient of variation than the Erlang-3 distribution. The representation $ME_3(\boldsymbol{\alpha}, \mathbf{S})$ is given as*

$$\boldsymbol{\alpha} = \frac{1}{1 + \frac{1+a}{2}\epsilon^2 - \epsilon} \begin{bmatrix} 1 & -\epsilon & \frac{1+a}{2}\epsilon^2 \end{bmatrix}, \text{ and } \mathbf{S} = \begin{bmatrix} -\lambda & \lambda & 0 \\ 0 & -\lambda & \lambda \\ 0 & 0 & -\lambda \end{bmatrix} \text{ see e.g. [BN11].}$$
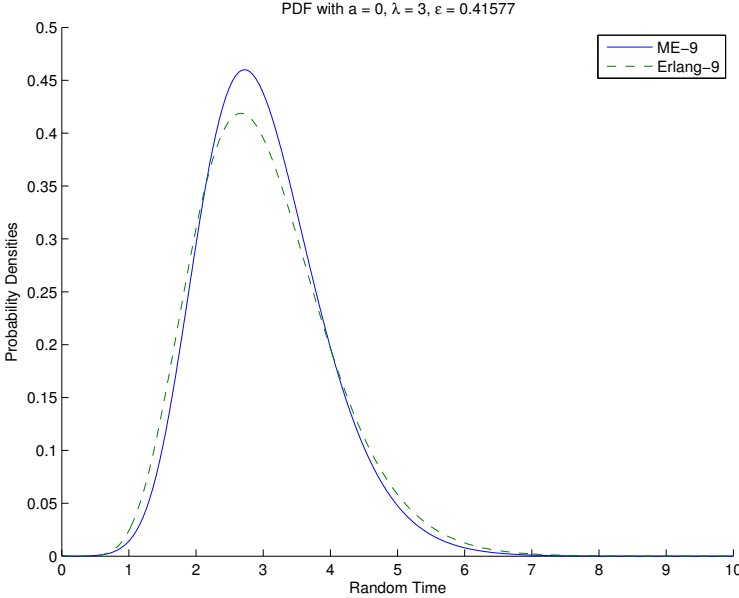

Figure 5.1: Density Plot: ME-9 versus Erlang-9.

Let ME-9 stand for the convolution of three $ME_3(\boldsymbol{\alpha}, \mathbf{S})$. From Fig. 5.1, we see that the ME-9 has a lower coefficient of variation than Erlang-9.

To model multiple exits, we extend the vector $\mathbf{s}^0$ to a matrix having multiple columns, such that each column corresponds to an exit. In this manner, we generalise ME distributions with an additional dimension to indicate a random exit among several possible candidates.

DEFINITION 5.2 (*Multi–Exists Matrix–Exponential Distribution*) Let n be a positive integer, a discrete random variable G be distributed on $[1, n]$, and $W \sim \text{ME}_p(\boldsymbol{\alpha}, \mathbf{S})$. A joint distribution of (W, G) is called a Multi-Exits Matrix–Exponential (MEME) distribution if the joint probability density function is defined by

$$f(w, g) = \boldsymbol{\alpha} e^{\mathbf{S}w} \mathbf{T}_g, \text{ for } w > 0 \text{ and } 1 \leq g \leq n, \tag{5.1}$$

where $\mathbf{T}_g$ is the g-th column of a $p \times n$ matrix $\mathbf{T}$. We term the matrix $\mathbf{T}$ a multi-exits matrix and write $(W, G) \sim \text{MEME}_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ to express that (W, G) is MEME distributed with representation $\text{MEME}_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ such that p is the order of the MEME representation (inherited from $\text{ME}_p(\boldsymbol{\alpha}, \mathbf{S})$) and n is the number of multiple exits.

The probabilistic interpretation of the joint density $f(w, g)$ is given as the probability density of a process finishing at the g-th exit at the w time units. From $f(w, g)$, one can derive $F(w, g)$, cumulative distribution function of the first variable w, as

$$F(w, g) = \mathbb{P}(W \leq w, G = g) = \int_0^w f(x, g) dx = \boldsymbol{\alpha} \mathbf{S}^{-1} (e^{\mathbf{S}w} - \mathbf{I}) \mathbf{T}_g. \tag{5.2}$$

The value of $F(w, g)$ represents the cumulated probability of a process finishing at the g-th exit after at most the w time units. From $F(w, g)$, one can calculate the marginal probability of a process eventually finishing at the g-th exit as

$$F(\infty, g) = \mathbb{P}(G = g) = \lim_{w \to \infty} F(w, g) = \boldsymbol{\alpha} (-\mathbf{S})^{-1} \mathbf{T}_g. \tag{5.3}$$

Like phase–type distributions (cf. Section 3.2.2), it is known that the class of ME distributions are closed under finite probabilistic mixture, minimum and maximum. For simplicity, we only show how a MEME representation having two exits is constructed from two independent ME representations. The construction for a general number of ME distributions follows the same pattern. Let $W_1 \sim \text{ME}_{p_1}(\boldsymbol{\alpha}_1, \mathbf{S}_1)$ and $W_2 \sim \text{ME}_{p_2}(\boldsymbol{\alpha}_2, \mathbf{S}_2)$ be two independent ME variables, we have the following results.

- Probabilistic mixture
  The random variable $W = I_1 W_1 + I_2 W_2$, such that $I_1 + I_2 = 1$, $\pi_1 + \pi_2 =$

1, $\mathbb{P}(I_1 = 1) = \pi_1$ and $\mathbb{P}(I_2 = 1) = \pi_2$, is itself ME distributed with representation $\mathrm{ME}_{p_1+p_2}(\boldsymbol{\alpha}, \mathbf{S})$ given by:

$$\boldsymbol{\alpha} = (\pi_1 \boldsymbol{\alpha}_1, \pi_2 \boldsymbol{\alpha}_2) \text{ and } \mathbf{S} = \begin{bmatrix} \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2 \end{bmatrix}.$$

A valid multi-exits matrix $\mathbf{T}$ associated with $\mathbf{S}$ is

$$\mathbf{T} = \begin{bmatrix} -\mathbf{S}_1 \mathbf{e} & \mathbf{0} \\ \mathbf{0} & -\mathbf{S}_2 \mathbf{e} \end{bmatrix}.$$

We have $(W, G) \sim \mathrm{MEME}_{p_1+p_2,2}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$, where $g \in \{1, 2\}$. That is the probabilistic mixture of two ME representations is MEME distributed with representation $\mathrm{MEME}_{p_1+p_2,2}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$, such that the exit corresponds to the chosen ME distribution of the mixture.

- Minimum

  The random variable $W = \min(W_1, W_2)$ is itself ME distributed with representation $\mathrm{ME}_{p_1 p_2}(\boldsymbol{\alpha}, \mathbf{S})$ given by:

  $$\boldsymbol{\alpha} = \boldsymbol{\alpha}_1 \otimes \boldsymbol{\alpha}_2 \text{ and } \mathbf{S} = \mathbf{S}_1 \otimes \mathbf{I}_{p_2} + \mathbf{I}_{p_1} \otimes \mathbf{S}_2 = \mathbf{S}_1 \oplus \mathbf{S}_2.$$

  A valid multi-exits matrix $\mathbf{T}$ associated with $\mathbf{S}$ is

  $$\mathbf{T} = \begin{bmatrix} (-\mathbf{S}_1 \mathbf{e}) \otimes \mathbf{I}_{p_2} & \mathbf{I}_{p_1} \otimes (-\mathbf{S}_2 \mathbf{e}) \end{bmatrix}.$$

  We have $(W, G) \sim \mathrm{MEME}_{p_1 p_2, 2}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$, where $g \in \{1, 2\}$. That is the minimum of two ME representations is MEME distributed with representation $\mathrm{MEME}_{p_1 p_2, 2}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$, such that the exit corresponds to which of the two variables that was the smallest. The event that both variables attain the same value has probability 0, and does not need special consideration.

- Maximum

  The random variable $W = \max(W_1, W_2)$ is itself ME distributed with representation $\mathrm{ME}_{p_1 p_2 + p_1 + p_2}(\boldsymbol{\alpha}, \mathbf{S})$ given by:

  $$\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1 \otimes \boldsymbol{\alpha}_2, \mathbf{0}, \mathbf{0}) \text{ and } \mathbf{S} = \begin{bmatrix} \mathbf{S}_1 \oplus \mathbf{S}_2 & \mathbf{I}_{p_1} \otimes (-\mathbf{S}_2 \mathbf{e}) & (-\mathbf{S}_1 \mathbf{e}) \otimes \mathbf{I}_{p_2} \\ \mathbf{0} & \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_2 \end{bmatrix}.$$

  A valid multi-exits matrix $\mathbf{T}$ associated with $\mathbf{S}$ is

  $$\mathbf{T} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -\mathbf{S}_1 \mathbf{e} & \mathbf{0} \\ \mathbf{0} & -\mathbf{S}_2 \mathbf{e} \end{bmatrix}.$$

  We have $(W, G) \sim \mathrm{MEME}_{p_1 p_2 + p_1 + p_2, 2}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$, where $g \in \{1, 2\}$. That is the maximum of two ME representations is MEME distributed with representation $\mathrm{MEME}_{p_1 p_2 + p_1 + p_2, 2}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$, such that the exit corresponds to which variable attained the maximum.

5.3.2 Compositionality

Like CPH and ME distributions, MEME distributions are closed under a number of compositional operations. Here, we study the compositionality considering MEME distributions as basic building blocks. For simplicity, we show our desired construction for each operator involving two MEME distributions, but the extension to a general number is straightforward by associativity.

Minimum/Race The minimum of two MEME distributions with representations $\text{MEME}_{p_1,n_1}(\boldsymbol{\alpha}_1, \mathbf{S}_1, \mathbf{T}_1)$ and $\text{MEME}_{p_2,n_2}(\boldsymbol{\alpha}_2, \mathbf{S}_2, \mathbf{T}_2)$ respectively is given by MEME $_{p_1 p_2, n_1 p_2 + p_1 n_2}(\bar{\boldsymbol{\alpha}}, \bar{\mathbf{S}}, \bar{\mathbf{T}})$, such that $\bar{\boldsymbol{\alpha}} = (\boldsymbol{\alpha}_1 \otimes \boldsymbol{\alpha}_2)$,

$$\bar{\mathbf{S}} = \mathbf{S}_1 \oplus \mathbf{S}_2, \quad \bar{\mathbf{T}} = \begin{bmatrix} \mathbf{T}_1 \otimes \mathbf{I}_{p_2} & \mathbf{I}_{p_1} \otimes \mathbf{T}_2 \end{bmatrix}. \tag{5.4}$$

The matrix block $\mathbf{T}_1 \otimes \mathbf{I}_{p_2}$ corresponds to the case where the random variable W_1 given by the $\text{MEME}_{p_1,n_1}(\boldsymbol{\alpha}_1, \mathbf{S}_1, \mathbf{T}_1)$ is smaller than the random variable W_2 given by the $\text{MEME}_{p_2,n_2}(\boldsymbol{\alpha}_2, \mathbf{S}_2, \mathbf{T}_2)$. The matrix block $\mathbf{I}_{p_1} \otimes \mathbf{T}_2$ corresponds to the case where the W_2 is smaller than the W_1.

Maximum/Synchronization The maximum of two MEME distributions with representations $\text{MEME}_{p_1,n_1}(\boldsymbol{\alpha}_1, \mathbf{S}_1, \mathbf{T}_1)$ and $\text{MEME}_{p_2,n_2}(\boldsymbol{\alpha}_2, \mathbf{S}_2, \mathbf{T}_2)$ respectively is given by $\text{MEME}_{p_1 p_2 + p_1 + p_2, n_1 + n_2}(\bar{\boldsymbol{\alpha}}, \bar{\mathbf{S}}, \bar{\mathbf{T}})$, such that $\bar{\boldsymbol{\alpha}} = (\boldsymbol{\alpha}_1 \otimes \boldsymbol{\alpha}_2, \mathbf{0}, \mathbf{0})$,

$$\bar{\mathbf{S}} = \begin{bmatrix} \mathbf{S}_1 \oplus \mathbf{S}_2 & \mathbf{I}_{p_1} \otimes (\mathbf{T}_2 \mathbf{e}) & (\mathbf{T}_1 \mathbf{e}) \otimes \mathbf{I}_{p_2} \\ \mathbf{0} & \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_2 \end{bmatrix}, \quad \bar{\mathbf{T}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 \end{bmatrix}. \tag{5.5}$$

The first column of $\bar{\mathbf{T}}$ corresponds to the case where the random variable W_1 given by the $\text{MEME}_{p_1,n_1}(\boldsymbol{\alpha}_1, \mathbf{S}_1, \mathbf{T}_1)$ is bigger than the random variable W_2 given by the $\text{MEME}_{p_2,n_2}(\boldsymbol{\alpha}_2, \mathbf{S}_2, \mathbf{T}_2)$. The second column of $\bar{\mathbf{T}}$ corresponds to the case where the W_2 is bigger than the W_1.

Probabilistic Mixture/Execution Let $\boldsymbol{\pi} = (\pi_1, \pi_2)$ be a valid probability distribution, the probabilistic mixture of two MEME distributions with repre-

sentations $\text{MEME}_{p_1,n_1}(\boldsymbol{\alpha}_1, \mathbf{S}_1, \mathbf{T}_1)$ and $\text{MEME}_{p_2,n_2}(\boldsymbol{\alpha}_2, \mathbf{S}_2, \mathbf{T}_2)$ respectively is given by $\text{MEME}_{p_1+p_2,n_1+n_2}(\bar{\boldsymbol{\alpha}}, \bar{\mathbf{S}}, \bar{\mathbf{T}})$, such that $\bar{\boldsymbol{\alpha}} = (\pi_1\boldsymbol{\alpha}_1, \pi_2\boldsymbol{\alpha}_2)$,

$$\bar{\mathbf{S}} = \begin{bmatrix} \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2 \end{bmatrix}, \quad \bar{\mathbf{T}} = \begin{bmatrix} \mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 \end{bmatrix}. \tag{5.6}$$

The first column of $\bar{\mathbf{T}}$ corresponds the distribution $\text{MEME}_{p_1,n_1}(\boldsymbol{\alpha}_1, \mathbf{S}_1, \mathbf{T}_1)$ is chosen from the mixture, and the second column corresponds the distribution $\text{MEME}_{p_2,n_2}(\boldsymbol{\alpha}_2, \mathbf{S}_2, \mathbf{T}_2)$ is chosen from the mixture.

Formulae (5.4 - 5.6) serve mathematical constructions for each compositional operator, such that one can compose a large and complicated distribution using simple MEME distributions as components. The proofs for the above results follow the same pattern as the compositionality results for PH distributions [Neu75], so we omit the details.

5.3.3   Order minimisation

Apart from compositionality, the most appealing feature of MEME distributions is minimality, i.e. for any MEME distribution one shall always find a representation with a minimal order. This minimality reduces the state space explosion problem. Using minimality, one can replace each MEME representation with its smallest equivalent counterpart. In this manner, we ensure that the capacity of handling huge system reaches the maximum.

DEFINITION 5.3 (***A Minimal MEME Representation***) A representation $\text{MEME}_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ of a MEME distribution is called minimal if no representation $\text{MEME}_{\bar{p},n}(\bar{\boldsymbol{\alpha}}, \bar{\mathbf{S}}, \bar{\mathbf{T}})$ of order $\bar{p} < p$ exist.

For a $\text{MEME}_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ representation, we define the initial vector space $\boldsymbol{L}_p = span\{\boldsymbol{\alpha}, \boldsymbol{\alpha}\mathbf{S}, \dots, \boldsymbol{\alpha}\mathbf{S}^{p-1}\}$. From the Cayley-Hamilton theorem [AM69], we know that $\mathbf{S}^p$ can be expressed as a linear combination of the lower power terms $\mathbf{S}^i$ with $i = \{0, 1, \dots, p-1\}$. Therefore, the initial vector space is sufficiently defined by the terms $\boldsymbol{\alpha}\mathbf{S}^i$ up to $i = p - 1$. The standard minimisation results of ME distributions from [AB99] are partially compatible to MEME distributions, as the analysis of the initial vector space $\boldsymbol{L}_p$ still applies for MEME distributions.

THEOREM 5.4 *For a $MEME_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ representation, let $d_l = dim(\boldsymbol{L}_p) <$*

p and $\mathbf{l}_1, \ldots, \mathbf{l}_{d_l}$ be a basis for $\boldsymbol{L}_p$. Define a $d_l \times p$ matrix $\mathbf{A} = \begin{bmatrix} \mathbf{l}_1 \\ \vdots \\ \mathbf{l}_{d_l} \end{bmatrix}$ and a

$p \times d_l$ matrix $\mathbf{B}$ such that $\mathbf{l}_i \mathbf{B} = \mathbf{e}_i^{tr}$ for $i = 1, \ldots, d_l$. Then, the representation $MEME_{d_l,n}(\boldsymbol{\alpha}\mathbf{B}, \mathbf{ASB}, \mathbf{AT})$ is a smaller representation of the $MEME_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ distribution.

Proof. First, we prove that $\forall i \in \{0, \ldots, p-1\}$: $\boldsymbol{\alpha}\mathbf{S}^i\mathbf{BA} = \boldsymbol{\alpha}\mathbf{S}^i$. Since $\boldsymbol{\alpha}\mathbf{S}^i \in \boldsymbol{L}_p$, we have

$$\boldsymbol{\alpha}\mathbf{S}^i\mathbf{BA} = \left(\sum_{j=1}^{d_l} \theta_j \mathbf{l}_j \right) \mathbf{BA} = \left(\sum_{j=1}^{d_l} \theta_j \mathbf{e}_j^{tr} \right) \begin{bmatrix} \mathbf{l}_1 \\ \vdots \\ \mathbf{l}_{d_l} \end{bmatrix} = \sum_{j=1}^{d_l} \theta_j \mathbf{l}_j = \boldsymbol{\alpha}\mathbf{S}^i.$$

We now show that $\mathrm{MEME}_{d_l,n}(\boldsymbol{\alpha}\mathbf{B}, \mathbf{ASB}, \mathbf{AT})$ and $\mathrm{MEME}_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ are representations of the same distribution. For all $\mathbf{T}_g$ where $g \in \{1, \ldots, n\}$, we have the joint density function

$$f(w, g) = \boldsymbol{\alpha}\mathbf{B}e^{\mathbf{ASB}w}\mathbf{AT}_g = \boldsymbol{\alpha}\mathbf{B} \sum_{i=0}^{\infty} \frac{w^i}{i!} (\mathbf{ASB})^i \mathbf{AT}_g = \sum_{i=0}^{\infty} \frac{w^i}{i!} \underbrace{\boldsymbol{\alpha}\mathbf{BA}}_{\boldsymbol{\alpha}}(\mathbf{SBA})^i\mathbf{T}_g$$

$$= \sum_{i=0}^{\infty} \frac{w^i}{i!} \underbrace{\boldsymbol{\alpha}\mathbf{SBA}}_{\boldsymbol{\alpha}\mathbf{S}}(\mathbf{SBA})^{i-1}\mathbf{T}_g = \cdots = \sum_{i=0}^{\infty} \frac{w^i}{i!} \boldsymbol{\alpha}\mathbf{S}^i\mathbf{T}_g = \boldsymbol{\alpha}e^{\mathbf{S}w}\mathbf{T}_g,$$

which is the joint density of the $\mathrm{MEME}_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ representation. $\square$

The standard treatments from [AB99] for the redundancy according to the closing (right) vector space is no longer valid for the multi-exits case. Instead, we generalise their results to handle the multi-exits matrix. For a $\mathrm{MEME}_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$, the matrix exponential part of the joint density function can be expanded into a power series, i.e. $f(w, g) = \boldsymbol{\alpha}e^{\mathbf{S}w}\mathbf{T}_g = \boldsymbol{\alpha} \sum_{i=0}^{\infty} \frac{w^i}{i!}\mathbf{S}^i\mathbf{T}_g$. For other MEME representations to be the same distribution, we shall require that all elements of $\{\mathbf{S}^i\mathbf{T}_g : \forall i \in [0, \mathrm{p}\text{-}1], \forall g \in [1, n]\}$ are the same (the Cayley-Hamilton theorem ensures that the terms $\mathbf{S}^i\mathbf{T}_g$ up to $i = p - 1$ are sufficient). Henceforth, we define the closing (right) vector space $\boldsymbol{R}_p$ as

$$\boldsymbol{R}_p = span\{\underbrace{\mathbf{T}_1, \ldots, \mathbf{T}_n}_{\mathbf{T}}, \underbrace{\mathbf{ST}_1, \ldots, \mathbf{ST}_n}_{\mathbf{ST}}, \ldots, \underbrace{\mathbf{S}^{p-1}\mathbf{T}_1, \ldots, \mathbf{S}^{p-1}\mathbf{T}_n}_{\mathbf{S}^{p-1}\mathbf{T}}\}.$$

Now, we shall remove the redundancy from $\boldsymbol{R}_p$ using Singular Value Decomposition (SVD) [Str88] inspired from [BT11a]. Since we only deal with real

numbers, the conjugate transpose of a matrix in standard SVD becomes just transpose in our case. Henceforth, an arbitrary matrix $\mathbf{A}$ of dimensions $p \times n$ (assume $p \geq n$, in the other case just apply $\mathbf{A}^{tr}$) and rank r can be factorized in the form

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{tr} = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_p \end{bmatrix} \begin{bmatrix} \iota_1 & 0 & \cdots & 0 \\ 0 & \iota_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \iota_n \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^{tr} \\ \vdots \\ \mathbf{v}_n^{tr} \end{bmatrix},$$

such that $\mathbf{u}_1, \ldots, \mathbf{u}_r$ are all orthonormal basis vectors in the column space of $\mathbf{A}$, $\mathbf{u}_{r+1}, \ldots, \mathbf{u}_p$ are all orthonormal basis vectors in the null column space of $\mathbf{A}^{tr}$, $\mathbf{v}_1, \ldots, \mathbf{v}_r$ are all orthonormal basis vectors in the row space of $\mathbf{A}$, and $\mathbf{v}_{r+1}, \ldots, \mathbf{v}_n$ are all orthonormal basis vectors in the null row space of $\mathbf{A}$. Henceforth, $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, thus $\mathbf{U}^{-1} = \mathbf{U}^{tr}$ and $\mathbf{V}^{-1} = \mathbf{V}^{tr}$. The diagonal matrix $\boldsymbol{\Sigma}$ has nonnegative decreasing real numbers on the diagonal, such that $\iota_1 \geq \cdots \geq \iota_r > \iota_{r+1} = \cdots = \iota_n = 0$. The computation time complexity of $\mathsf{SVD}(\mathbf{A})$ is given as $\mathcal{O}(pn^2)$ [TB97]. We shall use the SVD of a matrix to complete the following theorem.

THEOREM 5.5 *For a $MEME_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ representation, if $dim(\boldsymbol{R}_p) = d_r < p$, then there exists a $p \times p$ non-singular transformation matrix $\mathbf{B}$ such that*

$$\boldsymbol{\alpha}\mathbf{B} = \begin{bmatrix} \boldsymbol{\alpha}' & \mathbf{0} \end{bmatrix}, \ \mathbf{B}^{-1}\mathbf{S}\mathbf{B} = \begin{bmatrix} \mathbf{S}' & \star \\ \star & \star \end{bmatrix}, \mathbf{B}^{-1}\mathbf{T} = \begin{bmatrix} \mathbf{T}' \\ \mathbf{0} \end{bmatrix}, \ and \ \mathbf{B}^{-1}\mathbf{e} = \begin{bmatrix} \mathbf{e}' \\ \star \end{bmatrix},$$

where a $\star$ represents an irrelevant matrix block, $\boldsymbol{\alpha}'$ is a $1 \times d_r$ row vector, $\mathbf{S}'$ is a $d_r \times d_r$ matrix, $\mathbf{T}'$ is a $d_r \times n$ matrix and $\mathbf{e}'$ is a $d_r \times 1$ column vector of ones. The $MEME_{d_r,n}(\boldsymbol{\alpha}', \mathbf{S}', \mathbf{T}')$ representation is a smaller representation of the $MEME_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ distribution.

Proof. We first show that $\mathrm{MEME}_{d_r,n}(\boldsymbol{\alpha}', \mathbf{S}', \mathbf{T}')$ represents the same distribution as $\mathrm{MEME}_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ assuming the matrix $\mathbf{B}$ exists, and we show how to find $\mathbf{B}$ afterwards.

Notice that the $\mathbf{B}^{-1}\mathbf{T} = \begin{bmatrix} \mathbf{T}' \\ \mathbf{0} \end{bmatrix}$ implies $\mathbf{B}^{-1}\mathbf{T}_g = \begin{bmatrix} \mathbf{T}'_g \\ \mathbf{0} \end{bmatrix}, \forall g \in [1, n]$. Thus, we

have the joint density function

$$f(w,g) = \boldsymbol{\alpha} e^{\mathbf{S}w} \mathbf{T}_g = \boldsymbol{\alpha} \sum_{i=0}^{\infty} \frac{w^i}{i!} \mathbf{S}^i \mathbf{T}_g = \sum_{i=0}^{\infty} \frac{w^i}{i!} \boldsymbol{\alpha} \mathbf{B} \mathbf{B}^{-1} (\mathbf{S} \mathbf{B} \mathbf{B}^{-1})^i \mathbf{T}_g$$

$$= \sum_{i=0}^{\infty} \frac{w^i}{i!} \boldsymbol{\alpha} \mathbf{B} (\mathbf{B}^{-1} \mathbf{S} \mathbf{B})^i \mathbf{B}^{-1} \mathbf{T}_g = \sum_{i=0}^{\infty} \frac{w^i}{i!} \begin{bmatrix} \boldsymbol{\alpha}' & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{S}' & \star \\ \star & \star \end{bmatrix}^i \begin{bmatrix} \mathbf{T}'_g \\ \mathbf{0} \end{bmatrix}$$

$$= \sum_{i=0}^{\infty} \frac{w^i}{i!} \boldsymbol{\alpha}' \mathbf{S}'^i \mathbf{T}'_g = \boldsymbol{\alpha}' e^{\mathbf{S}'w} \mathbf{T}'_g = f'(w,g),$$

such that $f'(w,g)$ is the joint density function of $\text{MEME}_{d_r,n}(\boldsymbol{\alpha}', \mathbf{S}', \mathbf{T}')$.

Now we find the matrix $\mathbf{B}$ using SVD. Given $dim(\boldsymbol{R}_p) = d_r < p$, we represent the closing vector space $\boldsymbol{R}_p$ with a matrix $\mathbf{R}_p$. We then have $\mathbf{R}_p = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{tr}$. Using $\mathbf{U}^{-1} = \mathbf{U}^{tr}$, we have

$$\mathbf{U}^{tr}\mathbf{R}_p = \boldsymbol{\Sigma}\mathbf{V}^{tr} = \begin{bmatrix} \iota_1 & 0 & \cdots & 0 \\ 0 & \ddots & \cdots & 0 \\ \vdots & \vdots & \iota_{d_r} & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^{tr} \\ \vdots \\ \mathbf{v}_n^{tr} \end{bmatrix} = \begin{bmatrix} \iota_1 \mathbf{v}_1^{tr} \\ \vdots \\ \iota_{d_r} \mathbf{v}_{d_r}^{tr} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \star \\ \mathbf{0} \end{bmatrix},$$

where the $\star$ is of dimensions $d_r \times n$ representing potential non-zero entries. For all matrices $\mathbf{A} \subseteq \boldsymbol{R}_p$, we have $rank(\mathbf{A}) \leq rank(\mathbf{R}_p) = d_r$, that is $\mathbf{U}^{tr}\mathbf{A} = \begin{bmatrix} \star \\ \mathbf{0} \end{bmatrix}$. Therefore, we have $\underbrace{\mathbf{U}^{tr}\mathbf{T}} = \begin{bmatrix} \star \\ \mathbf{0} \end{bmatrix}$ and $\mathbf{U}^{tr}\mathbf{ST} = \begin{bmatrix} \star \\ \mathbf{0} \end{bmatrix}$ because $\mathbf{T} \subseteq \boldsymbol{R}_p$ and $\mathbf{ST} \subseteq \boldsymbol{R}_p$.

To find a structure containing $\mathbf{U}^{tr}\mathbf{SU}$ from $\mathbf{U}^{tr}\mathbf{ST}$, we insert $\mathbf{I} = \mathbf{U}\mathbf{U}^{tr}$:

$$\mathbf{U}^{tr}\mathbf{ST} = \underbrace{\mathbf{U}^{tr}\mathbf{SU}}\,\mathbf{U}^{tr}\mathbf{T} = \underbrace{\mathbf{U}^{tr}\mathbf{SU}} \begin{bmatrix} \star \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \star \\ \mathbf{0} \end{bmatrix}.$$

We observe that $\underbrace{\mathbf{U}^{tr}\mathbf{SU}}$ has the structure $\begin{bmatrix} \mathbf{S}' & \star \\ \star & \star \end{bmatrix}$, where the $\mathbf{S}'$ block has dimensions of $d_r \times d_r$. From the known structure $\mathbf{U}^{tr}\mathbf{ST}$, we find a structure containing $\boldsymbol{\alpha}$ as

$$\boldsymbol{\alpha}\mathbf{ST} = \underbrace{\boldsymbol{\alpha}\mathbf{U}}\,\mathbf{U}^{tr}\mathbf{ST} = \underbrace{\boldsymbol{\alpha}\mathbf{U}} \begin{bmatrix} \star \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}\mathbf{U}\star & \mathbf{0} \end{bmatrix},$$

where the $\boldsymbol{\alpha}\mathbf{U}\star$ is of dimension $1 \times d_r$.

Henceforth, the matrix $\mathbf{U}$ from $\mathsf{SVD}(\mathbf{R}_\mathsf{p})$ can be used as our desired $\mathbf{B}$ matrix in the theorem. However, to ensure the reduced representation having

$\boldsymbol{\alpha}'\mathbf{e} = 1$ requires an extra condition. From $\boldsymbol{\alpha}\mathbf{e} = 1$, we have that $\boldsymbol{\alpha}\mathbf{B}\mathbf{B}^{-1}\mathbf{e} = \begin{bmatrix} \boldsymbol{\alpha}' & \mathbf{0} \end{bmatrix} \mathbf{B}^{-1}\mathbf{e} = 1$. A necessary and sufficient condition is $\mathbf{B}^{-1}\mathbf{e} = \begin{bmatrix} \mathbf{e}' \\ \star \end{bmatrix}$, such that the condition is satisfied by normalising the $\mathbf{U}$ matrix.

Let $(\mathbf{U}^{tr}\mathbf{e})_i$ denote the i-th element of the vector $\mathbf{U}^{tr}\mathbf{e}$, and $diag\{\mathbf{U}^{tr}\mathbf{e}\}$ denote the diagonal matrix having diagonal entries from the vector $\mathbf{U}^{tr}\mathbf{e}$, such that all the 0 diagonal elements are substituted by 1 to allow later matrix inversion. The normalised matrix $\mathbf{B}$ is defined as $\mathbf{B} = \mathbf{U}diag\{\mathbf{U}^{tr}\mathbf{e}\}$, such that $\mathbf{B}^{-1}\mathbf{e} = (diag\{\mathbf{U}^{tr}\mathbf{e}\})^{-1}\mathbf{U}^{-1}\mathbf{e} = \begin{bmatrix} (\mathbf{U}^{tr}\mathbf{e})_1^{-1} & 0 & \cdots \\ 0 & (\mathbf{U}^{tr}\mathbf{e})_2^{-1} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} (\mathbf{U}^{tr}\mathbf{e})_1 \\ (\mathbf{U}^{tr}\mathbf{e})_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{e}' \\ \star \end{bmatrix}$. Thus, the normalised $\mathbf{B}$ matrix normalised ensure $\boldsymbol{\alpha}'\mathbf{e} = 1$ in the reduced representation.

Henceforth, we have found the $\mathbf{B}$ matrix in the theorem. By $\mathbf{R}_p = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{tr}$, $\mathbf{B} = \mathbf{U}diag\{\mathbf{U}^{tr}\mathbf{e}\}$, and

$$\boldsymbol{\alpha}\mathbf{B} = \begin{bmatrix} \boldsymbol{\alpha}' & \mathbf{0} \end{bmatrix}, \ \mathbf{B}^{-1}\mathbf{S}\mathbf{B} = \begin{bmatrix} \mathbf{S}' & \star \\ \star & \star \end{bmatrix}, \ \text{and } \mathbf{B}^{-1}\mathbf{T} = \begin{bmatrix} \mathbf{T}' \\ \mathbf{0} \end{bmatrix},$$

a smaller $\text{MEME}_{d_r,n}$ $(\boldsymbol{\alpha}', \mathbf{S}', \mathbf{T}')$ representation is obtained. $\square$

THEOREM 5.6 *A MEME representation $MEME_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ is minimal if and only if $dim(\boldsymbol{L}_p) = p$ and $dim(\boldsymbol{R}_p) = p$.*

Proof. We first prove that a minimal representation shall have $dim(\boldsymbol{L}_p) = p$ and $dim(\boldsymbol{R}_p) = p$. For an arbitrary representation, a smaller representation of order $dim(\boldsymbol{L}_p)$ shall be found following Theorem 5.4, and a smaller representation of order $dim(\boldsymbol{R}_p)$ shall be found following Theorem 5.5. That's the minimal representation shall have its order equals $dim(\boldsymbol{L}_p)$ and $dim(\boldsymbol{R}_p)$.

Now we show that if $dim(\boldsymbol{L}_p) = dim(\boldsymbol{R}_p) = p$, then $\text{MEME}_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ is minimal by contradiction. Assume $dim(\boldsymbol{L}_p) = dim(\boldsymbol{R}_p) = p$ and there exists a smaller representation of the same distribution of dimension $p' < p$, then either $dim(\boldsymbol{L}_p)$ or $dim(\boldsymbol{R}_p)$ of $\text{MEME}_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ shall be p'. The contradiction appears as needed. $\square$

COROLLARY 5.7 *Let p_m be the order of a minimal representation of $MEME_{p,n}$ $(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$, then $p_m = \min(dim(\boldsymbol{L}_p), dim(\boldsymbol{R}_p)) \leq p$.*

Proof. This corollary summarises the results from Theorem 5.4, Theorem 5.5 and Theorem 5.6, because only when $p_m = \min(dim(\boldsymbol{L}_p), dim(\boldsymbol{R}_p))$ it is not possible to minimise by Theorem 5.4 and Theorem 5.5, and meanwhile the minimality condition of Theorem 5.6 holds. $\qquad\square$

To sum up, let p be the order of a MEME representation, Theorem 5.6 states a necessary and sufficient condition for the minimality of a MEME representation, that's both $dim(\boldsymbol{L}_p)$ and $dim(\boldsymbol{R}_p)$ are equal to p. If $dim(\boldsymbol{L}_p) < dim(\boldsymbol{R}_p)$, the representation has the minimal order of $dim(\boldsymbol{L}_p)$ and shall be minimised according to Theorem 5.4. If $dim(\boldsymbol{R}_p) < dim(\boldsymbol{L}_p)$, the representation has the minimal order of $dim(\boldsymbol{R}_p)$ and shall be minimised according to Theorem 5.5. We formulate the routine into an algorithm using the MATLAB language to compute a minimal representation of MEME distributions.

In Algorithm 2, the function rref($\mathbf{L}_p$) in MATLAB produces a reduced row Echelon form of matrix $\mathbf{L}_p$, such that vector jb contains the indexes of linear independent columns. For a $\text{MEME}_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ representation, the matrix $\mathbf{L}_p$ is of dimensions $p \times p$, therefore the time complexity of rref($\mathbf{L}_p$) is $\mathcal{O}(p^3)$. The matrix $\mathbf{R}_p$ is of dimensions $p \times pn$, therefore the time complexity of $\mathsf{SVD}(\mathbf{R}_\mathsf{p})$ is given as $\mathcal{O}(np^3)$. Henceforth, the worst case complexity of Algorithm 2 is given as $\mathcal{O}(np^3)$.

EXAMPLE 5.2 *We consider a* $MEME_{5,2}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ *representation, such that*

$$\boldsymbol{\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} 0.7500 & 1.0000 \\ 1.3333 & 1.3333 \\ 0.2000 & 0.2000 \\ 0.1667 & 0.1667 \\ 1.0000 & 1.0000 \end{bmatrix},$$

$$\mathbf{S} = \begin{bmatrix} -7.0000 & 5.2500 & 0 & 0 & 0 \\ 0 & -6.0000 & 3.3333 & 0 & 0 \\ 0 & 0 & -4.0000 & 3.6000 & 0 \\ 0 & 0 & 0 & -3.0000 & 2.6667 \\ 0 & 0 & 0 & 0 & -2.0000 \end{bmatrix}.$$

For this representation, we have $rank(\mathbf{R}_p) = 4$, *and* $rank(\mathbf{L}_p) = 5$ *such that Theorem 5.5 applies. By Algorithm 2, the* $\mathbf{B}$ *matrix is obtained as*

$$\mathbf{B} = \begin{bmatrix} 0.6644 & 0.2931 & 0.0407 & 0.0018 & 0.0000 \\ -0.1517 & 1.1187 & -0.0872 & 0.1202 & -0.3235 \\ -0.0633 & 0.3026 & 0.8420 & -0.0813 & 0.5392 \\ 0.0216 & -0.2683 & 0.5746 & 0.6721 & -0.6470 \\ -0.0022 & 0.0583 & -0.2561 & 1.2000 & 0.4313 \end{bmatrix}.$$

Algorithm 2 Minimise a MEME representation.

Require: $\text{MEME}_{p,n}(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$

1: $\mathbf{L}_p = \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}\mathbf{S} \\ \cdots \\ \boldsymbol{\alpha}\mathbf{S}^{p-1} \end{bmatrix}$, $\mathbf{R}_p = \begin{bmatrix} \mathbf{T} & \mathbf{ST} & \cdots & \mathbf{S}^{p-1}\mathbf{T} \end{bmatrix}$

2: $\text{rank}(\mathbf{L}_p) = d_l$, $\text{rank}(\mathbf{R}_p) = d_r$

3: **if** $d_l = d_r = p$ **then**

4: $\quad \boldsymbol{\alpha}_m = \boldsymbol{\alpha}, \mathbf{S}_m = \mathbf{S}, \mathbf{T}_m = \mathbf{T}$

5: $\quad$ **return** $\boldsymbol{\alpha}_m, \mathbf{S}_m, \mathbf{T}_m$

6: **end if**

7: **if** $d_l \leq d_r$ **then**

8: $\quad$ [x, jb] = $\text{rref}(\mathbf{L}_p)$;

9: $\quad \mathbf{A} = \mathbf{L}_p(jb, :)$;

10: $\quad \mathbf{B} = \mathbf{A}\backslash\mathbf{I}$;

11: $\quad \boldsymbol{\alpha}_m = \boldsymbol{\alpha}\mathbf{B}, \mathbf{S}_m = \mathbf{ASB}, \mathbf{T}_m = \mathbf{AT}$

12: **else**

13: $\quad [\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{V}] = \text{SVD}(\mathbf{R}_p)$

14: $\quad \mathbf{N} = diag(\mathbf{U}^{tr}\mathbf{e})$

15: $\quad$ **for** i = 1:p **do**

16: $\quad\quad$ **if** $\mathbf{N}$(i,i) equals 0 **then**

17: $\quad\quad\quad \mathbf{N}$(i,i) = 1

18: $\quad\quad$ **end if**

19: $\quad$ **end for**

20: $\quad \mathbf{B}_n = \mathbf{UN}$

21: $\quad \boldsymbol{\alpha}_m = \boldsymbol{\alpha}\mathbf{B}_n(:, 1 : d_r)$

22: $\quad \mathbf{S}_m = \mathbf{B}_n^{-1}\mathbf{SB}_n(1 : d_r, 1 : d_r)$

23: $\quad \mathbf{T}_m = \mathbf{B}_n^{-1}\mathbf{T}(1 : d_r, :)$

24: **end if**

25: **return** $\boldsymbol{\alpha}_m, \mathbf{S}_m, \mathbf{T}_m$

A minimal representation of order 4 is obtained as $\text{MEME}_{4,2}(\boldsymbol{\alpha}_m, \mathbf{S}_m, \mathbf{T}_m)$, where $\boldsymbol{\alpha}_m = \begin{bmatrix} 0.6644 & 0.2931 & 0.0407 & 0.0018 \end{bmatrix}$,

$$\mathbf{S}_m = \begin{bmatrix} -7.9921 & 7.5985 & -2.0681 & 0.8927 \\ -0.4620 & -4.1103 & 2.5193 & -0.3694 \\ -0.0411 & -0.5365 & -2.6293 & 3.3373 \\ 0.0026 & 0.0021 & -0.2605 & -1.2682 \end{bmatrix}, \mathbf{T}_m = \begin{bmatrix} 0.6074 & 0.9617 \\ 1.1869 & 1.2356 \\ -0.0698 & -0.0606 \\ 0.7619 & 0.7621 \end{bmatrix}.$$

COROLLARY 5.8 *A MEME representation composed from minimal MEME representations is not necessarily minimal.*

Proof.   A simple counterexample shall be a probabilistic mixture (cf. Formula 5.6) for two of the same minimal MEME representation, where the resulting representation is always not minimal.   □

5.4   Languages for Models and Properties

In this section, we develop a stochastic process algebra MEME to express stochastic systems having matrix–exponential distributed holding times and multiple exits. We start by defining the syntax of our language, then describe some well-formedness conditions to ensure that the defined processes have a semantics. We then transform our language into a so-called explicit form in order to obtain the Markov renewal process semantics, and such that the compositionality and minimality are ensured during our transformation. To express a variety of properties of the system, we adapt continuous stochastic logic for Markov renewal processes.

5.4.1   Syntax

To describe a Markov renewal process with labels and matrix–exponential distributed durations, we introduce a process algebra MEME.

DEFINITION 5.9 (MEME *Syntax*) The language MEME of Markov renewal processes of matrix–exponential kernels and labels consists of global systems *sys*, defined using initial probability distributions *ini*, system definitions *def*, general processes $\mathcal{Q}$, processes P, and annotations *ann* (where we assume that $m, t \in \mathbb{N}$, $\lambda \in \Re_{>0}$ and $\forall i \colon \pi_i \in [0,1]$):

$$
\begin{aligned}
sys &::= ini \mapsto def \\
ini &::= X_1 : \pi_1, \cdots, X_m : \pi_m \\
def &::= X_1\{ann_1\}[rew_1] := \mathcal{Q}_1; \cdots; X_m\{ann_m\}[rew_m] := \mathcal{Q}_m \\
\mathcal{Q} &::= 0 \mid P \\
P &::= [\![(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T}) \mapsto \overrightarrow{\mathbf{X}}]\!] \mid [\![f(\cdot) \mapsto \overrightarrow{\mathbf{X}}]\!] \mid \sum_{i \in I}^{\min} P_i \mid \sum_{i \in I}^{\max} P_i \mid \sum_{i \in I}^{\Pr} \pi_i P_i \\
&\quad \mid \lambda.X \mid \text{let } MEME(\overrightarrow{\mathbf{X}})_i = P_i \text{ in } P \mid MEME(\overrightarrow{\mathbf{X}})_i \\
ann &::= a_1, \cdots, a_t
\end{aligned}
$$

A global system *sys* is defined by an initial probability distribution together with system definitions, i.e. $ini \mapsto def$. Here *ini* assigns initial probabilities to indicate which process will start with what probability; we shall impose well-formedness conditions to ensure that the listed probabilities add up to one.

A system definition *def* denotes a set of m recursively defined processes of the form, $X_1\{ann_1\}[rew_1] := \mathcal{Q}_1; \cdots; X_m\{ann_m\}[rew_m] := \mathcal{Q}_m$; the well-formedness conditions imposed below will ensure that all process names $X_1, \ldots, X_m$ are pairwise distinct. For each process we shall use *ann* to describe a set of *atomic propositions* taken from a finite and non-empty set AP of atomic propositions; duplicates are allowed but have no effect. We leave the field blank for the empty *ann*.

A general process $\mathcal{Q}$ denotes either a terminating process described by the terminal symbol $\mathbf{0}$, or an normal process P. A process P consists of distribution definitions and multiary compositional operators. We use $\llparenthesis\ \rrparenthesis$ to express distribution definitions in order to stress that a MEME distribution shall be treated as an atomic action.

- The phrases $\llparenthesis(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T}) \mapsto \overrightarrow{\mathbf{X}}\rrparenthesis$ and $\llparenthesis f(\cdot) \mapsto \overrightarrow{\mathbf{X}}\rrparenthesis$ are dedicated to specify a multi-exits matrix–exponential distribution using matrix representation $(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ and density function $f(\cdot)$, respectively. The notation $f(\cdot)$ stands for a collection of known MEME density functions perhaps taken from a library; we shall focus on the matrix representation to illustrate further results. In our well-formedness conditions we shall impose that the matrix representation and density function need to be valid according to Definition 5.2.

  CPH distributions constitute a subclass of MEl distributions and are very useful in stochastic model checking. The phrase $\llparenthesis(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T}) \mapsto \overrightarrow{\mathbf{X}}\rrparenthesis$ gives rise to valid Multi-Exits CPH (MECPH) distributions if we impose some extra conditions.

  The multiple exits from a MEME distribution map to an ordered list of continuation processes $\overrightarrow{\mathbf{X}}$, such that the i-th column of multi-exits matrix $\mathbf{T}$ links to the i-th continuation process of $\overrightarrow{\mathbf{X}}$; we shall impose well-formedness conditions to ensure that all the continuation processes of $\overrightarrow{\mathbf{X}}$ are defined, and the size of $\overrightarrow{\mathbf{X}}$ equals the number of columns of matrix $\mathbf{T}$.

- There are three multiary operators, $\sum_{i \in I}^{\min}$, $\sum_{i \in I}^{\max}$, and $\sum_{i \in I}^{\mathrm{Pr}} \pi_i$, that specify a composite process corresponding to the compositional operators mentioned in Section 5.3.2, such that each participating process can again be specified using multiary operators. This allows us to nest multiary operators for expressing more complicated compositional constructions.

A process $\sum_{i \in I}^{\min} P_i$ describes a *race* between all participating processes, such that the fastest process wins the race and its multi-exits correspond to the continuation. The race is defined as the minimum of MEME distributions following Rule (5.4). A process $\sum_{i \in I}^{\max} P_i$ describes a *synchronisation* between all participating processes such that the multi-exits of the slowest process correspond to the continuation. The synchronisation is defined as the maximum of MEME distributions following Rule (5.5). A process $\sum_{i \in I}^{\Pr} \pi_i P_i$ presents a *probabilistic execution* of all participating processes, such that one of candidate processes is chosen probabilistically and its multi-exits correspond to the continuation. The probabilistic execution is defined as the probabilistic mixture of MEME distributions following Rule (5.6), where we require $\sum_{i \in I}^{\Pr} \pi_i = 1$.

Without changing expressiveness, the phrase $\lambda.X$ in a process definition can be used as a shorthand for specifying a CTMC, such that the $\lambda > 0$ is the parameter of an EXP distribution. This is purely syntactic sugar, since $\lambda.X = \llbracket ([1], [-\lambda], [\lambda]) \mapsto X \rrbracket$. As another shorthand for writing MEME distributions, the phrases let $MEME(\overrightarrow{\mathbf{X}})_i : P_i$ in P and $MEME(\overrightarrow{\mathbf{X}})_i$ allow us to express any subprocess P_i as $MEME(\overrightarrow{\mathbf{X}})_i$, such that the corresponding distribution definition comes afterwards. E.g., we can write that $P := MEME(X_1, X_3, X_4)_1 +_{\min}$ $\cdots$ and $MEME(X_1, X_3, X_4)_1 := \llbracket (\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T}) \mapsto (X_1, X_3, X_4) \rrbracket$.

5.4.2 Well-formedness conditions

In the following we shall write $\mathsf{FN}(\cdots)$ for the process names occurring in non-defining positions in the construct, and $\mathsf{DN}(\cdots)$ for the defined process names of the construct. The definitions are standard so we omit the details; as an example we note that $\mathsf{DN}(X_1\{ann_1\}[rew_1] := \mathcal{Q}_1; \cdots; X_m\{ann_m\}[rew_m] := \mathcal{Q}_m) = \{X_1, \cdots, X_m\}$. For each syntactic category we have a well-formedness judgement; the details are given in Table 5.2, and are explained below.

The clause for a global system, $\vdash_{\text{sys}} ini \mapsto def$, ensures that all process variables used in *ini* and *def* are defined. The clause for an initial probability distribution, $\vdash_{\text{ini}}$, ensures that all initial state names are pairwise distinct and the listed probabilities add up to 1. The clause for system definitions, $\vdash_{\text{def}} \cdots; X_i\{ann_i\}[rew_i] := P_i; \cdots$, ensures that all the process names are pairwise distinct, and all terminating processes shall have no rewards.

The clause for process definition of MEME distribution in matrix representation, $X_1 \cdots X_m \vdash_P \llbracket (\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T}) \mapsto \overrightarrow{\mathbf{X}} \rrbracket$, requires that the joint density function $f(w, g)$ from representation $(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})$ is valid. That is all the elements of $\boldsymbol{\alpha}$

$$\frac{\vdash_{\text{ini}} ini \quad \vdash_{\text{def}} def}{\vdash_{\text{sys}} ini \mapsto def} \quad \text{if } (\mathsf{FN}(def) \cup \mathsf{FN}(ini)) \subseteq \mathsf{DN}(def);$$

$$\vdash_{\text{ini}} X_1 : \pi_1, \cdots X_m : \pi_m \quad \text{if } \begin{cases} X_i \neq X_j \text{ when } i \neq j; \\ \Sigma_{i=1}^m \pi_i = 1. \end{cases}$$

$$\frac{X_1 \cdots X_m \vdash_P P_i \quad \vdash_{\text{rew}} rew_i}{\vdash_{\text{def}} \cdots ; X_i\{ann_i\}[rew_i] := \mathcal{Q}_i; \cdots} \quad \text{if } X_i \neq X_j \text{ when } i \neq j.$$

$$\frac{}{X_1 \cdots X_m \vdash_P \llbracket(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T}) \mapsto \overrightarrow{\mathbf{X}}\rrbracket} \quad \text{if } \begin{cases} \sum_i \alpha_i = 1; \\ \text{let } \mathbf{T}_g \text{ be the } g\text{-th column of } \mathbf{T}, \\ \quad \forall w, g : f(w, g) = \boldsymbol{\alpha} e^{\mathbf{S}w} \mathbf{T}_g \geq 0, \text{ and} \\ \quad \int_0^\infty \sum_g f(w, g) dw = 1; \\ \forall X \in \overrightarrow{\mathbf{X}} : X \in \{X_1, \cdots, X_m\}; \\ |\overrightarrow{\mathbf{X}}| = \text{the number of columns of } \mathbf{T}. \end{cases}$$

$$\frac{}{X_1 \cdots X_m \vdash_P \llbracket f(\cdot) \mapsto \overrightarrow{\mathbf{X}}\rrbracket} \quad \text{if } \begin{cases} f(\cdot) \geq 0 \text{ and } \int_0^\infty f(\cdot) = 1; \\ \forall X \in \overrightarrow{\mathbf{X}} : X \in \{X_1, \cdots, X_m\}; \\ \text{let } n = \text{the number of columns of } \mathbf{T}, \\ \quad \text{then } |\overrightarrow{\mathbf{X}}| = n. \end{cases}$$

$$\frac{}{X_1 \cdots X_m \vdash_P \sum_{i \in I}^{\text{Pr}} \pi_i P_i} \quad \text{if } \sum_{i \in I} \pi_i = 1.$$

Table 5.2: Well-formedness rules for language MEME.

sum to one, and for all w, g that $f(w, g)$ is non-negative and integrates to one. For the continuation processes $\overrightarrow{\mathbf{X}}$, we ensure that all continuation processes $\overrightarrow{\mathbf{X}}$ are defined. Let $|\overrightarrow{\mathbf{X}}|$ be the cardinality of $\overrightarrow{\mathbf{X}}$, we ensure that the number of continuation processes equals the number of exits.

The clause $X_1 \cdots X_m \vdash_P \llbracket(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T}) \mapsto \overrightarrow{\mathbf{X}}\rrbracket$ additionally specify a MECPH distribution if some side conditions are satisfied. We shall require additionally

all the elements of $\boldsymbol{\alpha}$, $\mathbf{S}$ and $\mathbf{T}$ are non-negative except the diagonal elements of $\mathbf{S}$, and the i-th row sum of $\mathbf{S}$ added to the i-th row sum of $\mathbf{T}$ equals 0 for all i. The clause $X_1 \cdots X_m \vdash_P \llbracket f(\cdot) \mapsto \overrightarrow{\mathbf{X}} \rrbracket$ ensures a MEME density function $f(\cdot)$ is valid, with the same requirements on continuation processes.

For the clause of multiary operator $\sum_{i \in I}^{\mathrm{Pr}} \pi_i$, we ensure that all π_is add up to one.

EXAMPLE 5.3 *We shall consider a Fault-Tolerant Machine (FTM) consisting of two independent components C_1 and C_2. The FTM starts with both components in good conditions, i.e.* ON, *however as time elapses the components shall eventually fail. For each component, the time until a failure corresponds to a life time distribution followed by turning the FTM into two types of damages, i.e. minor damage* minD *and major damage* maxD. *Once the FTM gets damaged, its self-repair mechanism activates automatically, such that the time for performing self-repair depends on the damage type and individual repair time distribution of the components. The self-repair is not guaranteed to be always successful, which instead is followed by several endings, e.g.* ON, maxD *or* OFF. *The FTM breaks down completely, i.e.* OFF, *once the failure is unrepairable.*
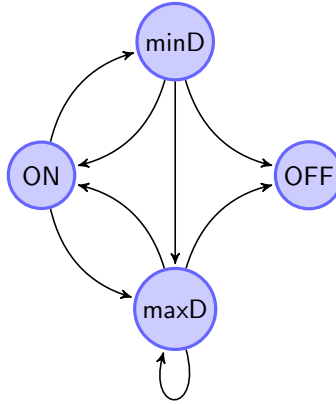
Figure 5.2: Graphical representation of a fault-tolerant machine.

Assume that, the component C_1 has a life time distribution $(\boldsymbol{\alpha}_1, \mathbf{S}_1, \mathbf{T}_1)$, while it has a repair time distribution $(\bar{\boldsymbol{\alpha}}_1, \bar{\mathbf{S}}_1, \bar{\mathbf{T}}_1)$ with the possibilities of turning the FTM to ON, maxD *or* OFF. *The component C_2 has a life time distribution $(\boldsymbol{\alpha}_2, \mathbf{S}_2, \mathbf{T}_2)$, while it has a repair time distribution $(\bar{\boldsymbol{\alpha}}_2, \bar{\mathbf{S}}_2, \bar{\mathbf{T}}_2)$ with possibilities of turning the FTM to* ON *or* OFF. *The FTM is illustrated graphically in Fig. 5.2; formally it consists of the definitions of Table 5.3 together with the initial probability distribution* ON : 1, minD : 0, maxD : 0, OFF : 0. *We explain*

the details below.

$$
\begin{aligned}
\mathsf{ON}\{\mathsf{normal}\} \quad &:= \pi_1 [\!((\boldsymbol{\alpha}_1, \mathbf{S}_1, \mathbf{T}_1) \mapsto (\mathsf{minD}, \mathsf{maxD}))\!] \\
&\quad +_{\mathrm{Pr}} \pi_2 [\!((\boldsymbol{\alpha}_2, \mathbf{S}_2, \mathbf{T}_2) \mapsto (\mathsf{minD}, \mathsf{maxD}))\!]; \\
\mathsf{minD}\{\mathsf{repair}\} \quad &:= [\!((\bar{\boldsymbol{\alpha}}_1, \bar{\mathbf{S}}_1, \bar{\mathbf{T}}_1) \mapsto (\mathsf{ON}, \mathsf{maxD}, \mathsf{OFF}))\!] \\
&\quad +_{\min} [\!((\bar{\boldsymbol{\alpha}}_2, \bar{\mathbf{S}}_2, \bar{\mathbf{T}}_2) \mapsto (\mathsf{ON}, \mathsf{OFF}))\!]; \\
\mathsf{maxD}\{\mathsf{repair}\} \quad &:= [\!((\bar{\boldsymbol{\alpha}}_1, \bar{\mathbf{S}}_1, \bar{\mathbf{T}}_1) \mapsto (\mathsf{ON}, \mathsf{maxD}, \mathsf{OFF}))\!] \\
&\quad +_{\max} [\!((\bar{\boldsymbol{\alpha}}_2, \bar{\mathbf{S}}_2, \bar{\mathbf{T}}_2) \mapsto (\mathsf{ON}, \mathsf{OFF}))\!]; \\
\mathsf{OFF}\{\mathsf{break}\} \quad &:= 0
\end{aligned}
$$

Table 5.3: A fault-tolerant machine in the MEME language.

The FTM always starts under good conditions. Depending on the external environment, either C_1 or C_2 will eventually fail according to a given probability distribution (π_1, π_2). Thus, the time for the FTM entering the damage states minD *or* maxD *is determined by a probabilistic mixture of the life time distributions of C_1 and C_2. When the FTM enters the state* minD, *the overall self-repair time and continuations are determined by the component which repairs faster. That corresponds the minimum of two repair time distributions. When the FTM enters the state* maxD, *the overall self-repair time and continuations are determined by the component which repairs slower. That corresponds the maximum of two repair time distributions. The unrepairable state* OFF *is simply a termination* 0.

There are labels to identify if the FTM is under normal, repair *or* break, *we shall discuss the FTM in more detail in subsequent examples.*

5.4.3   Aggregating Multi-exits

We have mentioned that a system may be composed using the multiary operators of the MEME language, such that the composition of MEME distributions follow the results of Section 5.3.2. In this subsection, we consider how to aggregate the multi-exits according to duplicate continuation processes during composition.

To remove the duplication from multiple occurrence of the same continuation processes in $\overrightarrow{\mathbf{X}}$, the exits having the same continuation processes shall be merged, and the corresponding elements of multi-exit matrix $\mathbf{T}$ shall be added up. As an example, $\overrightarrow{\mathbf{X}} = (X_1, X_2, X_1)$, X_1 is duplicate and shall be merged. Henceforth, the composite MEME distribution shall have pairwise distinct con-

tinuation processes as desired, and the number of columns for the multi-exit matrix $\mathbf{T}$ is reduced accordingly.

For each multiple continuation process, we shall use an *Incidence Matrix* $\mathbf{M}$ to produce a reduced multi-exit matrix $\mathbf{T}'$ and continuation process $\overrightarrow{\mathbf{X}}'$. The elements of an incidence matrix $\mathbf{M}$ consist of only 0 and 1, which are commonly used to indicate the relationships between two objects. Assume that a distribution $[\![(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})_{p,n} \mapsto \overrightarrow{\mathbf{X}}]\!]$ has the same continuation process $\check{X}$ appearing at the $t_1, \ldots, t_k$-th position of $\overrightarrow{\mathbf{X}}$, where $k \leq n$ and $t_1 < \cdots < t_k$. From a $n \times n$ identity matrix $\mathbf{I} = \begin{bmatrix} \mathbf{e}_1 & \ldots & \mathbf{e}_n \end{bmatrix}$, we define an incident matrix $\mathbf{M}$ by deleting the $t_1, \ldots, t_{k-1}$ columns of $\mathbf{I}$ and replacing the t_k-th column of $\mathbf{I}$ with the column vector $\mathbf{e}_{t_1,\ldots,t_k}$, such that the $t_1, \ldots, t_k$-th elements are 1 and the remaining ones are 0. The obtained incident matrix $\mathbf{M}$ is of dimensions $n \times (n - k + 1)$, which shall look like

$$\mathbf{M} = \begin{bmatrix} \mathbf{e}_1 & \ldots & \mathbf{e}_{t_1-1} & \mathbf{e}_{t_1+1} & \ldots & \mathbf{e}_{t_k-1} & \mathbf{e}_{t_1,\ldots,t_k} & \mathbf{e}_{t_k+1} & \ldots & \mathbf{e}_n \end{bmatrix}.$$

The MEME representation after merging the multiple occurrence $\check{X}$ is given as $[\![(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T}')_{p,n-k+1} \mapsto \overrightarrow{\mathbf{X}}']\!]$, such that $\mathbf{T}' = \mathbf{TM}$ and $\overrightarrow{\mathbf{X}}'$ are $\overrightarrow{\mathbf{X}}$ with all $\check{X}$ deleted except for the last. In this manner, the multiple occurrence of $\check{X}$ are merged into the last $\check{X}$ of $\overrightarrow{\mathbf{X}}$, and the reduced multi-exits $\mathbf{T}'$ is produced.

EXAMPLE 5.4 *Let $[\![(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T}) \mapsto (X_1, X_2, X_1)]\!]$ be a valid distribution definition of the language* MEME *with multiple occurrence of X_1. To aggregate the multiple occurrence of X_1, we construct an incidence matrix $\mathbf{M}$ as*

$$\mathbf{M} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

That produces a distribution definition with multiple X_1 merged as $[\![(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T}') \mapsto (X_2, X_1)]\!]$, where $\mathbf{T}' = \mathbf{TM}$.

We shall follow the same pattern to construct incidence matrices $\mathbf{M}$ for each duplicate element of $\overrightarrow{\mathbf{X}}$. To ensure that all multiple continuation processes are eliminated, we shall aggregate all duplicate elements in $\overrightarrow{\mathbf{X}}$ with the order for example from left to right.

5.4.4 Explicit form of the MEME calculus

We take a transformational approach to the semantics of the MEME calculus, such that we enforce the compositionality and minimality of the primitives in our language through the transformation towards the explicit form. The explicit form shall later be related to the semantics of the language.

DEFINITION 5.10 (MEME *Explicit Form*) The *explicit form* of the language MEME is obtained by first removing all syntactic sugar (i.e. replacing shorthand expressions with explicit distribution definitions), and then minimising and solving all sorts of compositions for multi-exits matrix–exponential distributions.

To remove syntactic sugar, we use the algorithm $\mathcal{T}_{\text{sys}}$ to collect a MEME system *sys* embedded in the system definition *def* and associate initial probabilities *ini*. Here, the system definitions *def* can be any process definitions, e.g. terminating processes, compositional operators and distribution definitions. All terminating processes $\mathbf{0}$ shall remain unchanged during transformation. The shorthand expressions $\lambda.X$, let $MEME(\overrightarrow{\mathbf{X}})_i : P_i$ in P and $MEME(\overrightarrow{\mathbf{X}})_i$, are replaced by their corresponding distribution definitions, i.e. $[\![(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T}) \mapsto \overrightarrow{\mathbf{X}}]\!]$. The system definitions without syntactic sugar should look like the MEME definitions in Table 5.3 of Example 5.2.

Before we enforce compositionality, we first minimise all distribution definitions $[\![(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T}) \mapsto \overrightarrow{\mathbf{X}}]\!]$ into their minimal size. We shall use algorithm $\mathcal{T}_{\text{mini}}$ to replace each distribution definition with their corresponding minimal representation following Algorithm 2. Thereafter, we deal with all occurrence of race, synchronisation and probabilistic execution among MEME distributions via algorithms $\mathcal{T}_{\text{min}}$, $\mathcal{T}_{\text{max}}$ and $\mathcal{T}_{\text{prob}}$, respectively. The matrix operations for each algorithm have been explained in Section 5.3.2. The algorithm $\mathcal{T}_{\text{min}}$ replaces $\sum_{i \in I}^{\min} [\![(\boldsymbol{\alpha}_i, \mathbf{S}_i, \mathbf{T}_i)_{p_i, n_i} \mapsto \overrightarrow{\mathbf{X}_i}]\!]$ with $[\![(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})_{p,n} \mapsto \overrightarrow{\mathbf{X}}]\!]$, such that the later is a composite MEME distribution for the minimum of all participating distributions obtained by the Rule (5.4). The algorithm $\mathcal{T}_{\text{max}}$ replaces $\sum_{i \in I}^{\max} [\![(\boldsymbol{\alpha}_i, \mathbf{S}_i, \mathbf{T}_i)_{p_i, n_i} \mapsto \overrightarrow{\mathbf{X}_i}]\!]$ with $[\![(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})_{p,n} \mapsto \overrightarrow{\mathbf{X}}]\!]$, such that the later is a composite MEME distribution for the maximum of all participating distributions obtained by the Rule (5.5). The algorithm $\mathcal{T}_{\text{prob}}$ replaces $\sum_{i \in I}^{Pr} \pi_i [\![(\boldsymbol{\alpha}_i, \mathbf{S}_i, \mathbf{T}_i)_{p_i, n_i} \mapsto \overrightarrow{\mathbf{X}_i}]\!]$ with $[\![(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})_{p,n} \mapsto \overrightarrow{\mathbf{X}}]\!]$, such that the later is a composite MEME distribution for the probabilistic mixture of all participating distributions obtained by the Rule (5.6). Notice that the multiple occurrence of continuation processes in $\overrightarrow{\mathbf{X}}$ have also been merged using incidence matrices as discussed in Section 5.4.3 using algorithms $\mathcal{T}_{\text{mini}}$, $\mathcal{T}_{\text{min}}$, $\mathcal{T}_{\text{max}}$ and $\mathcal{T}_{\text{prob}}$. Therefore, the continuation processes of composite MEME representations shall always be pairwise distinct.

From Corollary 5.8, we know that a composite MEME representation is not necessarily minimal, even given each participating MEME distribution is minimal. To ensure that the composite MEME distribution is of minimal order, we shall apply algorithm $\mathcal{T}_{\text{mini}}$ for all composite MEME representations. For process definitions having nested compositional operators, we shall repeat the algorithms $\mathcal{T}_{\text{mini}}$, $\mathcal{T}_{\text{min}}$, $\mathcal{T}_{\text{max}}$ and $\mathcal{T}_{\text{prob}}$ until we have one explicit and minimal distribution definition.

Henceforth, we obtain a list of system definitions, such that each system process is either a termination 0 or a minimal represented MEME distribution. The explicit form of the MEME calculus shall look like

$$\cdots ; X_i\{ann_i\} := 0 \mid [\![(\boldsymbol{\alpha}, \mathbf{S}, \mathbf{T})_{p,n} \mapsto \overrightarrow{\mathbf{X}}]\!] ; \cdots .$$

5.4.5 Markov Renewal Process Semantics

From the explicit form of the MEME calculus, we define the semantics of the language as a Markov renewal process of matrix–exponential kernels with labels.

A Markov renewal process (cf. Section 2.2.5, Definition 2.9) represents both the transition epoch and the state of the process at that epoch. In our formalism, we have a matrix–exponential distribution associated to each state as the kernel of an MRP. That means, a random time, e.g. $W_{n+1} - W_n$, is represented by a random variable W; and the next state, e.g. H_{n+1}, is represented by a random variable G, such that the (W, G) is distributed according to a MEME representation. Notice that, since we do not consider immediate transitions, i.e. a transition cannot happen at the holding time equals 0, the transition epochs shall not be equal to each other, i.e. $0 < W_0 < W_1 < W_2 < \cdots$. Now we define an MRP with ME kernels.

Definition 5.11 (*Markov Renewal Process of ME Kernels*) A Markov Renewal Process (MRP) of Matrix–Exponential (ME) kernels with Labels (abbreviated $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$) is given by a tuple $(X, \boldsymbol{\pi}, \mathbf{MEME}(\cdot), \mathbf{L})$ where:

- $X = [x_i]_{i \leq m}$ is a set of *states*, such that m is finite;

- $\boldsymbol{\pi} = [\pi_i]_{i \leq m}$ is an *initial probability distribution*;

- $\mathbf{MEME}(\cdot) = [meme(\cdot)_{i,j}]_{i,j \leq m}$ is a *density function matrix with an argument in* $\Re_{>0}$, where each $meme(\cdot)_{i,j}$ is a probability density function denoting the density of a state change from i to j when the holding time is given by the argument;

- $\mathbf{L} = [L(x_i)]_{i \leq m}$ is a set of *labelling vectors*, such that $L(x_i) \subseteq AP$ is a labelling vector for state x_i.

DEFINITION 5.12 (MEME **Semantics**) The semantics $ini \mapsto def$ of a well-formed MEME system in the explicit form is the following $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$ $(X, \boldsymbol{\pi}, \mathbf{MEME}(\cdot), \mathbf{L})$:

- the set of states X consists of all process variables defined in def;

- the initial probability distribution $\boldsymbol{\pi}$ is defined in ini;

- let w be the argument, the i-th row of $\mathbf{MEME}(w)$ corresponds to the $X_i\{ann_i\}$ defined in def, such that when $X_i\{ann_i\} := [\![(\boldsymbol{\alpha}_i, \mathbf{S}_i, \mathbf{T}_i)_{p,n} \mapsto \overrightarrow{\mathbf{X}}]\!]$:

$$
meme(w)_{i,j} = \begin{cases} \boldsymbol{\alpha}_i e^{\mathbf{S}_i w} \mathbf{T}_{ig}, \text{ with } g\text{'s element of } \overrightarrow{\mathbf{X}} \text{ is } x_j, \\ 0, \text{ otherwise}; \end{cases}
$$

  when $X_i\{ann_i\} := 0$: $meme(w)_{i,j} = 0$ for all j (the probability densities for a terminating process jumping to other states are always 0).

- the labelling vector $\mathbf{L} = [L(x_i)]_{i \leq m}$, where $L(x_i) = ann_i$ when def contains $X_i\{ann_i\} := \cdots$.

EXAMPLE 5.5 *Returning to Example 5.3, assume that we have transformed the language* MEME *of Table 5.3 into the explicit form showed in Table 5.4. Here, we write* $[\![(\boldsymbol{\alpha}_{\mathsf{ON}}, \mathbf{S}_{\mathsf{ON}}, \mathbf{T}_{\mathsf{ON}})_{p_{\mathsf{ON}}} \mapsto (\mathsf{minD}, \mathsf{maxD})]\!]$ *as the resulting minimal representation of MEME distributions for state* ON *after transformation. Similar notations have been used for states* minD, maxD *and* OFF.

$$
\begin{aligned}
\mathsf{ON}\{\{\mathsf{normal}\}\} &:= [\![(\boldsymbol{\alpha}_{\mathsf{ON}}, \mathbf{S}_{\mathsf{ON}}, \mathbf{T}_{\mathsf{ON}}) \mapsto (\mathsf{minD}, \mathsf{maxD})]\!]; \\
\mathsf{minD}\{\mathsf{repair}\} &:= [\![(\boldsymbol{\alpha}_{\mathsf{minD}}, \mathbf{S}_{\mathsf{minD}}, \mathbf{T}_{\mathsf{minD}}) \mapsto (\mathsf{ON}, \mathsf{maxD}, \mathsf{OFF})]\!]; \\
\mathsf{maxD}\{\mathsf{repair}\} &:= [\![(\boldsymbol{\alpha}_{\mathsf{maxD}}, \mathbf{S}_{\mathsf{maxD}}, \mathbf{T}_{\mathsf{maxD}}) \mapsto (\mathsf{ON}, \mathsf{maxD}, \mathsf{OFF})]\!]; \\
\mathsf{OFF}\{\mathsf{break}\} &:= 0
\end{aligned}
$$

Table 5.4: The fault tolerant machine in explicit form.

Using the ordering $(\mathsf{ON}, \mathsf{minD}, \mathsf{maxD}, \mathsf{OFF})$ *on process names, we obtain the initial probability distribution* $\boldsymbol{\pi}$ *and the labelling vector* $\mathbf{L}$ *as*

$$
\boldsymbol{\pi} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \ \mathbf{L} = \{\{\mathsf{normal}\}, \{\mathsf{repair}\}, \{\mathsf{repair}\}, \{\mathsf{break}\}\},
$$

and we get the density fudnction matrix at time w, **MEME**(w), *as*

$$
\begin{bmatrix}
0 & \boldsymbol{\alpha}_{\mathsf{ON}}e^{\mathbf{S}_{\mathsf{ON}}w}\mathbf{T}_{\mathsf{ON1}} & \boldsymbol{\alpha}_{\mathsf{ON}}e^{\mathbf{S}_{\mathsf{ON}}w}\mathbf{T}_{\mathsf{ON2}} & 0 \\
\boldsymbol{\alpha}_{\mathsf{minD}}e^{\mathbf{S}_{\mathsf{minD}}w}\mathbf{T}_{\mathsf{minD1}} & 0 & \boldsymbol{\alpha}_{\mathsf{minD}}e^{\mathbf{S}_{\mathsf{minD}}w}\mathbf{T}_{\mathsf{minD2}} & \boldsymbol{\alpha}_{\mathsf{minD}}e^{\mathbf{S}_{\mathsf{minD}}w}\mathbf{T}_{\mathsf{minD3}} \\
\boldsymbol{\alpha}_{\mathsf{maxD}}e^{\mathbf{S}_{\mathsf{maxD}}w}\mathbf{T}_{\mathsf{maxD1}} & 0 & \boldsymbol{\alpha}_{\mathsf{maxD}}e^{\mathbf{S}_{\mathsf{maxD}}w}\mathbf{T}_{\mathsf{maxD2}} & \boldsymbol{\alpha}_{\mathsf{maxD}}e^{\mathbf{S}_{\mathsf{maxD}}w}\mathbf{T}_{\mathsf{maxD3}} \\
0 & 0 & 0 & 0
\end{bmatrix}.
$$

5.4.6 Logics for $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$

Continuous Stochastic Logic (CSL) [ASSB96] allows to specify state as well as path-based properties on CTMCs [BHHK03]. Previous work has extended CSL on semi-Markov chains [LHK01] and semi–Markov Petri net [BDHK03]. In this work, we extend CSL to Markov renewal processes with matrix–exponential kernels. Before the formal syntax and semantics of CSL, we first introduce the basic concepts of $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$. Consider an $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$ $\mathcal{M} = (X, \boldsymbol{\pi}, \mathbf{MEME}(w), \mathbf{L})$.

Paths An infinite path σ of $\mathcal{M}$ is a sequence $x_0 \overset{w_0}{\to} x_1 \overset{w_1}{\to} \ldots$ with state $x_i \in X$ and holding time $w_i \in \Re_{>0}$, such that $meme(w_i)_{x_i, x_{i+1}} > 0$ for all i. A finite path σ of $\mathcal{M}$ is a sequence $x_0 \overset{w_0}{\to} x_1 \overset{w_1}{\to} \cdots x_{l-1} \overset{w_{l-1}}{\to} x_l$ such that $meme(w_i)_{x_i, x_{i+1}} > 0$ for all $i < l$ and x_l is absorbing, i.e. $\forall w \in \Re_{>0}, \forall x_{l'} \in X$: $meme(w)_{x_l, x_{l'}} = 0$. For an infinite path σ and $i \in \mathbb{N}_0$, let $\sigma(i) = x_i$ be the $(i+1)$-st state of σ and $\delta(\sigma, i) = w_i$ be the holding time spent in x_i. For $t \in \Re_{\geq 0}$ and i the smallest index satisfying $t \leq \sum_{j=0}^{i} w_j$, let $\sigma@t = \sigma(i)$ be the state in σ occupied at time t. Let $Path$ denote the set of paths in $\mathcal{M}$ and $Path(x)$ denote the set of paths in $\mathcal{M}$ that start in x.

Borel space A probability measure $\mathcal{P}^{\mathcal{M}}$ on the set of paths, $Path$, of $\mathcal{M}$ is defined using the standard cylinder construction. Let $x_0, \ldots, x_l \in X$ with $\exists w_i$: $meme(w_i)_{x_i, x_{i+1}} > 0$ for $0 \leq i < l$, and let $I_0, \ldots, I_{l-1}$ be non-empty intervals in $\Re_{\geq 0}$. Then, let $\mathcal{C}(x_0, I_0, \ldots, I_{l-1}, x_l)$ denote the *cylinder set* consisting of all paths $\sigma \in Path(x_0)$ such that $\sigma(i) = x_i$ for all $i \leq l$ and $\delta(\sigma, i) \in I_i$ for all $i < l$. Let $\mathcal{F}(path)$ be the smallest σ-algebra on $Path$ which contains all sets $\mathcal{C}(x, I_0, \ldots, I_{l-1}, x_l)$, where $x, \ldots, x_l$ range over all state sequences satisfying $x = x_0$, $\exists w_i$: $meme(w_i)_{s_i, s_{i+1}} > 0$ with $0 \leq i < l$, and $I_0, \ldots, I_{l-1}$ range over all sequences of nonempty intervals in $\Re_{\geq 0}$. The probability measure $\mathcal{P}^{\mathcal{M}}$ on $\mathcal{F}(path)$ is the unique measure defined by induction on l by the base case

$\mathcal{P}^{\mathcal{M}}(\mathcal{C}(x_0)) = 1$, and for $l \geq 0$:

$$\mathcal{P}^{\mathcal{M}}(\mathcal{C}(x, I_0, \ldots, I_{l-1}, x_l, I', x')) = \mathcal{P}^{\mathcal{M}}(\mathcal{C}(x, I_0, \ldots, I_{l-1}, x_l))\mathcal{P}^{\mathcal{M}}(x_l, I', x'),$$

such that $\mathcal{P}^{\mathcal{M}}(x_l, I', x')$ is the probability of a state jump from x_l to x' in the interval I'. Let $meme(w)_{x_l, x'} = \boldsymbol{\alpha}e^{\mathbf{S}w}\mathbf{T}_g$ be corresponding density function, $a = \inf I'$ and $b = \sup I'$, $\mathcal{P}^{\mathcal{M}}(x_l, cf(I'), x')$ is computed as

$$\mathcal{P}^{\mathcal{M}}(x_l, I', x') = \int_a^b \boldsymbol{\alpha}e^{\mathbf{S}w}\mathbf{T}_g dw. \tag{5.7}$$

For any state x_0, the probability measure of the set of infinite path $x_0 \xrightarrow{w_0} x_1 \xrightarrow{w_1}$ ... is zero. Now, we are ready to introduce the syntax of CSL for $\mathsf{MRP}_{\mathsf{MEME}}^{\mathsf{LR}}$.

DEFINITION 5.13 (***CSL Syntax***) Given probability $p \in [0,1]$, comparison operator $\bowtie \in \{<, \leq, >, \geq\}$, $I \subseteq \Re_{\geq 0}$, and $a \in AP$, the syntax of CSL consists of *state* formulae Φ and *path* formulae Ψ, is defined by the following formulae:

$$\Phi \quad ::= \quad \mathsf{true} \mid a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \mathsf{S}_{\bowtie p}[\Phi] \mid \mathsf{P}_{\bowtie p}[\Psi]$$
$$\Psi \quad ::= \quad \mathsf{X}^I\Phi \mid \Phi_1\mathsf{U}^I\Phi_2$$

Other boolean connectives can be derived as, e.g. $\mathsf{false} = \neg\mathsf{true}$, $\Phi_1 \vee \Phi_2 = \neg(\neg\Phi_1 \wedge \neg\Phi_2)$ and $\Phi_1 \rightarrow \Phi_2 = \neg\Phi_1 \vee \Phi_2$. The state formula $\mathsf{S}_{\bowtie p}[\Phi]$ asserts the *limiting probability measure* $\mathcal{P}_x^S$ for all Φ-states satisfying the condition $\bowtie p$ given the initial state x. We shall write $\mathcal{P}_x^S(x')$ as the limiting probability of state x' starting from the state x. The state formula $\mathsf{P}_{\bowtie p}[\Psi]$ asserts the *probabilistic path measure* $\mathcal{P}_x^{\mathcal{C}}$ of Ψ-$Path(x)$ satisfying the condition $\bowtie p$. The path formula $\mathsf{X}^I\Phi$ denote the paths where Φ holds in the next state at some time point within the interval I. The path formula $\Phi_1\,\mathsf{U}^I\,\Phi_2$ denote the paths where, $\exists t \in I$, Φ_1 continuously holds during the interval $(\inf I, t)$ and Φ_2 eventually holds from time t. The usual unbound until formula is coded when the interval I has $\sup I = \infty$.

The meaning of CSL state formulae and path formulae are defined by means of a satisfaction relation $\vDash$. Now we introduce the formal semantics.

DEFINITION 5.14 (***CSL Semantics***) The state formulae of CSL are interpreted over the states of an $\mathsf{MRP}_{\mathsf{MEME}}^{\mathsf{LR}}$. Let $\mathcal{M} = (X, \boldsymbol{\pi}, \mathbf{MEME}(w), \mathbf{L})$, the satisfaction relation $\vDash$ is defined by

- $x \vDash \mathsf{true}$ for all $x \in X$,

- $x \vDash a$ iff $a \in L(x)$,

- $x \vDash \neg\Phi$ iff $x \nvDash \Phi$,

- $x \vDash \Phi_1 \wedge \Phi_2$ iff $x \vDash \Phi_1$ and $x \vDash \Phi_2$,

- $x \vDash S_{\bowtie p}[\Phi]$ iff $\sum_{x' \vDash \Phi} \mathcal{P}_x^S(x') \bowtie p$,

- $x \vDash P_{\bowtie p}[\Psi]$ iff $\mathcal{P}_x^{\mathcal{C}}(\Psi) \bowtie p$,

such that $\mathcal{P}_x^{\mathcal{C}}(\Psi)$ denotes the probability measure of all paths $\sigma \in Path(x)$ satisfying Ψ. The satisfaction relation for the path formulae is defined by:

- $\sigma \vDash X^I \Phi$ iff $\sigma(1)$ is defined, $\sigma(1) \vDash \Phi$ and $\delta(\sigma, 0) \in I$,

- $\sigma \vDash \Phi_1 U^I \Phi_2$ iff $\exists t \in I$ such that $\sigma@t \vDash \Phi_2$ and $\forall t' \in [0, t)$ $\sigma@t' \vDash \Phi_1$.

Consider the FTM in Example 5.5, we shall write $P_{<0.1}[\text{repair } U^{[2,4]} \text{break}]$ to express a probabilistic timed reachability property that once the machine starts repairing, the probability of the machine breaking down between 2 and 4 hours is less than 0.1.

5.5 Transient and Limiting Probabilities

The semantics of a well-formed MEME system is a Markov renewal process with matrix–exponential kernels and labels $\mathsf{MRP}_{\mathsf{MEME}}^{\mathsf{LR}}$. In this section, we discuss transient and limiting probabilities on an $\mathsf{MRP}_{\mathsf{MEME}}^{\mathsf{LR}}$. We shall expand all MEME representations of an $\mathsf{MRP}_{\mathsf{MEME}}^{\mathsf{LR}}$ to express a stochastic process, which essentially is not a CTMC. However, the equation systems for computing transient and limiting probabilities are similar to those of CTMCs. Henceforth, we are able to apply most numerical methods of CTMCs in the MEME setting.

Given an $\mathsf{MRP}_{\mathsf{MEME}}^{\mathsf{LR}}$ $(X, \boldsymbol{\pi}, \mathbf{MEME}(w), \mathbf{L})$, for each state $x_i \in X$ we shall find a MEME representation $\mathrm{MEME}_{p_i, n_i}(\boldsymbol{\alpha}_i, \mathbf{S}_i, \mathbf{T}_i)$, such that we write the representation $\mathrm{MEME}_{1,1}([1], [0], [0])$ for all absorbing states. From that, we construct a matrix $\mathbf{Q}$ as the expansions of all MEME representations, which looks like

$$\mathbf{Q} = \begin{bmatrix} \mathbf{S}_1 & \mathbf{T}_{12}\boldsymbol{\alpha}_2 & \cdots & \mathbf{T}_{1m}\boldsymbol{\alpha}_m \\ \mathbf{T}_{21}\boldsymbol{\alpha}_1 & \mathbf{S}_2 & \cdots & \mathbf{T}_{2m}\boldsymbol{\alpha}_m \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{T}_{m1}\boldsymbol{\alpha}_1 & \mathbf{T}_{m2}\boldsymbol{\alpha}_2 & \cdots & \mathbf{S}_m \end{bmatrix}_{\sum_{i=1}^m p_i \times \sum_{i=1}^m p_i} ,$$

where $\mathbf{T}_{ij}$ is the g-th column of matrix $\mathbf{T}_i$, given the g-th element of $\overrightarrow{\mathbf{X}}_i$ is X_j; otherwise $\mathbf{T}_{ij} = \mathbf{0}$. Notice that there is no proper probabilistic interpretation of a single entry of the $\mathbf{Q}$ matrix. However, probabilistic interpretation in term of an MRP remains valid on block matrices: the matrix $\mathbf{S}_i$ corresponds to the state i of the MRP, and the matrix $\mathbf{T}_{ij}\boldsymbol{\alpha}_j$ corresponds to the jump from state i to state j of the MRP. Therefore, after a MEME system enters a state i, it shall stay in the state for some time corresponding to $\mathbf{S}_i$, then jump to other states, such that jumping to the state j corresponds to the block $\mathbf{T}_{ij}\boldsymbol{\alpha}_j$.

We shall also expand the initial probability vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_m)$ with respect to the different row vectors $\boldsymbol{\alpha}$ of the MEME representations. That defines an initial vector $\boldsymbol{\pi}_{\mathbf{Q}} = (\underbrace{\pi_1\boldsymbol{\alpha}_1}_{p_1}, \underbrace{\pi_2\boldsymbol{\alpha}_2}_{p_2}, \ldots, \underbrace{\pi_m\boldsymbol{\alpha}_m}_{p_m})$.

DEFINITION 5.15 (*Transient Vector*) Let $\boldsymbol{\beta}(t)$ be an transient vector for the time t having the same dimensions as $\boldsymbol{\pi}_{\mathbf{Q}}$, and $\boldsymbol{\beta}(t)_i \in \Re$ be the i-th element of $\boldsymbol{\beta}(t)$, the vector $\boldsymbol{\beta}(t)$ satisfies that

$$\sum_{j=1}^{\sum_{i=1}^{m} p_i} \boldsymbol{\beta}(t)_j = 1, \ \boldsymbol{\beta}(0) = \boldsymbol{\pi}_{\mathbf{Q}}, \ \text{and} \ \boldsymbol{\beta}'(t) = \boldsymbol{\beta}(t)\mathbf{Q}, \ \text{for} \ t > 0. \quad (5.8)$$

This entails that $\boldsymbol{\beta}(t) = \boldsymbol{\pi}_{\mathbf{Q}} e^{\mathbf{Q}t}$.

The equation system of (5.8) is standard for calculating transient probabilities on CTMCs, such that we need to solve a system of linear differential equations to compute the transient vector $\boldsymbol{\beta}(t)$, for instance using Runge-Kutta methods [Ste94]. To evaluate the transient probabilities of being in the states of $\mathsf{MRP}_{\mathsf{MEME}}^{\mathsf{LR}}$, we shall aggregate $\boldsymbol{\beta}(t)$. Let $(\widehat{\boldsymbol{\beta}(t)}_1, \widehat{\boldsymbol{\beta}(t)}_2, \ldots, \widehat{\boldsymbol{\beta}(t)}_m)$ be a partition of $\boldsymbol{\beta}(t)$, such that $\widehat{\boldsymbol{\beta}(t)}_1$ is a vector containing the first p_1 elements, $\widehat{\boldsymbol{\beta}(t)}_2$ is a vector containing the successive p_2 elements after $\widehat{\boldsymbol{\beta}(t)}_1$, and so on. The *transient probability vector* $\boldsymbol{\pi}(t)$ of an $\mathsf{MRP}_{\mathsf{MEME}}^{\mathsf{LR}}$ is defined as $\boldsymbol{\pi}(t) = (\widehat{\boldsymbol{\beta}(t)}_1\mathbf{e}, \widehat{\boldsymbol{\beta}(t)}_2\mathbf{e}, \ldots, \widehat{\boldsymbol{\beta}(t)}_m\mathbf{e})$, such that the probability of being in state i after t time units is calculated by the sum of all its elements from the transient vector, i.e. $\widehat{\boldsymbol{\beta}(t)}_i\mathbf{e}$, with the condition that $(\widehat{\boldsymbol{\beta}(0)}_1\mathbf{e}, \widehat{\boldsymbol{\beta}(0)}_2\mathbf{e}, \ldots, \widehat{\boldsymbol{\beta}(0)}_m\mathbf{e}) = \boldsymbol{\pi}$.

The *limiting probability distribution* $\boldsymbol{\pi}(\infty)$ is defined straightforwardly when time t tends to ∞, i.e. $\boldsymbol{\pi}(\infty) = (\widehat{\boldsymbol{\beta}(\infty)}_1\mathbf{e}, \widehat{\boldsymbol{\beta}(\infty)}_2\mathbf{e}, \ldots, \widehat{\boldsymbol{\beta}(\infty)}_m\mathbf{e})$. For the limiting of transient vector, the following equation holds

$$\boldsymbol{\beta}(\infty) = \boldsymbol{\beta}(\infty) e^{\mathbf{Q}t}, \ \text{for} \ t \geq 0.$$

Like CTMCs, an $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$ is called *irreducible* when each of its states can reach every other state with some finite paths in the embedded Markov chain, and *reducible* otherwise. For an irreducible $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$, the limiting probability distribution is independent of the initial distribution, which always exists and shall be calculated by solving the following linear equation systems

$$\boldsymbol{\beta}(\infty)\mathbf{Q} = \mathbf{0}, \text{ and } \sum_{j=1}^{\sum_{i=1}^{m} p_i} \boldsymbol{\beta}(\infty)_j = 1. \tag{5.9}$$

OBSERVATION 3 *We have that $\boldsymbol{\beta}(\infty) = \boldsymbol{\beta}(\infty)e^{\mathbf{Q}t}$ for $t \geq 0$ if and only if $\boldsymbol{\beta}(\infty)\mathbf{Q} = \mathbf{0}$.*

PROOF

$$\boldsymbol{\beta}(\infty)\mathbf{Q} = \mathbf{0} \Leftrightarrow \boldsymbol{\beta}(\infty)\mathbf{Q}^n = \mathbf{0}, \text{ for all } n \geq 1$$

$$\Leftrightarrow \boldsymbol{\beta}(\infty) \sum_{n=1}^{\infty} \frac{t^n}{n!}\mathbf{Q}^n = \mathbf{0}, \text{ for } t \geq 0$$

$$\Leftrightarrow \boldsymbol{\beta}(\infty) \sum_{n=0}^{\infty} \frac{t^n}{n!}\mathbf{Q}^n = \mathbf{0}, \text{ for } t \geq 0$$

$$\Leftrightarrow \boldsymbol{\beta}(\infty)e^{\mathbf{Q}t} = \boldsymbol{\beta}(\infty), \text{ for } t \geq 0$$

$\square$

The equation system of (5.9) is standard for the steady-state probabilities on CTMCs, such that well-known numerical methods, for instance Gaussian elimination [Ste94], could be used.

Numerical Considerations We have mentioned that the equation systems (5.8) and (5.9) of $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$ are standard as CTMCs, however some numerical methods for CTMCs may not be adaptable to our problem because of the generalisation from ME distributions, e.g. non-diagonal negative elements and row sum does not equal zero. For instance, the popular uniformisation method [BHHK03, KNP07] is not directly applicable for transient probabilities, as the discretised transition probability matrix might have negative entries. And in general the equations (5.9) might exist multiple solutions. However, this is outside the scope of this work and we put this as an open question, which needs further investigations.

To test our approach, we have made experimental studies on the MATLAB platform taking some numerical examples of the FTM. We report one of them as Example 5.6.

EXAMPLE 5.6 *Returning to Example 5.5, from the semantics of FTM, we shall find the square matrix* $\mathbf{Q}$ *of dimension* $p_{\mathsf{ON}} + p_{\mathsf{minD}} + p_{\mathsf{maxD}} + 1$ *as*

$$
\mathbf{Q} = \begin{bmatrix} \mathbf{S}_{\mathsf{ON}} & \mathbf{T}_{\mathsf{ON}1}\boldsymbol{\alpha}_{\mathsf{minD}} & \mathbf{T}_{\mathsf{ON}2}\boldsymbol{\alpha}_{\mathsf{maxD}} & \mathbf{0} \\ \mathbf{T}_{\mathsf{minD}1}\boldsymbol{\alpha}_{\mathsf{ON}} & \mathbf{S}_{\mathsf{minD}} & \mathbf{T}_{\mathsf{minD}2}\boldsymbol{\alpha}_{\mathsf{maxD}} & \mathbf{T}_{\mathsf{minD}3} \\ \mathbf{T}_{\mathsf{maxD}1}\boldsymbol{\alpha}_{\mathsf{ON}} & \mathbf{T}_{\mathsf{maxD}2}\boldsymbol{\alpha}_{\mathsf{minD}} & \mathbf{S}_{\mathsf{maxD}} & \mathbf{T}_{\mathsf{maxD}3} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 \end{bmatrix},
$$

initial probability vector $\boldsymbol{\pi}_{\mathbf{Q}}$ *as*

$$
\boldsymbol{\pi}_{\mathbf{Q}} = (\underbrace{\boldsymbol{\alpha}_{\mathsf{ON}}}_{p_{\mathsf{ON}}}, \underbrace{0, \ldots, 0}_{p_{\mathsf{minD}}}, \underbrace{0, \ldots, 0}_{p_{\mathsf{maxD}}}, 0).
$$

To compute transient probabilities, we compose three MEME representations for the states ON, minD *and* maxD *through probabilistic mixture, minimum and maximum, respectively, using the ME-3 distribution of Example 5.1 and the ME-3 distribution of [BN03]. The matrix* $\mathbf{Q}$ *after expansion is of dimensions* 34×34 *with* $p_{\mathsf{ON}} = 15$, $p_{\mathsf{minD}} = 9$, *and* $p_{\mathsf{maxD}} = 9$. *Since the expanded matrix* $\mathbf{Q}$ *is rather huge, we omit the detailed entries. We compute the transient vector* $\boldsymbol{\beta}(8.34) = (0.0059, -0.0041, 0.0008, -0.1754, 0.0099, -0.0153, 0.2202, -0.0012, 0.0213, -0.0002, -0.0178, 0.0252, 0.0477, 0.0595, 0.0712, 0.0049, -0.0032, 0.0008, -0.1370, 0.0089, -0.0110, 0.1726, -0.0020, 0.0157, 0.0007, -0.0508, 0.0686, 0.0395, 0.0313, 0.0407, 0.0018, -0.1270, 0.1714, 0.5268),$ *and then transient probability vector* $\boldsymbol{\pi}(8.34) = \begin{bmatrix} 0.2477 & 0.0495 & 0.1760 & 0.5268 \end{bmatrix}$. *That interprets, after running the FTM 8.34 time units, the probabilities of FTM being* ON, minD, maxD *and* OFF *are 0.2477, 0.0495, 0.1760 and 0.5268, respectively. After running 44.4 time unites, the chance of the FTM getting broken is more than 99%, i.e.* $\boldsymbol{\beta}(44.4) = (0.0001, -0.0001, 0.0000, -0.0035, 0.0002, -0.0003, 0.0044, -0.0000, 0.0004, -0.0000, -0.0003, 0.0005, 0.0010, 0.0013, 0.0016, 0.0001, -0.0001, 0.0000, -0.0027, 0.0001, -0.0002, 0.0033, -0.0000, 0.0003, 0.0000, -0.0011, 0.0015, 0.0009, 0.0006, 0.0008, 0.0000, -0.0028, 0.0038, 0.9900)$ *and* $\boldsymbol{\pi}(44.4) = \begin{bmatrix} 0.0054 & 0.0010 & 0.0037 & 0.9900 \end{bmatrix}$.

Constructing and computing such an example in MATLAB executes instantaneously.

5.6 Model Checking $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$ against CSL

In this section, we study model checking algorithms to ascertain stochastic properties on a well-formed MEME system. We first introduce how to transform an $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$ to a semi-Markov chain in order to adapt model checking algorithms for SMCs [LHK01]. Afterwards, we illustrate the direct model checking

algorithms for $\mathsf{MRP^{LR}_{MEME}}$, which are similar to the standard model checking algorithms on CTMCs [BHHK03, KNP07].

5.6.1 Transform $\mathsf{MRP^{LR}_{MEME}}$ to Semi-Markov Chains

Consider only the states of a Markov renewal process, one shall obtain a semi-Markov process. Since an $\mathsf{MRP^{LR}_{MEME}}$ has a discrete state space, its semi-Markov processes are Semi-Markov Chains (SMCs). Model checking SMCs against CSL formulae have been studied in [LHK01], we first show how to derive a SMC from an $\mathsf{MRP^{LR}_{MEME}}$, and the model checking CSL formulae are similar to the MRP case at the later section.

Consider an $\mathsf{MRP^{LR}_{MEME}}$ $\mathcal{M} = (X, \boldsymbol{\pi}, \mathbf{MEME}(w), \mathbf{L})$, each $meme(w)_{i,j}$ relates to a joint density function $f(w, g)$ of MEME distributions. Integrating each density function $meme(w)_{i,j}$ of $\mathbf{MEME}(w)$ on the non-negative real line, i.e. $F(\infty, g) = \int_0^\infty f(x, g)dx$ (cf. Formula (5.3)), we shall obtain a transition matrix $\mathbf{P}$ of the embedded Markov chain, denoted as $emb(\mathcal{M})$. Let $\mathcal{P}^{emb(\mathcal{M})}(x, \bar{x})$ denote the probability of reaching state $\bar{x}$ from state x in the embedded Markov chain, and let $\mathbf{P}_{x,x'}$ be the one step transition probability from state x to state x', it is known that $\mathcal{P}^{emb(\mathcal{M})}(x, \bar{x})$ can be computed as the least solution of the linear equation system:

$$\mathcal{P}^{emb(\mathcal{M})}(x, \bar{x}) = \begin{cases} 1, \text{ if } \mathrm{x} = \bar{x}; \\ \sum_{x' \in X} \mathbf{P}_{x,x'} \mathcal{P}^{emb(\mathcal{M})}(x', \bar{x}), \text{ otherwise.} \end{cases}$$

Recall Definition 2.9 of an MRP, for some state i and j, the $F(w, g)$ is the probability $\mathbb{P}(H_{n+1} = j, W_{n+1} - W_n \leq w \mid H_n = i)$, and the $F(\infty, g)$ is the probability $\mathbb{P}(H_{n+1} = j \mid H_n = i)$. Thus, we derive a conditional probability $Q(w)_{ij}$ as the probability to jump from i to j within w time units given that a transition from i to j will be taken, i.e.

$$\mathbb{P}\left(W_{n+1} - W_n \leq w \mid (H_{n+1} = j | H_n = i)\right) = \frac{\mathbb{P}(H_{n+1} = j, W_{n+1} - W_n \leq w \mid H_n = i)}{\mathbb{P}(H_{n+1} = j \mid H_n = i)}.$$

For some corresponding exit g, we compute $Q(w)_{ij}$ as

$$Q(w)_{ij} = \frac{F(w, g)}{F(\infty, g)} = \frac{\boldsymbol{\alpha}\mathbf{S}^{-1}(e^{\mathbf{S}w} - \mathbf{I})\mathbf{T}_g}{\boldsymbol{\alpha}(-\mathbf{S})^{-1}\mathbf{T}_g}.$$

Let $\mathbf{Q}(w)$ be the matrix consisting of $Q(w)_{ij}$ for $i, j = 1, \ldots, m$, X and $\mathbf{L}$ remain unchanged from $\mathsf{MRP^{LR}_{MEME}}$, the tuple $(X, \mathbf{P}, \mathbf{Q}(w), \mathbf{L})$ defines a Semi-Markov

chain. The model checking $(X, \mathbf{P}, \mathbf{Q}(w), \mathbf{L})$ against CSL follows the methods mentioned in [LHK01], which only consider the average amount of time spent in some state as long run behaviour, i.e. taking the expected values of all the distributions $Q(w)_{i,j}$. However, since the limiting distributions of an $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$ always exist, we shall be able to perform more precise measurements directly on $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$ rather than the SMCs approach.

5.6.2 Model Checking Markov Renewal Processes

For an $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$ $\mathcal{M} = (X, \boldsymbol{\pi}, \mathbf{MEME}(w), \mathbf{L})$, we could specify interesting measurements by CSL formulae. The model checking algorithm takes a $\mathcal{M}$ and a CSL formula Φ as inputs, and then produces a set of satisfied states, i.e. $Sat(\Phi) = \{x \in X \mid x \vDash \Phi\}$, as output. To know if a state x satisfies the formula Φ, we only need to find if $x \in Sat(\Phi)$. The set $Sat(\Phi)$ is computed recursively for the sub-formulas of Φ. The $Sat(\Phi)$ for each state formula is defined as

- $Sat(\mathsf{true}) = X$

- $Sat(a) = \{x \in X \mid a \in L(x)\}$

- $Sat(\neg\Phi) = X \setminus Sat(\Phi)$

- $Sat(\Phi_1 \wedge \Phi_2) = Sat(\Phi_1) \cap Sat(\Phi_2)$

- $Sat(\mathsf{S}_{\bowtie p}[\Phi]) = \{x \in X \mid \sum_{x' \vDash \Phi} \mathcal{P}^S_x(x') \bowtie p\}$

- $Sat(\mathsf{P}_{\bowtie p}[\Psi]) = \{x \in X \mid \mathcal{P}^{\mathcal{C}}_x(\Psi) \bowtie p\}$

Model checking for the majority of state formulae is trivial to implement with the exceptions of $\mathsf{S}_{\bowtie p}[\Phi]$ and $\mathsf{P}_{\bowtie p}[\Psi]$. We shall deal with them separately at below.

$\mathsf{S}_{\bowtie p}[\Phi]$ **formulae.** From the semantics of CSL state formulae, a state $x \vDash \mathsf{S}_{\bowtie p}[\Phi]$ iff $\sum_{x' \vDash \Phi} \mathcal{P}^S_x(x') \bowtie p$. If $\mathcal{M}$ is irreducible, the steady-state probability vector $\boldsymbol{\pi}(\infty)$ is computed directly by solving Formulae (5.9) and then aggregated accordingly. The case when $\mathcal{M}$ is reducible needs extra care.

Like CTMCs, we define $\mathcal{B}$, a *Bottom Strongly Connected Component (BSCC)* of $\mathcal{M}$, as a set of states of $\mathcal{M}$, such that once the process enters a BSCC, it cannot leave any more. Computation of all BSCCs of $\mathcal{M}$, i.e. $\mathcal{B}(\mathcal{M})$, requires an analysis

of the underlying graph structure and is known as for CTMCs [Tar71]. Each $\mathcal{B} \in \mathcal{B}(\mathcal{M})$ shall be treated as an irreducible $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$ to compute its steady-state probabilities $\boldsymbol{\pi}^{\mathcal{B}}(\infty)$, and we shall use $\boldsymbol{\pi}^{\mathcal{B}}(\infty)_{x'}$ to denote the corresponding steady-state probability of state $x' \in \mathcal{B}$. Afterwards, we compute the probability of reaching each $\mathcal{B}$ from the state x according to the embedded DTMC $emb(\mathcal{M})$, which is given as the reachability probability $\sum_{\bar{x} \in \mathcal{B}} \mathcal{P}^{emb(\mathcal{M})}(x, \bar{x})$. Henceforth, the limiting probability $\mathcal{P}^S_x(x')$ for a reducible $\mathcal{M}$ is computed as

$$\mathcal{P}^S_x(x') = \begin{cases} \left(\sum_{\bar{x} \in \mathcal{B}} \mathcal{P}^{emb(\mathcal{M})}(x, \bar{x}) \right) \boldsymbol{\pi}^{\mathcal{B}}(\infty)_{x'}, \text{if } x' \in \mathcal{B} \text{ for some } \mathcal{B} \in \mathcal{B}(\mathcal{M}); \\ 0 \quad \text{otherwise.} \end{cases}$$

$\mathsf{P}_{\bowtie p}[\Psi]$ **formulae.** From the CSL semantics, a state $x \vDash \mathsf{P}_{\bowtie p}[\Psi]$ iff $\mathcal{P}^{\mathcal{M}}(\mathcal{C}(x, \dots)) \bowtie p$, such that $\mathcal{C}(x, \dots)$ is the cylinder set of all paths $\sigma \in Path(x)$ and $\sigma \vDash \Psi$.

When $\Psi ::= \mathsf{X}^I \Phi$, we shall measure the set of paths satisfying $\mathsf{X}^I \Phi$ starting from x, i.e. $\mathcal{P}^{\mathcal{C}}_x(\mathsf{X}^I \Phi)$. Assume $x' \vDash \Phi$ and $meme(w)_{x,x'} = \boldsymbol{\alpha} e^{\mathbf{S}w} \mathbf{T}_g$, the probability of jumping from x to x' during the closure of interval I is given as $\mathcal{P}^{\mathcal{M}}(x, cf(I), x')$ (cf. Formula (5.7) in the Borel space construction). Henceforth, $x \vDash \mathsf{P}_{\bowtie p}[\mathsf{X}^I \Phi]$ iff $\sum_{x' \vDash \Phi} \mathcal{P}^{\mathcal{M}}(x, cf(I), x') \bowtie p$.

When $\Psi ::= \Phi_1 \mathsf{U}^I \Phi_2$, we shall measure the paths satisfying $\Phi_1 \mathsf{U}^I \Phi_2$ starting from x, i.e. $\mathcal{P}^{\mathcal{C}}_x(\Phi_1 \mathsf{U}^I \Phi_2)$, and then evaluate the condition $\bowtie p$. Depending on the interval type, we consider three cases.

1. If $I = [0, t]$, we shall determine the least solution of the following integral equation system.

$$\mathcal{P}^{\mathcal{C}}_x(\Phi_1 \mathsf{U}^{[0,t]} \Phi_2) = \begin{cases} 0, \text{ when } x \in Sat(\neg \Phi_1 \wedge \neg \Phi_2); \\ 1, \text{ when } x \in Sat(\Phi_2); \\ \int_0^t \mathcal{P}^{\mathcal{M}}(x, cf([0, u]), x') \mathcal{P}^{\mathcal{C}}_{x'}(\Phi_1 \mathsf{U}^{[0,t-u]} \Phi_2) du, \text{ otherwise.} \end{cases}$$

The computation can be classified as a system of Volterra integral equations, such that one could solve it using Volterra-Runge-Kutta methods with worst case complexity $\mathcal{O}(n^4)$, where n is the number of states in $\mathcal{M}$ [HB86]. The above numerical approach is generally slow and only works for small systems. We shall adapt the alternative transformation approach mentioned in [BHHK03, KNP07] to our problem.

DEFINITION 5.16 (*CSL Driven Transformation*) For any $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$ $\mathcal{M} = (X, \boldsymbol{\pi}, \mathbf{MEME}(w), \mathbf{L})$ and CSL formula Φ, let $\mathsf{MRP}^{\mathsf{LR}}_{\mathsf{MEME}}$ $\mathcal{M}[\Phi]$ result from $\mathcal{M}$ by making all Φ states in $\mathcal{M}$ absorbing, i.e. $\mathcal{M}[\Phi] =$

$(X, \boldsymbol{\pi}, \mathbf{MEME}(w)', \mathbf{L})$, such that $meme(w)'_{i,j} = meme(w)_{i,j}$ if $x_i \nvDash \Phi$ and 0 otherwise.

For $\mathcal{P}_x^{\mathcal{C}}(\Phi_1 \mathsf{U}^{[0,t]} \Phi_2)$, we are safe to make all $\neg \Phi_1$ states absorbing because any path going through a $\neg \Phi_1$ state before reaching Φ_2 violates Ψ. We could also make Φ_2 states absorbing, because once a Φ_2 state is reached, the future behaviour becomes irrelevant for the validity of Ψ. Henceforth, we have $\mathcal{P}_x^{\mathcal{C}}(\Phi_1 \mathsf{U}^{[0,t]} \Phi_2) = \sum_{x' \vDash \Phi_2} \mathcal{P}^{\mathcal{M}[\neg \Phi_1 \vee \Phi_2]}(\mathcal{C}(\{\sigma \in Path(x) \mid \sigma @ t = x'\}))$, such that $\mathcal{P}^{\mathcal{M}[\neg \Phi_1 \vee \Phi_2]}(\mathcal{C}(\{\sigma \in Path(x) \mid \sigma @ t = x'\}))$ is the transient probability of state x' after t time units given the initial state is x. The detailed computation has been explained in Section 5.5 (cf. equation (5.8)).

2. If $I = [t, t']$, we have two parts of probabilities: the probability of staying at Φ_1 state up to time t (similar to the former) and the probability of reaching Φ_2 within $t' - t$ given Φ_1 is satisfied with $[0.t)$. That is

$$\mathcal{P}_x^{\mathcal{C}}(\Phi_1 \mathsf{U}^{[t,t']} \Phi_2) = \begin{cases} 0, \text{ when } x \notin Sat(\Phi_1); \\ \sum_{x' \vDash \Phi_1} \mathcal{P}^{\mathcal{M}[\neg \Phi_1]}(\mathcal{C}(\{\sigma \in Path(x) \mid \sigma @ t = x'\})) \\ \quad \mathcal{P}_{x'}^{\mathcal{C}}(\Phi_1 \mathsf{U}^{[0,t'-t]} \Phi_2). \end{cases}$$

3. If $[t, \infty]$, we have the similar situation as $I = [t, t']$ except the second part is a unbound until formula. That is to use embedded DTMC in this case. That is

$$\mathcal{P}_x^{\mathcal{C}}(\Phi_1 \mathsf{U}^{[t,\infty]} \Phi_2) = \begin{cases} 0, \text{ when } x \notin Sat(\Phi_1); \\ \sum_{x' \vDash \Phi_1} \mathcal{P}^{\mathcal{M}[\neg \Phi_1]}(\mathcal{C}(\{\sigma \in Path(x) \mid \sigma @ t = x'\})) \\ \quad \mathcal{P}_{x'}^{emb(\mathcal{M})}(\Phi_1 \mathsf{U}^{[0,\infty)} \Phi_2), \end{cases}$$

such that we have

$$\mathcal{P}_{x'}^{emb(\mathcal{M})}(\Phi_1 \mathsf{U}^{[0,\infty)} \Phi_2) = \begin{cases} 1, \text{ if } x' \in Sat(\Phi_2); \\ 0, \text{ if } x' \in Sat(\neg \Phi_1 \wedge \neg \Phi_2); \\ \sum_{x'' \in X} \mathbf{P}_{x',x''} \mathcal{P}_{x''}^{emb(\mathcal{M})}(\Phi_1 \mathsf{U}^{[0,\infty)} \Phi_2), \text{ otherwise.} \end{cases}$$

5.7   Summary

Stochastic model checking continuous–time Markov chains has been a hot research topic in both theory and practice for many years, such that quantitative

properties, such as performance, reliability, and efficiency, are evaluated and verified. However, there has been increasing attention that the exponentially distributed holding times in CTMCs are unrealistic in modelling many phenomena. That calls for the extension of stochastic model checking to more general distributed holding times, for instance continuous phase–type distributions. In this work, we consider the stochastic systems having holding time distributions of matrix–exponential, which contains and is algebraically equivalent with CPH distributions, however, without the probability interpretation in terms of Markov chains. For any representation of ME distributions, one could always find a representation of minimal order. This attractive property allows us to model stochastic systems using a minimal state space representation.

Apart from general probability distributions, we present how to add a second dimension to probability distributions in order to express the multiple exits followed by a randomised holding time. We illustrate this by defining a multi-exits matrix–exponential distribution, such that its joint density $f(w, g)$ interprets the probability density of a process finishing at g-th exit after w time units. Inherited from ME distributions, we show the compositionality and minimality results for MEME distributions. We then introduce a stochastic process algebra MEME to support ME distributed holding time and multiple exits, which includes several composite operators. As a unique feature of the language MEME, all the components before and after compositions are secured to a minimal state space representation. That shall ease for deploying stochastic model checking in practice.

The semantics of the language MEME is a Markov renewal process with matrix–exponential kernels, $\mathsf{MRP}_{\mathsf{MEME}}^{\mathsf{LR}}$. We then explore the stochastic model checking algorithms on $\mathsf{MRP}_{\mathsf{MEME}}^{\mathsf{LR}}$ against CSL formulae. We show that the equations systems to compute transient and limiting probabilities on $\mathsf{MRP}_{\mathsf{MEME}}^{\mathsf{LR}}$ are as standard as CTMCs, but require some extra care for choosing appropriate numerical methods (e.g. uniformisation does not work for transient probabilities). The model checking algorithms on $\mathsf{MRP}_{\mathsf{MEME}}^{\mathsf{LR}}$ are similar to CTMC model checking.

CHAPTER 6

Conclusion and Outlook

This dissertation is an interdisciplinary study involving formal methods in theoretical computer science and stochastic processes in applied probability. The aim is to develop advanced logics and models for stochastic analysis information systems, which underlying processes are possibly beyond Markov chains. Our contributions can be summaried in four important aspects:

- *Stochastic abstraction and analysis.* When analysing the performance of a complex stochastic system, often we need to abstract the model to a level that are small enough to handle. This dissertation demonstrates a high level abstraction approach to model and analyse stochastic wireless sensor networks using combinatorics. We also illustrate how discrete phase–type distributions accelerate performance evaluation of the model.

- *Compositional reasoning of purely stochastic systems.* Compositionality is an important property in system modelling, which allows a complex system to be composed by simple subsystems followed by rules. This dissertation develops new stochastic process calculus for stochastic systems in the classes of phase–type distributions and matrix–exponential distributions to reason their compositionality.

- *Characterising stochastic equivalence relations.* The study of stochastic equivalence relations is crucial in understanding the behaviours and prop-

erties for stochastic systems. This dissertation proposes time-lapse bisimulation, a new state-based bisimulation relation on Markov chains and its corresponding logical characterisation. A variety of bisimulation relations comparing to stochastic equivalence are also clarified.

- *Minimal state space representation for general distributed durations.* The state space explosion problem is a main obstacle against efficient stochastic verification. To reduce the problem of state space explosion, this thesis presents new techniques to obtain the minimal state space representation for Markov renewal processes with matrix–exponential kernels. In order to support formal verification, we illustrate stochastic model checking algorithms using continuous stochastic logics.

This dissertation has shown several interesting theoretical results, which can be naturally extended to a more general setting. The phase–type process algebra (cf. Section 3.2) explores the compositionality and algebraic properties for continuous phase–type distributions, which shall hold almost immediate to the discrete case. The minimum operator and maximum operator in discrete–time need special care, because they are a bit different from the continuous case. Moreover, the newly identified time-lapse bisimulation on labelled DTMCs (cf. Section 4.3) can be defined analogously on labelled continuous–time Markov chains, which is then a bisimulation characterisation of stochastic equivalence for continuous–time marked Markovian arrival processes.

This dissertation has discovered several modelling techniques and algorithms, which can be implemented into tools. The phase–type process algebra (cf. Section 3.2) can be implemented as a tool to parse the language into basic matrix constructions, such that one could use it to produce large-scaled absorbing continuous–time Markov chains. Even more, one could operate phase–type representations backwardly to decompose a large system into several smaller subsystems. The language MEME, the minimisation algorithm, together with corresponding stochastic model checking algorithms introduced in Chapter 5 can be implemented into a completely new stochastic model checker to support all nice features mentioned in the chapter.

This dissertation has proposed several new findings, which should be evaluated in industrial case studies to weigh pros and cons. We have shown that time-lapse bisimulation is a new contribution to the existing bisimulation family, thus there must be cases that time-lapse bisimulation outperforms with other bisimulations. It would be very interesting to see how time-lapse bisimulation can assist verification in practice. The stochastic model checking Markov renewal process with matrix–exponential kernels has the nice minimality property, it is also very interesting to know how the minimality performs when it meets the complicated models from real world.

APPENDIX A

Appendix

A.1 Definition of Function $p_i(x)$ in LMAC

The number of ordered arrangements of n objects, in which there are k_1 objects of type 1, k_2 objects of type 2, ..., and k_m objects of type m, and where $k_1 + k_2 + \cdots + k_m = n$, is given by

$$\binom{n}{k_1, k_2, \ldots, k_m} = \frac{n!}{k_1! k_2! \cdots k_m!}.$$

This number is called a *multinomial coefficient*. For instance, suppose that we have 16 sensors, and we want to put them into 6 slots, with 2, 2, 2, 3, 3, and 4 sensors respectively, then

$$\binom{16}{2, 2, 2, 3, 3, 4} = \frac{16!}{2!2!2!3!3!4!},$$

is the number of ordered ways of doing so. If we do not distinguish the sensors, we have to eliminate the ordering. That gives

$$\frac{\binom{16}{2,2,2,3,3,4}}{3!2!1!} = \frac{16!}{2!2!2!3!3!4! \times 3!2!1!}$$

is the number of non-ordered arrangements. In order to use these arguments and put them into our case, we define the following function

$$A(\mathbf{v}, g, j) = \frac{1}{\prod_{s=1}^{x} (\mathbf{v}'[s])!},$$

where $\mathbf{v}$ is an input vector with dimension x, $\mathbf{v}' = \mathbf{v} + \mathbf{e}_j + \mathbf{e}_{g-j}$, where $\mathbf{e}_j$ is a unit vector with 1 in the j-th entry, and 0 otherwise. This function corresponds to the term $\frac{1}{3!2!1!}$ in the previous example, and it is used to eliminate the ordering of any arrangement. Having the definition of function A, we will give the general form of $p_i(x)$, for $i \geq 1$.

When the collision only happens in a single slot, we have that

$$p_1(x) = \frac{1}{x!}.$$

According to the interpretation of the multinomial coefficient, we have only one slot within all the x sensors. Now, define the function

$$q_2(g, \mathbf{v}, s) = \sum_{j=s}^{\lfloor \frac{g}{2} \rfloor} \frac{1}{j!(g-j)!} A(\mathbf{v}, g, j), \quad s \geq 2,$$

which indicates that collisions can happen in two slots. The parameter g stands for the number of collided sensors. Note that in order to eliminate the ordering we call the function A. Within the summation, j is the number of sensors in the first slot, and $g - j$ is the number of sensors in the second slot. Thus, we get that $p_2(x)$ is a particular case of q_2 given by $p_2(x) = q_2(x, \mathbf{0}, 2)$.

In general, the function $q_i(g, \mathbf{v}, s)$ for $i \geq 3$, indicates that there are i slots with collisions. It is calculated using recursive calls as follows

$$q_i(g, \mathbf{v}, s) = \sum_{j=s}^{\lfloor \frac{g}{i} \rfloor} \frac{1}{j!} q_{i-1}(g - j, \mathbf{v} + \mathbf{e}_j, j), \quad s \geq 2.$$

We compute this function by putting j sensors into the first slot (corresponding to the term $\frac{1}{j!}$), and putting the others $(g - j)$ sensors into $(i - 1)$ slots, which we can get it from the previous q_{i-1}.

Thus, the general form of $p_i(x)$ for $i \geq 3$, is given by

$$p_i(x) = \sum_{j=2}^{\lfloor \frac{x}{i} \rfloor} \frac{1}{j!} q_{i-1}(x - j, \mathbf{e}_j, j), \quad i \geq 3.$$

A.2   Kronecker Operations

Many operations with phase type distributions are conveniently expressed using the Kronecker product $\otimes$. For two matrices $\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1k} \\ a_{21} & a_{22} & \ldots & a_{2k} \\ \vdots & \vdots & \ldots & \vdots \\ a_{l1} & a_{l2} & \ldots & a_{lk} \end{bmatrix}$ with

dimension $l \times k$ and $\boldsymbol{B} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ldots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}$ with dimension $m \times n$, the

Kronecker product $\otimes$ is defined by

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}B & a_{12}B & \ldots & a_{1k}B \\ a_{21}B & a_{22}B & \ldots & a_{2k}B \\ \vdots & \vdots & \ldots & \vdots \\ a_{l1}B & a_{l2}B & \ldots & a_{lk}B \end{bmatrix}$$

with dimension $lm \times kn$.

There are some important properties of Kronecker product.

- Associativity:
  $\boldsymbol{A} \otimes (\boldsymbol{B} \otimes \boldsymbol{C}) = (\boldsymbol{A} \otimes \boldsymbol{B}) \otimes \boldsymbol{C}$

- No Commutativity:
  $\boldsymbol{A} \otimes \boldsymbol{B} \neq \boldsymbol{B} \otimes \boldsymbol{A}$ in general

- Distributivity over ordinary matrix addition:
  $(\boldsymbol{A}_1 + \boldsymbol{B}_1) \otimes (\boldsymbol{A}_2 + \boldsymbol{B}_2) = \boldsymbol{A}_1 \otimes \boldsymbol{A}_2 + \boldsymbol{A}_1 \otimes \boldsymbol{B}_2 + \boldsymbol{B}_1 \otimes \boldsymbol{A}_2 + \boldsymbol{B}_1 \otimes \boldsymbol{B}_2$

- Compatibility with ordinary matrix multiplication I:
  $(\boldsymbol{A}_1 \otimes \boldsymbol{B}_1) \times (\boldsymbol{A}_2 \otimes \boldsymbol{B}_2) = (\boldsymbol{A}_1 \times \boldsymbol{A}_2) \otimes (\boldsymbol{B}_1 \times \boldsymbol{B}_2)$

- Compatibility with ordinary matrix multiplication II:
  $(\boldsymbol{A}_1 \boldsymbol{A}_2 \boldsymbol{A}_3) \otimes (\boldsymbol{B}_1 \boldsymbol{B}_2 \boldsymbol{B}_3) = (\boldsymbol{A}_1 \otimes \boldsymbol{B}_1)(\boldsymbol{A}_2 \otimes \boldsymbol{B}_2)(\boldsymbol{A}_3 \otimes \boldsymbol{B}_3)$

- Compatibility with ordinary matrix inversion:
  $(\boldsymbol{A} \otimes \boldsymbol{B})^{-1} = \boldsymbol{B}^{-1} \otimes \boldsymbol{A}^{-1}$

The Kronecker sum $\oplus$ of two matrics $\boldsymbol{A}$ and $\boldsymbol{B}$ is defined by

$$\boldsymbol{A} \oplus \boldsymbol{B} = \boldsymbol{A} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{B},$$

where I is the identity matrix.

The Kronecker sum has one important rule

$$exp(\boldsymbol{A}_1) \otimes exp(\boldsymbol{A}_2) = exp(\boldsymbol{A}_1 \oplus \boldsymbol{A}_2).$$

A.3 Proof: Distributive Properties for CPH composite operators

In this section, we proof for CPH closed operators (convolution $\sum$, minimum min and maximum max) that there is no distributivity holds outside Table 3.2 of operational rules.

For the binary convex mixture operator $+$. Since $+$ always assign probability p on the left side and $q = 1 - p$ on the right side. Each binary $+$ operation adds in one parameter p to the initial probability vector, i.e. $(p\boldsymbol{\alpha}_1, q\boldsymbol{\alpha}_2)$. Therefore, in general it is impossible to have any CPH closed operator, denoted by $\cdot$, which is distributive over $+$. To see this, let us assume distributivity holds, then

$$(W_{1(\boldsymbol{\alpha}_1, \mathbf{T}_1)} \cdot W_{2(\boldsymbol{\alpha}_2, \mathbf{T}_2)}) + W_{3(\boldsymbol{\alpha}_3, \mathbf{T}_3)} = (W_{1(\boldsymbol{\alpha}_1, \mathbf{T}_1)} + W_{3(\boldsymbol{\alpha}_3, \mathbf{T}_3)}) \cdot (W_{2(\boldsymbol{\alpha}_2, \mathbf{T}_2)} + W_{3(\boldsymbol{\alpha}_3, \mathbf{T}_3)}).$$

It has one pair of p and q on the final initial probability vector according to one appearance of $+$ on the left side of above formula, however there are two different pairs of p and q for the right side according to two appearances of $+$ operators. Therefore, the resulted initial probability vectors cannot agree, which implies that $\sum$, min and max are not distributive over $+$.

The counterexamples we employed in our later proofs are instances of exponential distributions. With the counterexamples, we show no distributivity among $\sum$, min and max holds. Without losing generality, we have three CPH distributed random variables $W_{1(1,[-\lambda])}$, $W_{2(1,[-\mu])}$, and $W_{3(1,[-\gamma])}$.

PROPOSITION A.1 *The convolution $\sum$ is not distributive over minimum* min *and maximum* max.

PROOF For $\sum$ is not distributive over min:

$$(W_{1\,(1,[-\lambda])} \sum W_{2\,(1,[-\mu])}) \min W_{3\,(1,[-\gamma])} = W'_{\left((1,0),\begin{bmatrix} -\lambda & \lambda \\ 0 & -\mu \end{bmatrix}\right)} \min W_{3\,(1,[-\gamma])}$$

$$= \bar{W}_{\left((1,0),\begin{bmatrix} -\lambda-\gamma & \lambda \\ 0 & -\mu-\gamma \end{bmatrix}\right)},$$

such that LST of above CPH distributed random variable is

$$H(s) = 0 + [1,0]\left(s\boldsymbol{I} - \begin{bmatrix} -\lambda-\gamma & \lambda \\ 0 & -\mu-\gamma \end{bmatrix}\right)^{-1}\begin{bmatrix} \gamma \\ \mu+\gamma \end{bmatrix}$$

$$= [1,0]\left(\begin{bmatrix} s+\lambda+\gamma & -\lambda \\ 0 & s+\lambda+\gamma \end{bmatrix}\right)^{-1}\begin{bmatrix} \gamma \\ \mu+\gamma \end{bmatrix}$$

$$= [1,0]\begin{bmatrix} \frac{1}{s+\lambda+\gamma} & \frac{\lambda}{(s+\lambda+\gamma)(s+\mu+\gamma)} \\ 0 & \frac{1}{s+\mu+\gamma} \end{bmatrix}\begin{bmatrix} \gamma \\ \mu+\gamma \end{bmatrix}$$

$$= \frac{\gamma(s+\mu+\gamma) + \lambda(\mu+\gamma)}{(s+\lambda+\gamma)(s+\mu+\gamma)}.$$

On the other side, we have

$$(W_{1\,(1,[-\lambda])} \min W_{3\,(1,[-\gamma])}) \sum (W_{2\,(1,[-\mu])} \min W_{3\,(1,[-\gamma])})$$

$$= W'_{\left(1,[-\lambda-\gamma]\right)} \sum W''_{\left(1,[-\mu-\gamma]\right)}$$

$$= \bar{W}_{\left((1,0),\begin{bmatrix} -\lambda-\gamma & \lambda+\gamma \\ 0 & -\mu-\gamma \end{bmatrix}\right)},$$

such that the LST of above CPH distributed random variable is

$$H(s) = 0 + [1,0]\left(s\boldsymbol{I} - \begin{bmatrix} -\lambda-\gamma & \lambda+\gamma \\ 0 & -\mu-\gamma \end{bmatrix}\right)^{-1}\begin{bmatrix} 0 \\ \mu+\gamma \end{bmatrix}$$

$$= [1,0]\left(\begin{bmatrix} s+\lambda+\gamma & -\lambda-\gamma \\ 0 & s+\mu+\gamma \end{bmatrix}\right)^{-1}\begin{bmatrix} 0 \\ \mu+\gamma \end{bmatrix}$$

$$= [1,0]\begin{bmatrix} \frac{1}{s+\lambda+\gamma} & \frac{\lambda+\gamma}{(s+\lambda+\gamma)(s+\mu+\gamma)} \\ 0 & \frac{1}{s+\mu+\gamma} \end{bmatrix}\begin{bmatrix} 0 \\ \mu+\gamma \end{bmatrix}$$

$$= \frac{(\lambda+\gamma)(\mu+\gamma)}{(s+\lambda+\gamma)(s+\mu+\gamma)}.$$

Two LSTs are not identical, thus we show $\sum$ is not distributive over min.

For $\sum$ is not distributive over max:

$$(W_{1\,(1,[-\lambda])} \sum W_{2\,(1,[-\mu])}) \max W_{3\,(1,[-\gamma])}$$

$$= W'_{\left((1,0),\begin{bmatrix} -\lambda & \lambda \\ 0 & -\mu \end{bmatrix}\right)} \max W_{3\,(1,[-\gamma])}$$

$$= \bar{W}_{\left((1,0,0,0,0),\begin{bmatrix} -\lambda-\gamma & \lambda & \gamma & 0 & 0 \\ 0 & -\mu-\gamma & 0 & \gamma & \mu \\ 0 & 0 & -\lambda & \lambda & 0 \\ 0 & 0 & 0 & -\mu & 0 \\ 0 & 0 & 0 & 0 & -\gamma \end{bmatrix}\right)},$$

and

$$(W_{1\,(1,[-\lambda])} \max W_{3\,(1,[-\gamma])}) \sum (W_{2\,(1,[-\mu])} \max W_{3\,(1,[-\gamma])})$$

$$= W'_{\left((1,0,0),\begin{bmatrix} -\lambda-\gamma & \gamma & \lambda \\ 0 & -\lambda & 0 \\ 0 & 0 & -\gamma \end{bmatrix}\right)} \sum W''_{\left((1,0,0),\begin{bmatrix} -\mu-\gamma & \gamma & \mu \\ 0 & -\mu & 0 \\ 0 & 0 & -\gamma \end{bmatrix}\right)}$$

$$= \bar{W}_{\left((1,0,0,0,0,0),\begin{bmatrix} -\lambda-\gamma & \gamma & \lambda & 0 & 0 & 0 \\ 0 & -\lambda & 0 & \lambda & 0 & 0 \\ 0 & 0 & -\gamma & \gamma & 0 & 0 \\ 0 & 0 & 0 & -\mu-\gamma & \gamma & \mu \\ 0 & 0 & 0 & 0 & -\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & -\gamma \end{bmatrix}\right)}.$$

Comparing two above LSTs directly are computational heavy. Therefore, we look into numerical case by comparing the first order moment (cf. Formula (2.13)). The case we took is having that λ, μ, γ all equals 1. That gives the first random variable is 2.25 and the second random variable is 3. That is the counterexample, thus we conclude that $\sum$ is not distributive over max. □

PROPOSITION A.2 *The minimum* min *is not distributive over convolution* $\sum$ *and maximum* max.

PROOF For min operator is not distributive over $\sum$:

$$(W_{1\,(1,[-\lambda])} \min W_{2\,(1,[-\mu])}) \sum W_{3\,(1,[-\gamma])} = W'_{(1,[-\lambda-\mu])} \sum W_{3\,(1,[-\gamma])}$$

$$= \bar{W}_{\left((1,0),\begin{bmatrix} -\lambda-\mu & \lambda+\mu \\ 0 & -\gamma \end{bmatrix}\right)},$$

such that the LST of above CPH distributed random variable is

$$H(s) = 0 + [1,0] \left(s\boldsymbol{I} - \begin{bmatrix} -\lambda - \mu & \lambda + \mu \\ 0 & -\gamma \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ \gamma \end{bmatrix}$$

$$= [1,0] \left(\begin{bmatrix} s + \lambda + \mu & -\lambda - \mu \\ 0 & s + \gamma \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ \gamma \end{bmatrix}$$

$$= [1,0] \begin{bmatrix} \frac{1}{s+\lambda+\mu} & \frac{\lambda+\mu}{(s+\lambda+\mu)(s+\gamma)} \\ 0 & \frac{1}{s+\gamma} \end{bmatrix} \begin{bmatrix} 0 \\ \gamma \end{bmatrix} = \frac{\gamma(\lambda + \mu)}{(s + \lambda + \mu)(s + \gamma)}.$$

On the other side, we have

$$(W_{1(1,[-\lambda])} \sum W_{3(1,[-\gamma])}) \min(W_{2(1,[-\mu])} \sum W_{3(1,[-\gamma])})$$

$$= W' \left({}_{(1,0),} \begin{bmatrix} -\lambda & \lambda \\ 0 & -\gamma \end{bmatrix} \right) \min W'' \left({}_{(1,0),} \begin{bmatrix} -\mu & \mu \\ 0 & -\gamma \end{bmatrix} \right)$$

$$= \bar{W} \left({}_{(1,0,0,0),} \begin{bmatrix} -\lambda - \mu & \mu & \lambda & 0 \\ 0 & -\lambda - \gamma & 0 & \lambda \\ 0 & 0 & -\mu - \gamma & \mu \\ 0 & 0 & 0 & -2\gamma \end{bmatrix} \right),$$

such that the LST of above CPH distributed random variable is

$$H(s) = 0 + [1,0,0,0] \left(s\boldsymbol{I} - \begin{bmatrix} -\lambda - \mu & \mu & \lambda & 0 \\ 0 & -\lambda - \gamma & 0 & \lambda \\ 0 & 0 & -\mu - \gamma & \mu \\ 0 & 0 & 0 & -2\gamma \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ \gamma \\ \gamma \\ 2\gamma \end{bmatrix}$$

$$= [1,0,0,0] \left(\begin{bmatrix} s + \lambda + \mu & -\mu & -\lambda & 0 \\ 0 & s + \lambda + \gamma & 0 & -\lambda \\ 0 & 0 & s + \mu + \gamma & -\mu \\ 0 & 0 & 0 & s + 2\gamma \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ \gamma \\ \gamma \\ 2\gamma \end{bmatrix}$$

$$= \frac{\mu\gamma}{(s + \lambda + \gamma)(s + \lambda + \mu)} + \frac{\lambda\gamma}{(s + \mu + \gamma)(s + \lambda + \mu)} +$$

$$\frac{2\lambda\mu\gamma(2s + \mu + 2\gamma + \lambda)}{(s + \lambda + \mu)(s + \lambda + \gamma)(s + \mu + \gamma)(s + 2\gamma)}.$$

Two LSTs are not equal, thus the distributivity does not hold.

For min operator is not distributive over max: we have

$$(W_{1(1,[-\lambda])} \min W_{2(1,[-\mu])}) \max W_{3(1,[-\gamma])}$$
$$= W'_{(1,[-\lambda-\mu])} \max W_{3(1,[-\gamma])}$$
$$= \bar{W}_{\left((1,0,0), \begin{bmatrix} -\lambda-\mu-\gamma & \gamma & \lambda+\mu \\ 0 & -\lambda-\mu & 0 \\ 0 & 0 & -\gamma \end{bmatrix}\right)},$$

and

$$(W_{1(1,[-\lambda])} \max W_{3(1,[-\gamma])}) \min(W_{2(1,[-\mu])} \max W_{3(1,[-\gamma])})$$
$$= W'_{\left((1,0,0), \begin{bmatrix} -\lambda-\gamma & \gamma & \lambda \\ 0 & -\lambda & 0 \\ 0 & 0 & -\gamma \end{bmatrix}\right)} \min W''_{\left((1,0,0), \begin{bmatrix} -\mu-\gamma & \gamma & \mu \\ 0 & -\mu & 0 \\ 0 & 0 & -\gamma \end{bmatrix}\right)}$$
$$= \bar{W}_{\left((1,0,0,0,0,0,0,0,0),\right.}$$
$$-\begin{bmatrix} \lambda+\mu+2\gamma & -\gamma & -\mu & -\gamma & 0 & 0 & -\lambda & 0 & 0 \\ 0 & \lambda+\mu+\gamma & 0 & 0 & -\gamma & 0 & 0 & -\lambda & 0 \\ 0 & 0 & \lambda+2\gamma & 0 & 0 & -\gamma & 0 & 0 & -\lambda \\ 0 & 0 & 0 & \lambda+\mu+\gamma & -\gamma & -\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda+\mu & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda+\gamma & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mu+2\gamma & -\gamma & -\mu \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu+\gamma & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\gamma \end{bmatrix}\right).$$

To obtain LSTs is too complicated, therefore we compute the mean of two CPH representations for a special case such that λ, μ, γ all equals 1. In this case, we have the first random variable has the mean equals to 1.1667 and the second one is 0.9167. The counterexample is found, and these two CPH representations must not agree. $\qquad\square$

PROPOSITION A.3 *The maximum* max *is not distributive over convolution* $\sum$ *and minimum* min.

PROOF For max operator is not distributive over $\sum$: we have

$$
(W_{1\,(1,[-\lambda])} \max W_{2\,(1,[-\mu])}) \sum W_{3\,(1,[-\gamma])}
$$

$$
= W'_{\left((1,0,0),\begin{bmatrix} -\lambda-\mu & \mu & \lambda \\ 0 & -\lambda & 0 \\ 0 & 0 & -\mu \end{bmatrix}\right)} \sum W_{3\,(1,[-\gamma])}
$$

$$
= \bar{W}_{\left((1,0,0,0),\begin{bmatrix} -\lambda-\mu & \mu & \lambda & 0 \\ 0 & -\lambda & 0 & \lambda \\ 0 & 0 & -\mu & \mu \\ 0 & 0 & 0 & -\gamma \end{bmatrix}\right)},
$$

and

$$
(W_{1\,(1,[-\lambda])} \sum W_{3\,(1,[-\gamma])}) \max (W_{2\,(1,[-\mu])} \sum W_{3\,(1,[-\gamma])})
$$

$$
= W'_{\left((1,0),\begin{bmatrix} -\lambda & \lambda \\ 0 & -\gamma \end{bmatrix}\right)} \max W''_{\left((1,0),\begin{bmatrix} -\mu & \mu \\ 0 & -\gamma \end{bmatrix}\right)}
$$

$$
= \bar{W}_{\left((1,0,0,0,0,0,0,0),\begin{bmatrix} -\lambda-\mu & \mu & \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & -\lambda-\gamma & 0 & \lambda & \gamma & 0 & 0 & 0 \\ 0 & 0 & -\mu-\gamma & \mu & 0 & 0 & \gamma & 0 \\ 0 & 0 & 0 & -2\gamma & 0 & \gamma & 0 & \gamma \\ 0 & 0 & 0 & 0 & -\lambda & \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\gamma & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\mu & \mu \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\gamma \end{bmatrix}\right)}.
$$

Let λ, μ, γ all equal 1, we have the mean of first random variable is 2.5 and the mean of the second is 2.75. Thus, these two CPH distributed random variable must not agree.

For maximum is not distributive over the minimum: we have

$$
(W_{1\,(1,[-\lambda])} \max W_{2\,(1,[-\mu])}) \min W_{3\,(1,[-\gamma])}
$$

$$
= W'_{\left((1,0,0),\begin{bmatrix} -\lambda-\mu & \mu & \lambda \\ 0 & -\lambda & 0 \\ 0 & 0 & -\mu \end{bmatrix}\right)} \min W_{3\,(1,[-\gamma])}
$$

$$
= \bar{W}_{\left((1,0,0),\begin{bmatrix} -\lambda-\mu-\gamma & \mu & \lambda \\ 0 & -\lambda-\gamma & 0 \\ 0 & 0 & -\mu-\gamma \end{bmatrix}\right)},
$$

such that the LST of above CPH distributed random variable is

$$H(s) = 0 + [1,0,0] \left(s\boldsymbol{I} - \begin{bmatrix} -\lambda - \mu - \gamma & \mu & \lambda \\ 0 & -\lambda - \gamma & 0 \\ 0 & 0 & -\mu - \gamma \end{bmatrix} \right)^{-1} \begin{bmatrix} \gamma \\ \lambda + \gamma \\ \mu + \gamma \end{bmatrix}$$

$$= [1,0] \left(\begin{bmatrix} s + \lambda + \mu + \gamma & -\mu & -\lambda \\ 0 & s + \lambda + \gamma & 0 \\ 0 & 0 & s + \mu + \gamma \end{bmatrix} \right)^{-1} \begin{bmatrix} \gamma \\ \lambda + \gamma \\ \mu + \gamma \end{bmatrix}$$

$$= [1,0] \begin{bmatrix} \frac{1}{s+\lambda+\mu+\gamma} & \frac{\mu}{(s+\lambda+\mu+\gamma)(s+\lambda+\gamma)} & \frac{(\lambda+\gamma)^2}{(s+\lambda+\mu+\gamma)(s+\lambda+\gamma)(s+\mu+\gamma)} \\ 0 & \frac{1}{s+\lambda+\gamma} & 0 \\ 0 & 0 & \frac{1}{s+\mu+\gamma} \end{bmatrix} \begin{bmatrix} \gamma \\ \lambda + \gamma \\ \mu + \gamma \end{bmatrix}$$

$$= \frac{\gamma(s+\lambda+\gamma)(s+\mu+\gamma) + \mu(s+\mu+\gamma)(\lambda+\gamma) + (\lambda+\gamma)^2(\mu+\gamma)}{(s+\lambda+\mu+\gamma)(s+\lambda+\gamma)(s+\mu+\gamma)}.$$

On the other side, we have

$$(W_{1(1,[-\lambda])} \min W_{3(1,[-\gamma])}) \max(W_{2(1,[-\mu])} \min W_{3(1,[-\gamma])})$$

$$= W'_{\left(1,[-\lambda-\gamma]\right)} \max W''_{\left(1,[-\mu-\gamma]\right)}$$

$$= \bar{W}_{\left((1,0,0), \begin{bmatrix} -\lambda-\mu-2\gamma & \mu+\gamma & \lambda+\gamma \\ 0 & -\lambda-\gamma & 0 \\ 0 & 0 & -\mu-\gamma \end{bmatrix}\right)},$$

such that the LST of above CPH distributed random variable is

$$H(s) = 0 + [1,0,0] \left(s\boldsymbol{I} - \begin{bmatrix} -\lambda - \mu - 2\gamma & \mu + \gamma & \lambda + \gamma \\ 0 & -\lambda - \gamma & 0 \\ 0 & 0 & -\mu - \gamma \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ \lambda + \gamma \\ \mu + \gamma \end{bmatrix}$$

$$= [1,0,0] \left(\begin{bmatrix} s + \lambda + \mu + 2\gamma & -\mu - \gamma & -\lambda - \gamma \\ 0 & s + \lambda + \gamma & 0 \\ 0 & 0 & s + \mu + \gamma \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ \lambda + \gamma \\ \mu + \gamma \end{bmatrix}$$

$$= [1,0,0] \begin{bmatrix} \frac{1}{s+\lambda+\mu+2\gamma} & \frac{\mu+\gamma}{(s+\lambda+\mu+2\gamma)(s+\lambda+\gamma)} & \frac{(\lambda+\gamma)}{(s+\lambda+\mu+2\gamma)(s+\mu+\gamma)} \\ 0 & \frac{1}{s+\lambda+\gamma} & 0 \\ 0 & 0 & \frac{1}{s+\mu+\gamma} \end{bmatrix} \begin{bmatrix} 0 \\ \lambda + \gamma \\ \mu + \gamma \end{bmatrix}$$

$$= \frac{(\lambda+\gamma)(\mu+\gamma)(s+\mu+\gamma) + (\lambda+\gamma)(\mu+\gamma)(s+\lambda+\gamma)}{(s+\lambda+\mu+2\gamma)(s+\lambda+\gamma)(s+\mu+\gamma)}.$$

By evaluating LSTs, we conclude the distributivity does not hold. □

Bibliography

[AB97]   S. Asmussen and M. Bladt. Renewal theory and queueing algorithms
          for Matrix-Exponential distributions. *Matrix-analytic methods in
          stochastic models*, pages 313–341, 1997.

[AB99]   S. Asmussen and M. Bladt. Point processes with finite-dimensional
          conditional probabilities. *Stochastic process and their applications*,
          82:127–142, 1999.

[AILS07] Luca Aceto, Anna Ingólfsdóttir, Kim Guldstrand Larsen, and Jiri
          Srba. *Reactive Systems: Modelling, Specification and Verification.*
          Cambridge University Press, New York, NY, USA, 2007.

[AL82]   David Assaf and Benny Levikson. Closure of phase type distribu-
          tions under operations arising in reliability theory. *The Annals of
          Probability*, 10(1):pp. 265–269, 1982.

[AM69]   M. F. Atiyah and I. G. MacDonald. *Introduction to Commutative
          Algebra.* Westview Press, 1969.

[AN95]   A. S. Alfa and M. F. Neuts. Modelling Vehicular Traffic Using the
          Discrete Time Markovian Arrival Process. *Transportation Science*,
          29:109–117, 1995.

[ANO96]   S. Asmussen, O. Nerman, and M. Olsson. Fitting phase-type distri-
          butions via the EM algorihtm. *Scandinavian Journal of Statistics*,
          23:419–441, 1996.

[Asm92]   S. Asmussen. Phase-type representations in random walk and queue-
          ing problems. *The Annals of Probability*, 20:772–789, 1992.

[Asm00]   S. Asmussen. Matrix-analytic models and their analysis. *Scandinavian Journal of Statistics*, 27:193–226, 2000.

[ASSB96]   Adnan Aziz, Kumud Sanwal, Vigyan Singhal, and Robert Brayton. Verifying continuous time Markov chains. In Rajeev Alur and Thomas Henzinger, editors, *Computer Aided Verification*, volume 1102 of *Lecture Notes in Computer Science*, pages 269–276. Springer Berlin / Heidelberg, 1996.

[BB12]   P. Buchholz and M. Buchholz. Rational processes related to communicating Markov processes. *Journal of Applied Probability*, 49(1):40–59, 2012.

[BDHK03]   Jeremy T. Bradley, Nicholas J. Dingle, Peter G. Harrison, and William J. Knottenbelt. Performance queries on semi-Markov stochastic Petri nets with an extended Continuous Stochastic Logic. In *in PNPM 03, Proceedings of Petri Nets and Performance Models*, pages 62–71. IEEE Computer Society, 2003.

[BDL04]   Gerd Behrmann, Alexandre David, and Kim G. Larsen. A tutorial on uppaal. *Formal Methods for the Design of Real-Time Systems: 4th International School on Formal Methods for the Design of Computer, Communication, and Software Systems (SFM-RT 2004)*, pages 200–236, 2004.

[BG96]   Marco Bernardo and Roberto Gorrieri. Extended Markovian Process Algebra. In *CONCUR '96: Concurrency Theory*, volume 1119 of *Lecture Notes in Computer Science*, pages 315–330. Springer Berlin / Heidelberg, 1996.

[BH97]   Christel Baier and Holger Hermanns. Weak bisimulation for fully probabilistic processes. In Orna Grumberg, editor, *Computer Aided Verification*, volume 1254 of *Lecture Notes in Computer Science*, pages 119–130. Springer Berlin / Heidelberg, 1997.

[BHHK03]   C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *Software Engineering, IEEE Transactions on*, 29(6):524 – 541, june 2003.

[BKK10]   Peter Buchholz, Peter Kemper, and Jan Kriege. Multi-class Markovian arrival processes and their parameter fitting. *Performance Evaluation*, 67(11):1092 – 1106, 2010.

[Blo93]   C. Blondia. A Discrete-time Batch Markovian Arrival Process as B-ISDN Traffic model. *Belgian Journal of Operations Research*, 1993.

[BN03]   M. Bladt and M. F. Neuts. Matrix-exponential distributions: cal-
          culus and interpretations via flows. *Stochastic Models*, 19:113–124,
          2003.

[BN10]   N. G. Bean and B. F. Nielsen. Quasi-Birth-and-Death Process with
          Rational Arrival Process Components. *Stochastic Models*, 26:309–
          334, 2010.

[BN11]   Nigel Bean and Bo Friis Nielsen. Analysis of Queues with Ratio-
          nal Arrival Process Components - A General Approach. Techni-
          cal Report IMM-Technical Report-2011-05, DTU Informatics, Kgs.
          Lyngby, Denmark, 2011.

[Bot86]   Robert Franklin Botta. *Sochastic Modeling with Generalized Expo-
          nential Mixtures*. PhD thesis, University of Virginia, 1986.

[BT11a]   P. Buchholz and M. Telek. On minimal representations of Rational
          Arrival Processes. *Annals of Operations Research*, pages 1–24, 2011.

[BT11b]   Peter Buchholz and Miklós Telek. Composition and Equivalence
          of Markovian and Non-Markovian Models. In *Proceedings of the
          2011 Eighth International Conference on Quantitative Evaluation of
          SysTems*, QEST '11, pages 213–222, Washington, DC, USA, 2011.
          IEEE Computer Society.

[CC77]   Patrick Cousot and Radhia Cousot. Abstract interpretation: a
          unified lattice model for static analysis of programs by construc-
          tion or approximation of fixpoints. In *Proceedings of the 4th ACM
          SIGACT-SIGPLAN symposium on Principles of programming lan-
          guages*, POPL '77, pages 238–252, New York, NY, USA, 1977. ACM.

[CGL94]   Edmund M. Clarke, Orna Grumberg, and David E. Long. Model
          checking and abstraction. *ACM Trans. Program. Lang. Syst.*,
          16(5):1512–1542, September 1994.

[CNI04]   T.K. Christensen, Bo Friis Nielsen, and Villy Baek Iversen. Phase-
          type models of channel-holding times in cellular communication sys-
          tems. *Vehicular Technology, IEEE Transactions on*, 53(3):725 – 733,
          may 2004.

[CP09]   Pepijn Crouzen and Reza Pulungan. Acyclic phase-type distribu-
          tions in fault trees. In *Proceedings of the International Workshop on
          Performability Modeling of Computer and Communication Systems
          (PMCCS)*, 2009.

[CW90]   Don Coppersmith and Shmuel Winograd. Matrix multiplication via
          arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251
          – 280, 1990.

[DEP98]   J. Desharnais, A. Edalat, and P. Panangaden. A logical charac-
           terization of bisimulation for labeled Markov processes. In *Logic
           in Computer Science, 1998. Proceedings. Thirteenth Annual IEEE
           Symposium on*, pages 478–487, jun 1998.

[DVJ88]   D.J Daley and D. Vere-Jones. *An Introduction to the Theory of
           Point Processes*. Springer, New York, 1988.

[ERKN99]   Amani El-Rayes, Marta Kwiatkowska, and Gethin Norman. Solving
           infinite stochastic process algebra models through matrix-geometric
           methods. In *Proc. 7th Process Algebras and Performance Modelling
           Workshop (PAPM'99)*, pages 41–62, 1999.

[EZN11]   Luz R. Esparza, K. Zeng, and B. F. Nielsen. A Probabilistic Model
           of the LMAC Protocol for Concurrent Wireless Sensor Networks. In
           *The 11th international conference on application of concurrency to
           system design (ACSD)*, pages 98–107, 2011.

[Fel68]   William Feller. *An Introduction to Probability Theory and Its Ap-
           plications*. Wiley, 3 edition, January 1968.

[FvHM07]   Ansgar Fehnker, Lodewijk van Hoesel, and Angelika Mader. Mod-
           elling and verification of the lmac protocol for wireless sensor net-
           works. In *Integrated Formal Methods (IFM 2007)*, volume 4591 of
           LNCS, pages 253–272. Springer, 2007.

[GHR92]   N. Gotz, U. Herzog, and M. Rettelbach. Tipp – a language for
           timed processes and performance evaluation. report, University of
           Erlangen-Nurnberg, 1992.

[GS01]   Geoffrey Grimmett and David R. Stirzaker. *Probability and Random
           Processes*. Oxford university press, third edition, 2001.

[HB86]   Pieter Jacobus Houwen Hermann Brunner. *The numerical solution
           of Volterra equations*. North Holland, 1986.

[Her02]   Holger Hermanns. *Interactive Markov Chains: The Quest for Quan-
           tified Quality*, volume 2428 of *Lecture Notes in Computer Science*.
           Springer, 2002.

[Hil96]   Jane Hillston. *A compositional approach to performance modelling*.
           Cambridge University Press, New York, NY, USA, 1996.

[HN98]   Q. M. He and M. F. Neuts. Markov chains with marked transitions.
           Stochastic process and their applications, 74:37–52, 1998.

[HT02]   A. Horváth and M. Telek. PhFit: A general phase-type fitting tool.
           *Proceedings International Conference on Dependable Systems and
           Networks*, page 543, 2002.

[HZ07]   Q. He and H. Zhang. On matrix-exponential distributions. *Advances in Applied Probability*, 39:271–292, 2007.

[JS90]   Chi-Chang Jou and ScottA. Smolka. Equivalences, congruences, and complete axiomatizations for probabilistic processes. In J.C.M. Baeten and J.W. Klop, editors, *CONCUR '90 Theories of Concurrency: Unification and Extension*, volume 458 of *Lecture Notes in Computer Science*, pages 367–383. Springer Berlin Heidelberg, 1990.

[JT88]   M. Johnson and M. Taaffe. The denseness of phase distributions. Research memorandum, Purdue University, 1988.

[KNP07]   Marta Kwiatkowska, Gethin Norman, and David Parker. Stochastic model checking. In Marco Bernardo and Jane Hillston, editors, *Formal Methods for Performance Evaluation*, volume 4486 of *Lecture Notes in Computer Science*, pages 220–270. Springer Berlin / Heidelberg, 2007.

[KT75]   S. Karlin and H.E. Taylor. *A first course in stochastic processes*. Academic press, 1975.

[LHK01]   Gabriel Lopez, Holger Hermanns, and Joost-Pieter Katoen. Beyond Memoryless Distributions: Model Checking Semi-Markov Chains. In Luca de Alfaro and Stephen Gilmore, editors, *Process Algebra and Probabilistic Methods. Performance Modelling and Verification*, volume 2165 of *Lecture Notes in Computer Science*, pages 57–70. Springer Berlin / Heidelberg, 2001.

[LN91]   D. Liu and M. F. Neuts. Counter-examples involving Markovian arrival processes. *Stochastic models*, 7(3):199–509, 1991.

[LS91]   Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.

[Luc91]   D. M. Lucatoni. New results on the single server queue with a batch Markovian arrival process. *Stochastic Models*, 7:1–46, 1991.

[MW08]   Thomas Moscibroda and Roger Wattenhofer. Coloring unstructured radio networks. *Distributed Computing*, 21(4):271–284, 2008.

[Neu75]   M. F. Neuts. Probability distributions of phase-type. *In Liber Amicorum Prof. Emeritus H. Florin*, pages 173–206, 1975.

[Neu79]   M. F. Neuts. A versatile Markovian point process. *Journal of Applied Probability*, 16:764–779, 1979.

[Neu81]   M. F. Neuts. *Matrix-geometric solutions in stochastic models*, volume 2. Johns Hopkins University Press, Baltimore, Md., 1981.

[Neu89]   M. F. Neuts. *Structured stochastic matrices of the M/G/1 type and their applications*, volume 5 of *Probability: pure and applied*. Marcel Dekker Inc, New York, 1989.

[NH83]   R. De Nicola and M. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1983.

[Nie98]   B. F. Nielsen. *Note on Markovian Arrival process for 04141 Stochastic Process*, 1998.

[NNN10]   B. F. Nielsen, F. Nielson, and H. R. Nielson. Model checking multivariate state rewards. *QEST*, 17:7–16, 2010.

[O'C89]   C. A. O'Cinneide. On non-uniqueness of representations of Phase-type distributions. *Stochastic models*, 5:247–259, 1989.

[O'C90]   C. A. O'Cinneide. Characterization of phase-type distributions. *Comm. Statist. Stochastic Models*, 6:1–57, 1990.

[Pan09]   Prakash Panangaden. *Labelled Markov Processes.* Imperial College Press, 2009.

[Plo81]   G. D. Plotkin. A structural approach to operational semantics, 1981.

[Pri95]   C. Priami. Stochastic pi-calculus. *The Computer Journal*, 38(7):578–589, 1995.

[PS07]   Augusto Parma and Roberto Segala. Logical characterizations of bisimulations for discrete probabilistic systems. In Helmut Seidl, editor, *Foundations of Software Science and Computational Structures*, volume 4423 of *Lecture Notes in Computer Science*, pages 287–301. Springer Berlin / Heidelberg, 2007.

[Pul09]   Reza Pulungan. *Reduction of Acyclic Phase-Type Representations.* PhD thesis, Universität des Saarlandes, 2009.

[Ros96]   S. M. Ross. *Stochastic processes.* John Wiley & Sons, Inc, University of California, Berkeley, second edition, 1996.

[Ste94]   William J. Stewart. *Introduction to the numerical solution of Markov chains.* Princeton University Press, 1994.

[Str88]   Gilbert Strang. *Linear Algebra and Its Applications.* Addison Wesley, 3rd edition, 1988.

[Tar71]   Robert Tarjan. Depth-first search and linear graph algorithms. In *Switching and Automata Theory, 1971., 12th Annual Symposium on*, pages 114 –121, oct. 1971.

[TB97]   Lloyd N. Trefethen and David Bau. *Numerical linear algebra*. Society for Industrial Mathematics, 1997.

[Tel11]   M. Telek. Some Structural Properties of Markov and Rational Arrival Processes. *Stochastic Models*, 27(4):723–746, 2011.

[TH07]   M. Telek and G. Horváth. A minimal representation of Markov arrival processes and a moments matching method. *Perform. Eval.*, 64:1153–1168, 2007.

[TSP$^+$03] T.Nieberg, S.Dulman, P.Havinga, L. van Hoesel, and J. Wu. *Collaborative Algorithms for Communication in Wireless Sensor Networks*. Kluwer Academic Publishers, 2003.

[Van90]   A. Van de Liefvoort. The moment problem for continuous distributions. Technical report, University of Missouri, 1990.

[vGSST90] R. van Glabbeek, S.A. Smolka, B. Steffen, and C.M.N. Tofts. Reactive, generative, and stratified models of probabilistic processes. In *Logic in Computer Science, 1990. LICS '90, Proceedings., Fifth Annual IEEE Symposium on e*, pages 130–141, June 1990.

[vH07]   L.F.W. van Hoesel. *Sensors on speaking terms: schedule-based medium access control protocols for wireless sensor networks*. PhD thesis, University of Twente, 2007.

[vHH04]   L.F.W. van Hoesel and P.J.M. Havinga. A lightweight medium access protocol (lmac) for wireless sensor networks: Reducing preamble transmissions and tranceiver state switches. In *1st International Workshop on Networked Sensing Systems (INSS 2004)*, pages 205–208. Society of Instrument and Control Engineers(SICE), 2004.

[vHH06]   L.F.W. van Hoesel and P.J.M. Havinga. Design aspects of an energy-efficient, lightweight medium access control protocol for wireless sensor networks. Technical report, Centre for Telematics and Information Technology, University of Twente, Enschede, 2006.

[VR09]   M.S. Vighio and A.P. Ravn. Analysis of collisions in wireless sensor networks. In *21st Nordic Workshop on Programming Theory (NWPT 2009)*, Lyngby, Denmark, 2009.

[Wol08]   Verena Wolf. *Equivalences on Phase Type Processes*. PhD thesis, University of Mannheim, 2008.

[ZENN12]   Kebin Zeng, Luz J. R. Esparza, Bo Friis Nielsen, and Flemming Nielson. Stochastic equivalence and time-lapse bisimulation. In *under submission*, 2012.

[ZNN12]   Kebin Zeng, Bo Friis Nielsen, and Flemming Nielson. Phase–type
           process algebra. Unpublished manuscript, October 2012.

[ZNNH12]   Kebin Zeng, Bo Friis Nielsen, Flemming Nielson, and Holger Her-
           manns. Stochastic model checking without Markov chains. Unpub-
           lished manuscript, October 2012.