



Reduction of Acyclic Phase-Type Representations

Dissertation zur Erlangung des Grades des Doktors der Ingenieurwissenschaften der
Naturwissenschaftlich-Technischen Fakultäten der Universität des Saarlandes

Muhammad Reza Pulungan

Saarbrücken, 2009

Tag des Kolloquiums
Dekan

27.05.2009
Prof. Dr. Joachim Weickert

Prüfungsausschuss

Vorsitzender
Berichterstattende

Prof. Dr. Dr. h.c. mult. Reinhard Wilhelm
Prof. Dr.-Ing. Holger Hermanns
Prof. Dr.-Ing. Markus Siegle
Prof. Dr. Raymond Marie

Akademischer Mitarbeiter

Dr.-Ing. Jan Reineke

Abstract

Acyclic phase-type distributions are phase-type distributions with triangular matrix representations. They constitute a versatile modelling tool, since they (1) can serve as approximations to any continuous probability distribution, and (2) exhibit special properties and characteristics that usually make their analysis easier. The size of the matrix representations has a strong effect on the computational efforts needed in analyzing these distributions. These representations, however, are not unique, and two representations of the same distribution can differ drastically in size.

This thesis proposes an effective algorithm to reduce the size of the matrix representations without altering their associated distributions. The algorithm produces significantly better reductions than existing methods. Furthermore, the algorithm consists in only standard numerical computations, and therefore is straightforward to implement. We identify three operations on acyclic phase-type representations that arise often in stochastic models. Around these operations we develop a simple stochastic process calculus, which provides a framework for stochastic modelling and analysis. We prove that the representations produced by the three operations are “almost surely” minimal, and the reduction algorithm can be used to obtain these almost surely minimal representations. The applicability of these contributions is exhibited on a variety of case studies.

Zusammenfassung

Azyklische Phasentypverteilungen sind Phasentypverteilungen, deren Matrixdarstellung eine Dreiecksmatrix ist. Sie stellen ein vielseitiges Modellierungswerkzeug dar, da sie einerseits als Approximationen jeder beliebigen kontinuierlichen Wahrscheinlichkeitsverteilung dienen können, und andererseits spezielle Eigenschaften und Charakteristiken aufweisen, die ihre Analyse vereinfachen. Die Größe der Matrixdarstellung hat dabei einen starken Einfluss auf den Berechnungsaufwand, der zur Analyse solcher Verteilungen nötig ist. Die Matrixdarstellung ist jedoch nicht eindeutig, und zwei verschiedene Darstellungen ein und derselben Verteilung können sich drastisch in ihrer Größe unterscheiden.

In dieser Arbeit wird ein effektiver Algorithmus zur Verkleinerung der Matrixdarstellung vorgeschlagen, der die mit der Darstellung assoziierte Verteilung nicht verändert. Dieser Algorithmus verkleinert die Matrizen dabei beträchtlich stärker als bereits existierende Methoden. Darüberhinaus bedient er sich nur numerischer Standardverfahren, wodurch er einfach zu implementieren ist. Wir identifizieren drei Operationen auf azyklischen Phasentypdarstellung, die in stochastischen Modellen häufig anzutreffen sind. Von diesen Operationen ausgehend entwickeln wir einen einfachen stochastischen Prozess-Kalkül, der eine grundlegende Struktur für stochastische Modellierung und Analyse darstellt. Wir zeigen, dass die durch die drei Operationen erzeugten Darstellungen „beinahe gewiss“ minimal sind und dass der Reduktionsalgorithmus benutzt werden kann, um diese beinahe gewiss minimalen Darstellungen zu erzeugen. Die Anwendbarkeit dieser Beiträge wird an einer Reihe von Fallstudien exemplifiziert.

Acknowledgements

First and foremost, I thank my supervisor Holger Hermanns for his patience and diligence in guiding me throughout my years in Saarbrücken. He has been an excellent mentor who is ever ready for discussions.

I would also like to thank my colleagues in the group: Pepijn Crouzen, Christian Eisentraut, Moritz Hahn, David Jansen, Sven Johr, Christa Schäfer and Lijun Zhang, for their friendship and support.

Special thanks go to Basileios Anastasatos, Pepijn Crouzen and Christian Eisentraut for helping me proofreading this thesis. In the end, however, mistakes are mine alone.

I am lucky to have a big and happy family. I am grateful to my mother; my sisters: Nevi and Elvida; my brothers: Fauzi, Lian, Randi and Yazid; for their endless love and support.

This thesis is dedicated to the memory of my father.

Contents

1	Introduction	1
2	Preliminaries	5
2.1	Mathematical Notations	6
2.2	Random Variables	6
2.3	Exponential Distributions	7
2.4	Markov Chains	7
2.4.1	Stochastic Processes	7
2.4.2	Markov Processes	8
2.4.3	Discrete-Time Markov Chains	8
2.4.4	Continuous-Time Markov Chains	11
2.5	Phase-Type Distributions	13
2.5.1	General Notions and Concepts	16
2.5.2	Order and Degree	18
2.5.3	Dual Representations	19
2.5.4	Characterization	20
2.5.5	Closure Properties	21
2.6	Acyclic Phase-Type Distributions	23
2.6.1	Characterization	24
2.6.2	Closure Properties	25
2.6.3	Erlang and Hypoexponential Distributions	25
2.7	The Polytope of Phase-Type Distributions	26
2.7.1	Residual-Life Operator	27
2.7.2	Simplicity and Majorization	28
2.7.3	Geometrical View of APH Representations	30
2.8	Matrix-Exponential Distributions	33
3	Reducing APH Representations	35
3.1	Acyclic Canonical Forms	36
3.1.1	Ordered Bidiagonal Representation	37
3.1.2	Cox Representation	38
3.2	Transformation to Ordered Bidiagonal	39
3.3	Transformation Algorithms	41
3.3.1	Cumani's Algorithm	41
3.3.2	O'Conneide's Algorithm	42
3.3.3	Spectral Polynomial Algorithm	43

3.4	Reducing the Representations	44
3.4.1	The L -terms	44
3.4.2	Reduction	45
3.4.3	Algorithm	48
3.5	Examples: Fault-Tolerant System	50
3.6	Relation to Lumping	51
3.7	Conclusion	56
4	Operations on Erlang Distributions	59
4.1	Refining the Basic Series	60
4.2	Convolution Operation	63
4.3	Minimum Operation	64
4.3.1	The Minimum of Two Erlang Distributions	64
4.3.2	The Minimum of More Erlang Distributions	65
4.3.3	The Minimal of the Minimum	67
4.4	Maximum Operation	69
4.4.1	The Maximum of Two Erlang Distributions	70
4.4.2	The Maximum of More Erlang Distributions	71
4.4.3	The Minimal of the Maximum	73
4.5	Conclusion	76
5	The Use of APH Reduction	79
5.1	Minimal and Non-Minimal Representations	80
5.2	When Order = Algebraic Degree	82
5.2.1	Known Results	82
5.2.2	Our Reduction Algorithm	86
5.3	Operations on APH Representations	90
5.3.1	Convolution Operation	90
5.3.2	Minimum and Maximum Operations	93
5.4	Almost Surely Minimal	96
5.5	Conclusion	99
6	A Simple Stochastic Calculus	101
6.1	CCC Processes	102
6.1.1	Intuitions	102
6.1.2	Syntax	104
6.1.3	Semantics	105
6.2	CCC Processes and PH Distributions	109
6.3	Some Notions of Equivalence	112
6.3.1	Bisimulations	112
6.3.2	PH-Equivalence	114
6.4	Equivalence Checking and Process Reduction	115
6.4.1	Algorithmic Considerations	116
6.4.2	Compositional Considerations	117
6.5	Conclusion	118

7 Case Studies	119
7.1 Fault Trees with PH Distributions	120
7.2 Fault-Tolerant Parallel Processors	123
7.3 Delay in a Railway Network	127
7.4 Conclusion	138
8 Conclusion	141
Appendices	
A Basic Concepts	143
A.1 Poisson Processes	143
A.2 Kronecker Product and Sum	144
A.3 Some Concepts from Convex Analysis	144
A.3.1 Affine Sets	144
A.3.2 Convex Sets	145
B Proofs	147
B.1 Lemma 3.14	147
B.2 Lemma 5.14	149
B.3 Lemma 6.12	150
B.4 Lemma 6.18	154
Bibliography	159

Chapter 1

Introduction

Due to the incomplete nature of our knowledge of the physical systems we interact with, they appear to exhibit stochastic behaviors. This is also true for most of computer and electronic communication systems we increasingly rely on. These stochastic behaviors usually influence the performance of a system, but they may also affect the functional characteristics or even the correctness of the system. When such systems are analyzed, it is important therefore not to neglect their stochastic behaviors.

One of the many approaches we can take in analyzing such systems is *model-based analysis*. In this approach, the analysis is directed at an abstract description of the system—its components and their interactions as well as the system’s interactions with the environment—instead of at the system itself [Hav98]. In many cases, this approach is taken because access to the real system is difficult or impossible. But more importantly, this approach allows us to abstract from the real system, to focus and concentrate on the parts of the system that are relevant and interesting. This way, analysis can be greatly simplified.

Further, one of the ways to carry out a model-based analysis is *state-based analysis*. A state-based analysis proceeds by modelling the system in terms of its states (*i.e.*, some distinguishable characteristics) and how the system changes from one state to others. An obvious requirement in this analysis is that the possible states to which the system changes from a particular state must be a well-defined subset of the state space. In this field, Markov chains play a befitting role: since the immediate future state of a Markov chain depends only on its current state and not on the states visited prior to the current one, they meet the requirement.

Research in the field of stochastic state-based analysis has been flourishing for many years. Progress has been made in the foundation and algorithms for model checking Markov models, especially in the continuous-time setting [ASSB00, BHHK03]. Various Markovian process calculi, such as MTIPP [HR94], PEPA [Hil96], EMPA [BG96] and IMC [Her02], have been proposed, which open the possibility of compositionality in the construction of models. The field also enjoys a proliferation of tools, such as PRISM [HKNP06], ETMCC [HKMKS03], its successor MRMC [KKZ05] and CASPA [RSS08]. A comparative study of some of these prominent tools can be found in [JKO⁺08].

Markovian modelling and analysis are widely used in diverse fields of computer science and engineering, from queueing theory [Neu81, Asm92], computer network design [CR91, KSH03], to reliability analysis [CKU92], for instance in the analysis of dynamic fault trees [MDGS98, BCS07a].

However, the field of stochastic state-based analysis is faced by a major challenge:

the state-space explosion problem. Compositions of Markov models are usually accomplished by the cross products of the state spaces of the involved models. As a result, the state space of composite models grows too big very quickly, they exceed the size of the memory of standard modern computers. One of the ways to deal with this challenge is to avoid representing and working with explicit states, but instead to encode the state space in a symbolic and more compact way, for instance by using binary decision diagrams (BDDs) [HKN⁺03] or Kronecker representations [PA91, BCDK00, HK01].

Another way to deal with the challenge is by using lumping [KS76, Nic89, Buc94]. Lumping defines an aggregation of a Markov chain by identifying a partitioning of its state space such that all states in a particular partition share some common characteristics. When such partitioning can be identified, all states that belong to a single partition can be aggregated by (or lumped together into) a single state without altering the overall stochastic behavior of the original Markov chain. Several methods exist to identify the partitioning; the most widely used among them are those based on bisimulation equivalences on Markov chains [Bra02, DHS03]. More recently, improvements on the bisimulation-based lumping algorithm especially tailored for models without cycles are provided in [CHZ08].

A question, nevertheless, arises: is bisimulation-based lumping useful in practice? The answer is affirmative as shown by the authors of [KKZJ07]. They showed that the time needed to analyze a Markov chain *mostly exceeds* the combination of the time needed to aggregate it and the time needed to analyze the aggregated Markov chain. Their results also indicate that enormous state-space reductions (up to logarithmic savings) may be obtained through lumping. As it stands nowadays, lumping is the best mechanism we have to reduce the size of the state space of Markov models.

This thesis proposes a reduction algorithm that goes beyond lumping, in the sense that, the new algorithm is guaranteed to reduce the state space of Markov models no less than lumping. The algorithm performs state-space reduction of representations of (continuous-time) phase-type distributions.

Phase-type distributions are a versatile and tractable class of probability distributions, retaining the principal analytical tractability of exponential distributions, on which they are based. Phase-type distributions are topologically dense [JT88] on the support set $[0, \infty)$. Therefore, they can be used to approximate arbitrarily closely other probability distributions or traces of empirical distributions obtained from experimental observations. This broadens the applicability of stochastic analysis of this type, since we can then incorporate in our models probability distributions that would be otherwise intractable by fitting phase-type distributions to them.

Any phase-type distribution agrees with the distribution of the time until absorption in some (continuous-time) Markov chain with an absorbing state [Neu81]. Such a Markov chain is actually the basis of numerical or analytical analysis for models involving that phase-type distribution, and it is therefore called the *representation* of that distribution. These representations are not unique: distinct absorbing Markov chains may represent the same distribution, and any phase-type distribution is represented by infinitely many distinct absorbing Markov chains. The representations differ in particular with respect to their *size*, *i.e.*, the number of their states and transitions. Thus, for a given phase-type distribution, an obvious question to pose is what the *minimal* representation of that distribution may be, and how to construct it. Lumping is of course equally applicable to absorbing Markov chains too, but the computed fix-point is not guaranteed to be the minimal representation in the above sense.

The problem of identifying and constructing smaller-sized representations is one of the most interesting theoretical research questions in the field of phase-type distributions. The focus of this thesis is on the class of acyclic phase-type distributions, *i.e.*, phase-type distributions with (upper) triangular matrix representations. Like the general phase-type distributions, acyclic phase-type distributions are also topologically dense on the support set $[0, \infty)$ [JT88].

The systematic study of acyclic phase-type representations was initiated in [Cum82], and later, in [O’C91, O’C93], minimality conditions are identified, but without algorithmic considerations. A closed-form solution for the transient analysis of acyclic Markov chains—hence also acyclic phase-type representations—is presented in [MRT87]. The quest for an algorithm to construct the minimal representation of any acyclic phase-type distribution has recently seen considerable advances: an algorithm for computing minimal representations of acyclic phase-type distributions is provided in [HZ07a]. This algorithm involves converting a given acyclic phase-type distribution to a representation that only contains states representing the poles of the distribution. This representation does not necessarily represent an acyclic phase-type distribution, but a matrix-exponential distribution [Fac03]. If this is the case, another state and its total outgoing rate are determined and added to the representation. This is performed one by one until an acyclic phase-type representation is obtained. This results in a representation of provably minimal size. This algorithm involves solving a system of non-linear equations for each additional state.

Our Contribution The algorithm developed in this thesis addresses the same problem, but in the opposite way. Instead of adding states to a representation until it becomes an acyclic phase-type representation, we eliminate states from the given representation as we proceed, until no further elimination is possible. An elimination of a state involves solving a system of linear equations. The reduction algorithm we propose is of cubic complexity in the size of the state space of the given representation, and only involves standard numerical computations. The algorithm is guaranteed to return a smaller or equal size representation than the given one, and also than the aggregated representation produced by lumping. This state-space reduction algorithm is the core contribution of this thesis, and it is embedded in a collection of observations of both fundamental and pragmatic nature.

We also identify three operations on acyclic phase-type distributions and representations that arise often when constructing a stochastic model. These operations are convolution, minimum and maximum. Around these operations we develop a simple stochastic process calculus that captures the generation and manipulation of acyclic phase-type representations. This process calculus provides a framework for stochastic modelling and analysis. In this calculus, we analyze the reduction algorithm more deeply to identify the circumstances where it is beneficial to use it. We prove that representations produced by the three operations are “almost surely” minimal, and the reduction algorithm can be used to obtain these almost surely minimal representations. On the more specific Erlang distributions, we show that the representations obtained from the application of each of these operations can always be reduced to minimal representations.

Finally, to demonstrate the practical potential of the reduction algorithm, we apply it to several case studies.

Organization of the Thesis The thesis is divided into eight chapters. They are organized as follows:

- Chapter 2 provides a comprehensive introduction to phase-type distributions, acyclic phase-type distributions and other basic concepts and notions required throughout the thesis. We do our best to include most of contemporary knowledge and results in the field of acyclic phase-type distributions in this chapter.
- In Chapter 3, we develop our reduction algorithm. We first set the ground by visiting several previous algorithms that form the basis of our algorithm. A small example is provided to guide the reader and to clarify the inner workings of the algorithm. We discuss the relation of the algorithm to weak-bisimulation-based lumping.
- Chapter 4 deals primarily with the minimal representations of the convolution, minimum and maximum of Erlang distributions. In proving minimalities, we put forward a new concept of the core series, which improves on a similar and existing concept, and furthermore provides a handy tool in many proofs.
- In Chapter 5, we delve more deeply into the reduction algorithm. We compare it with several existing results related to acyclic phase-type representations. We identify the conditions under which the algorithm is useful. We discuss the effect of the three operations on acyclic phase-type representations, and the role the algorithm plays in reducing the results of the operations.
- In Chapter 6, we develop a simple stochastic process calculus to generate and manipulate acyclic phase-type representations. We define three congruent notions of equivalence on processes defined in this calculus.
- In Chapter 7, we demonstrate the applicability of the reduction algorithm by analyzing three case studies. From the case studies we learn the strength and the weakness of the algorithm.
- In Chapter 8, we conclude the thesis by looking back at what we have achieved in this thesis, and then looking forward at future possibilities worth exploring.

Chapter 2

Preliminaries

This chapter lays out general concepts, notions and notations used throughout the thesis. It also provides a coherent introduction to phase-type distributions, acyclic phase-type distributions, and their representations.

Related Work Most of the material in this chapter is based on existing literature. Standard textbooks, such as [Ste94, Hav98, LR99, Tij07, Ros07], are useful guidelines in the expositions of the topics related to stochastic processes and Markov chains. NEUTS' monograph [Neu81] and O'CONNOR's seminal paper [O'C90] are the major sources for the topic of phase-type distributions. The concept of dual representations was first discussed in [CC93], and later expanded in [CM02]. The characterization of phase-type distributions was proved in [O'C90], while that of acyclic phase-type distributions was proved in [Cum82, O'C91]. The closure characterizations of phase-type distributions and acyclic phase-type distributions were presented in [MO92] and [AL82], respectively. The idea of the polytope of phase-type distributions was first developed in [O'C90], while the notions of PH-simplicity and PH-majorization were developed in [O'C89]. The main references for the section on the geometrical view of acyclic phase-type representations are [DL82] and [HZ06a]. The short discussion on matrix-exponential distributions is mainly based on [AO98, Fac03].

Structure The chapter is organized as follows: Several basic mathematical notations related to matrices and vectors are summarized in Section 2.1. Sections 2.2 and 2.3 give an overview on the basic concepts of random variables and exponential distributions. These two sections provide the necessary foundation for discussing stochastic and Markov processes, especially Markov chains, which are described in Section 2.4. In Sections 2.5 and 2.6, the concepts of phase-type distributions and acyclic phase-type distributions are introduced, covering their parameterization and structural definitions, important properties, characterizations, and closures. In Section 2.7, we discuss a geometrical method for studying phase-type distributions by examining their polytopes. Finally, Section 2.8 provides a brief introduction to matrix-exponential distributions, touching only the notions that are relevant to the thesis.

2.1 Mathematical Notations

The set of real numbers is denoted by \mathbb{R} , and the set of integers is denoted by \mathbb{Z} . The nonnegative restriction of the sets \mathbb{R} and \mathbb{Z} is denoted by $\mathbb{R}_{\geq 0}$ and respectively $\mathbb{Z}_{\geq 0}$, while the strictly positive restriction of the sets is denoted by \mathbb{R}_+ and \mathbb{Z}_+ , respectively.

Vectors are written with an arrow over them, e.g., \vec{v} . The *dimension* of a vector is the number of its components. Let the dimension of \vec{v} be n , then its i -th component, for $1 \leq i \leq n$, is denoted by \vec{v}_i . Vector \vec{e} is a vector whose components are all equal to 1. Vector $\vec{0}$ is a vector whose components are all equal to 0. The dimension of \vec{e} or $\vec{0}$ should be clear from the context. Column and row vectors are indistinguishable notationally; the context should clarify the distinction. Vector \vec{v}^\top is the transpose of vector \vec{v} . A vector $\vec{v} \in \mathbb{R}_{\geq 0}^n$ is stochastic if $\vec{v}\vec{e} = 1$, and sub-stochastic if $\vec{v}\vec{e} \leq 1$.

Matrices are written in bold capital letters, e.g., \mathbf{A} . The dimension of a matrix with m rows and n columns is mn . For square matrices, the dimension is shortened to just m instead of mm . The component in the i -th row and the j -th column, for $1 \leq i \leq m$ and $1 \leq j \leq n$, is denoted by $\mathbf{A}(i, j)$. Matrices \mathbf{A}^\top and \mathbf{A}^{-1} are the transpose and the inverse of matrix \mathbf{A} , respectively.

2.2 Random Variables

The basic concept in probability theory is *probability space*, namely a tuple $(\Omega, \mathcal{F}, \Pr)$, where:

- Ω is the *sample space*, a set containing all possible outcomes of an experiment,
- \mathcal{F} is a σ -algebra on subsets of Ω called *events*, i.e., $\mathcal{F} \subseteq 2^\Omega$ satisfying
 1. $\Omega \in \mathcal{F}$,
 2. if $A \in \mathcal{F}$ then so is the complement of A relative to Ω , i.e., $A^c \in \mathcal{F}$, and
 3. for every sequence of $A_i \in \mathcal{F}$, $i \geq 1$, then $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$,
- \Pr is a *probability measure*, namely a function $\Pr : \mathcal{F} \rightarrow [0, 1]$ that satisfies
 1. $\Pr(\Omega) = 1$, and
 2. for every sequence of pair-wise disjoint events $A_i \in \mathcal{F}$, $i \geq 1$, \Pr is σ -additive, i.e., $\Pr(\sum_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \Pr(A_i)$.

On the probability space, we can define a random variable, which can be continuous or discrete. In this thesis, we focus on the continuous (real-valued) random variables. Let $\mathbb{R}_\infty := \mathbb{R} \cup \{+\infty, -\infty\}$.

Definition 2.1. Let $(\Omega, \mathcal{F}, \Pr)$ be a probability space. A (continuous) random variable X over the probability space is a function $X : \Omega \rightarrow \mathbb{R}_\infty$ such that for all $t \in \mathbb{R}$

$$\{\omega \mid X(\omega) \leq t\} \in \mathcal{F}.$$

The continuous random variable defined above is characterized by its *distribution function* $F : \mathbb{R} \rightarrow [0, 1]$ given by

$$F(t) = \Pr(\{\omega \mid X(\omega) \leq t\}).$$

To shorten the notation, we write $\Pr(X \leq t)$ instead of $\Pr(\{\omega \mid X(\omega) \leq t\})$. The random variable is also characterized by its *probability density function* $f : \mathbb{R} \rightarrow [0, 1]$ such that

$$F(t) = \int_0^t f(x)dx.$$

For the given probability space, the *support* of the probability measure \Pr is defined as the smallest subset $A \in \mathcal{F}$ of Ω such that $\Pr(A^c) = 0$.

2.3 Exponential Distributions

The negative exponential distributions, which in this thesis will always be referred to simply as the exponential distributions, are continuous probability distributions. They are widely used in stochastic models, such as in the fields of performance analysis, dependability, and queueing theory. The exponential distributions are memoryless, which means that their distribution functions do not depend on the amount of time that has passed. This property makes them suitable for modelling the time between independent occurrences of events that occur at some constant rate. Random variables, such as the interarrival time of jobs to a file server or the service time of a server in a queueing network, are often modelled by exponential distributions [Tri02].

Definition 2.2. A random variable X is distributed according to an exponential distribution with rate $\lambda \in \mathbb{R}_+$ if its distribution function is given by

$$F(t) = \Pr(X \leq t) = \begin{cases} 1 - e^{-\lambda t}, & t \in \mathbb{R}_{\geq 0}, \\ 0, & \text{otherwise.} \end{cases} \quad (2.1)$$

The probability density function of the exponential distribution is

$$f(t) = \begin{cases} \lambda e^{-\lambda t}, & t \in \mathbb{R}_+, \\ 0, & \text{otherwise.} \end{cases}$$

An exponential distribution with rate λ is denoted by $\text{Exp}(\lambda)$.

2.4 Markov Chains

In this section, we introduce discrete-time and continuous-time Markov chains. In order to do that, we need first to provide the background and underlying concepts of stochastic processes and Markov processes.

2.4.1 Stochastic Processes

A *stochastic process* is a collection of random variables $\{X_t \mid t \in \mathcal{T}\}$ that are indexed by a parameter t , which takes values from a set \mathcal{T} (usually the time domain). The values that X_t assumes are called *states*, and the set of all possible states is called the *state space*, denoted by \mathcal{S} . Both sets \mathcal{S} and \mathcal{T} can be discrete or continuous.

At a particular time $t \in \mathcal{T}$, the random variable X_t may take different values. The distribution function of the random variable X_t at $t \in \mathcal{T}$ is

$$F(x, t) = \Pr(X_t \leq x).$$

This is called the *cumulative distribution function* (cdf) of the random variable or the first-order distribution of the stochastic process $\{X_t \mid t \in \mathcal{T}\}$. This function can be extended to the n -th joint distribution of the process

$$F(\vec{x}, \vec{t}) = \Pr(X_{\vec{t}_1} \leq \vec{x}_1, \dots, X_{\vec{t}_n} \leq \vec{x}_n),$$

where vectors \vec{x} and \vec{t} are of dimension n , $\vec{x}_i \in \mathcal{S}$ and $\vec{t}_i \in \mathcal{T}$, for $1 \leq i \leq n$.

A stochastic process where the state occupied at a certain time point does not depend on the state(s) being occupied at any other time point is an *independent process*. Mathematically, an independent process is a stochastic process whose n -th order joint distribution satisfies

$$F(\vec{x}, \vec{t}) = \prod_{i=1}^n F(\vec{x}_i, \vec{t}_i) = \prod_{i=1}^n \Pr(X_{\vec{t}_i} \leq \vec{x}_i).$$

A stochastic process can also be a *dependent process*, in which case some form of dependency exists among successive states.

2.4.2 Markov Processes

A stochastic process where a dependency exists only between two successive states is called a *Markov process*. Such a dependence is called Markov dependence or first-order dependence.

Definition 2.3. A stochastic process $\{X_t \mid t \in \mathcal{T}\}$ is a Markov process if for all $t_0 < t_1 < \dots < t_n < t_{n+1}$, the distribution of the conditional probability of $X_{t_{n+1}}$, given the values of X_{t_0}, \dots, X_{t_n} , depends only on X_{t_n} , i.e.,

$$\Pr(X_{t_{n+1}} \leq x_{n+1} \mid X_{t_0} = x_0, \dots, X_{t_n} = x_n) = \Pr(X_{t_{n+1}} \leq x_{n+1} \mid X_{t_n} = x_n). \quad (2.2)$$

Equation (2.2) expresses the *Markov property*: that for any given time point t_n , the future behavior—i.e., the values the random variable $X_{t_{n+1}}$ can take—depends only on the current state at time point t_n (X_{t_n}).

A Markov process is *time-homogeneous* if it is invariant to time shifts. This means that the behavior of the process is independent of the time of observation. In this case, for any $t_1 < t_2$, x_1 and x_2

$$\Pr(X_{t_2} \leq x_2 \mid X_{t_1} = x_1) = \Pr(X_{t_2-t_1} \leq x_2 \mid X_0 = x_1).$$

All Markov processes discussed in this thesis are time-homogeneous.

If the state space \mathcal{S} of a Markov process is discrete, the Markov process is a *Markov chain*. Two types of Markov chains can be identified: discrete-time and continuous-time Markov chains.

2.4.3 Discrete-Time Markov Chains

A *discrete-time Markov chain* (DTMC) is a stochastic process $\{X_t \mid t \in \mathbb{Z}_{\geq 0}\}$ that has a discrete state space \mathcal{S} and a discrete index set $\mathcal{T} := \mathbb{Z}_{\geq 0}$ and satisfies the Markov property (which in this case can be written as)

$$\Pr(X_{n+1} = x_{n+1} \mid X_0 = x_0, \dots, X_n = x_n) = \Pr(X_{n+1} = x_{n+1} \mid X_n = x_n),$$

where $x_0, \dots, x_n, x_{n+1} \in \mathcal{S}$.

Without loss of generality, assume that the elements of \mathcal{S} range over a subset of the natural numbers. We call the conditional probabilities $p_{i,j}(n) := \Pr(X_{n+1} = j \mid X_n = i)$ the *transition probabilities*. They are the probabilities of making a transition from state $i \in \mathcal{S}$ to state $j \in \mathcal{S}$ when time increases from n to $n+1$. In time-homogeneous DTMCs, $p_{i,j}(n)$ is independent of n , namely for all $n, m \in \mathbb{Z}_+$, $p_{i,j}(n) = p_{i,j}(m)$. The transition probabilities for all states $i, j \in \mathcal{S}$ can be represented by a *transition probability matrix* \mathbf{P} , where

$$\mathbf{P}(i, j) = p_{i,j}(n).$$

A time-homogeneous DTMC is fully described by its initial probability vector $\vec{p}(0)$ —where $\vec{p}_i(0)$ gives the probability that the DTMC starts in state i —and its transition probability matrix \mathbf{P} .

Graphical Representation A DTMC can be represented by a labelled directed graph. The vertices of the graph stand for the states of the DTMC, and the name of a state is placed inside the vertex that represents the state. A transition is represented by an edge in the graph. The probability associated with a transition is placed near the edge representing the transition. An example of such graph is depicted in Figure 2.1.

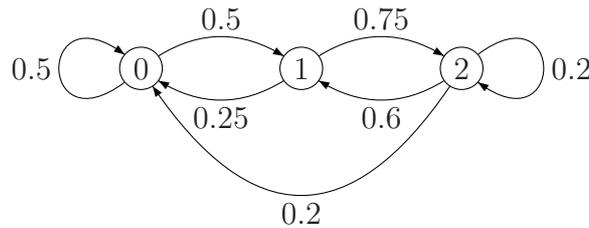


Figure 2.1: A Discrete-Time Markov Chain

Example 2.4. Consider the DTMC depicted in Figure 2.1. There are three states in the DTMC, named 0, 1 and 2. The DTMC also has seven transitions. These transitions are shown together with their probabilities, for instance, transition from state 0 to state 1 occurs with probability 0.5. The transition probability matrix of the DTMC is

$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.25 & 0 & 0.75 \\ 0.2 & 0.6 & 0.2 \end{bmatrix}.$$

Transient Analysis The purpose of transient analysis is to determine the probability with which a DTMC occupies a state after a given number of transitions have occurred. This probability is called the *transient probability* after the given number of transitions. The transient probabilities of all states after n transitions ($\vec{p}(n)$) can be obtained by evaluating

$$\vec{p}(n) = \vec{p}(0) \mathbf{P}^n, \quad n \in \mathbb{Z}_{\geq 0},$$

where $\vec{p}(0)$ is the initial probability vector and \mathbf{P} is the transition probability matrix of the DTMC. The transient probabilities, $\vec{p}(n)$, describe the transient behavior of the DTMC.

Example 2.5. We use the DTMC depicted in Figure 2.1. Let $\vec{p}_0(0) = 1$ and $\vec{p}_i(0) = 0$, for $i = 1, 2$. The transient probability vector of the DTMC after 3 transitions can be computed as follows

$$\vec{p}(3) = \vec{p}(0) \mathbf{P}^3 = [1, 0, 0] \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.25 & 0 & 0.75 \\ 0.2 & 0.6 & 0.2 \end{bmatrix}^3 = [0.325, 0.4125, 0.2625].$$

These transient probabilities describe the probabilities of being in some state after 3 transitions starting from state 0. For instance, after 3 transitions the DTMC will be in state 2 with probability 0.2625. Transient probabilities after more transitions have occurred can be computed in a similar fashion. The transient probability vector after 15 and 25 transitions, for instance, is

$$\begin{aligned} \vec{p}(15) &= [0.3111, 0.35567, 0.33323], \text{ and} \\ \vec{p}(25) &= [0.3111, 0.35556, 0.33333]. \end{aligned}$$

In the previous example, we can observe that after a certain number of transitions, the transient probabilities converge to a limiting stationary distribution. It is interesting to know whether the limiting probabilities can be obtained directly since for some measures of interest these probabilities may be sufficient. However, such limiting probabilities do not always exist for all DTMCs. The conditions under which these probabilities exist—ergodicity—can be found in standard textbooks, such as [Ste94, LR99, Hav98, Ros07, Tij07].

Steady-State Analysis Steady-state analysis is used to determine the transient probabilities when the *equilibrium*—i.e., when the effect of the initial distribution has disappeared [Ste94]—has been reached. These probabilities are called the *steady-state probabilities*. If the limit $\vec{v} = \lim_{n \rightarrow \infty} \vec{p}(n)$ exists, the steady-state probabilities \vec{v} can be obtained from the system of linear equations

$$\vec{v} = \vec{v} \mathbf{P}, \quad \sum_{i \in \mathcal{S}} \vec{v}_i = 1, \quad \text{and} \quad 0 \leq \vec{v}_i \leq 1. \quad (2.3)$$

If $\lim_{n \rightarrow \infty} \vec{p}(n)$ does not exist, for instance when the DTMC is periodic [Ste94], it can be shown by the Cesàro-limit [Tij07, LR99] that the system of equations (2.3) still yields a unique solution. Therefore, it is safe to define the solution of the system of equations as the steady-state probabilities.

Example 2.6. We continue using the DTMC depicted in Figure 2.1 in this example. For the given DTMC, the steady-state probability vector is $[\frac{14}{45}, \frac{16}{45}, \frac{1}{3}]$.

Steady-state probabilities can be interpreted as the probabilities of discovering that the DTMC is in some state after it has been running for a long time. They can also be interpreted as the fraction of time the DTMC spends in some state in the long run. Thus, for the example, it can be said that after a long running time the DTMC will be in state 2 with probability $\frac{1}{3}$, or the fraction of time the DTMC spends in state 2 is $\frac{1}{3}$.

2.4.4 Continuous-Time Markov Chains

A *continuous-time Markov chain* (CTMC) is a stochastic process $\{X_t \mid t \in \mathbb{R}_{\geq 0}\}$ that has a discrete state space \mathcal{S} , a continuous index set $\mathcal{T} := \mathbb{R}_{\geq 0}$, and satisfies the Markov property (which in this case can be written as)

$$\Pr(X_{t_{n+1}} = x_{n+1} \mid X_{t_0} = x_0, \dots, X_{t_n} = x_n) = \Pr(X_{t_{n+1}} = x_{n+1} \mid X_{t_n} = x_n),$$

for all $t_0 < t_1 < \dots < t_n < t_{n+1} \in \mathcal{T}$.

While a DTMC is described by its transition probability matrix, a CTMC is described by its *rate matrix* \mathbf{R} . With every pair of states $i, j \in \mathcal{S}$, we associate a *rate* $r(i, j)$, such that $r(i, j)$ determines the delay and the probability of the transition from state i to state j . The rates of the transitions between all pairs of states can then be conveniently represented by \mathbf{R} , where

$$\mathbf{R}(i, j) = r(i, j).$$

for all $i, j \in \mathcal{S}$. Let the sum of the rates of all outgoing transitions from state i be $E(i)$, i.e., $E(i) = \sum_{j \in \mathcal{S}} \mathbf{R}(i, j)$. We call this sum the *total outgoing rate* or the *rate of residence* of the state.

The semantics of the CTMC is as follows: at any given time point, the CTMC is in one of the states. Let the CTMC enter state i at some time point. From this state, the CTMC can transition to any state $j \in \mathcal{S}$ if $\mathbf{R}(i, j) > 0$. Assuming that the only outgoing transition from state i is to state j , then the delay of this transition is governed by an exponential distribution with rate $\mathbf{R}(i, j)$, whose distribution function is

$$F_{i,j}(t) = 1 - e^{-\mathbf{R}(i,j)t}.$$

However, when other outgoing transitions exist from state i , this delay must compete with the delays of the other transitions. A *race condition* occurs among the delays, and the shortest among them wins. This shortest delay corresponds to the minimum of exponential distributions governing the delays of all outgoing transitions, which itself is an exponential distribution with rate $\sum_{j \in \mathcal{S}} \mathbf{R}(i, j)$, whose distribution function is

$$F_i(t) = 1 - e^{-\sum_{j \in \mathcal{S}} \mathbf{R}(i,j)t} = 1 - e^{-E(i)t}.$$

This distribution describes the residence time in state i , namely the time the CTMC resides there before making any transition. The probability that a particular transition to state j occurs (thus, its corresponding delay wins the race) given other transitions from state i is

$$p(i, j) = \frac{\mathbf{R}(i, j)}{\sum_{k \in \mathcal{S}} \mathbf{R}(i, k)} = \frac{\mathbf{R}(i, j)}{E(i)}.$$

Hence, the probability of making a transition from state i to a state j within t time units is given by $p(i, j)F_i(t) = p(i, j)(1 - e^{-E(i)t})$.

A CTMC is completely specified by its initial probability vector $\vec{p}(0)$ and its rate matrix \mathbf{R} . Besides its rate matrix \mathbf{R} , a CTMC can also be specified by its *infinitesimal generator matrix* \mathbf{Q} , where

$$\mathbf{Q}(i, j) = \begin{cases} \mathbf{R}(i, j), & i \neq j, \\ -E(i), & i = j. \end{cases}$$

Transient Analysis The transient probabilities ($\vec{p}(t)$) of a CTMC with initial probability vector $\vec{p}(0)$ and rate matrix \mathbf{Q} can be obtained by solving the system of differential equations

$$\frac{d}{dt}\vec{p}(t) = \vec{p}(t) \mathbf{Q}. \quad (2.4)$$

The solution of the system of equation is

$$\vec{p}(t) = \vec{p}(0)e^{\mathbf{Q}t}. \quad (2.5)$$

One of the ways to evaluate Equation (2.5) is by computing the matrix exponential directly. Several methods for computing matrix exponentials are available [ML03, ML78]. However, most of them suffer from numerical instabilities.

A numerically stable method for evaluating Equation (2.5) is the *uniformization* or *randomization* method [Jen53, GM84, Gra91]. Using this method, the transient analysis of the CTMC proceeds by analyzing its uniformized CTMC. A uniformized CTMC can be obtained by (1) choosing a rate $\Lambda \geq \max_{i \in \mathcal{S}} E(i)$, and then (2) uniformizing the rate of residence of all states with Λ , namely by setting $\mathbf{Q}(i, i) = -\Lambda$, for all $i \in \mathcal{S}$. The original and the uniformized CTMCs are stochastically equivalent [BKHW05].

Now that all states have the same total outgoing rate, the residence times in all states have the same distribution. The only distinguishing feature of all states now is their transition probabilities

$$\mathbf{P} = \mathbf{I} + \frac{\mathbf{Q}}{\Lambda},$$

where \mathbf{I} is the identity matrix. Matrix \mathbf{P} together with the initial probability vector $\vec{p}(0)$ describe the *embedded* DTMC of the original CTMC.

Given the embedded DTMC and since $\mathbf{Q} = \Lambda(\mathbf{P} - \mathbf{I})$, the transient probabilities of the original CTMC ($\vec{p}(t)$) can now be written as

$$\begin{aligned} \vec{p}(t) &= \vec{p}(0)e^{\mathbf{Q}t} = \vec{p}(0)e^{\Lambda(\mathbf{P}-\mathbf{I})t} = \vec{p}(0)e^{-\Lambda t}e^{\Lambda \mathbf{P}t}, \\ &= \vec{p}(0)e^{-\Lambda t} \sum_{n=0}^{\infty} \frac{(\Lambda t)^n}{n!} \mathbf{P}^n. \end{aligned} \quad (2.6)$$

In the last equation, the matrix exponential is defined through the Taylor-MacLaurin expansion of the exponential function $e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$.

The term $\psi(\Lambda t, n) := e^{-\Lambda t} \frac{(\Lambda t)^n}{n!}$ is the density function of a Poisson¹ process $\{N_t \mid t \in \mathbb{R}_{\geq 0}\}$ with rate Λ . It is called a Poisson probability, and it gives the probability that exactly n transitions occur in the uniformized CTMC within t time units. A careful and stable method for computing the Poisson probabilities is described by FOX and GLYNN in [FG88].

Steady-State Analysis The steady-state analysis on a CTMC is used to determine the transient probabilities after an infinite time period has elapsed, *i.e.*, $\vec{v} = \lim_{t \rightarrow \infty} \vec{p}(t)$.

For finite and strongly connected CTMCs, $\lim_{t \rightarrow \infty} \vec{p}(t)$ always exists [Ros07, Hav98], and it corresponds to the time when the equilibrium has been reached, *i.e.*, when the transient probabilities no longer change. A CTMC is strongly connected if it is always possible to reach all states from any state. From Equation (2.4), the equilibrium

¹See Appendix A.1 for more information on Poisson processes and their relationship with CTMCs.

is reached when $\frac{d}{dt}\vec{p}(t) = \vec{0}$. Consequently, these steady-state probabilities can be described by a system of linear equations

$$\vec{v} \mathbf{Q} = \vec{0}, \quad \sum_{i \in \mathcal{S}} v_i = 1. \quad (2.7)$$

A comprehensive overview for efficient numerical methods for computing Equations (2.6) and (2.7) is available in [Ste94].

2.5 Phase-Type Distributions

Parameterization Definition Let $\{X_t \in \mathcal{S} \mid t \in \mathbb{R}_{\geq 0}\}$ be a Markov process defined on a discrete and finite state space \mathcal{S} of size $n+1$, for $n \in \mathbb{Z}_+$. The Markov process is a finite continuous-time Markov chain. If state $n+1$ is absorbing (*i.e.*, its total outgoing rate is equal to zero) and all other states are transient (*i.e.*, there is a nonzero probability that the state will never be visited once it is left, or equivalently, there exists at least one path from the state to the absorbing state), the infinitesimal generator matrix of the Markov chain can be written as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A} & \vec{A} \\ \vec{0} & 0 \end{bmatrix}. \quad (2.8)$$

Matrix \mathbf{A} is nonsingular because the first n states in the Markov chain are transient. The component $\mathbf{A}(i, j) \geq 0$, for $1 \leq i \leq n$, $1 \leq j \leq n$, and $i \neq j$, represents the rate of a transition from state i to state j . The component $\mathbf{A}(i, i) < 0$, for $1 \leq i \leq n$, is the negative sum of the rates of all transitions originating from state i .

Vector \vec{A} is a column vector, whose component \vec{A}_i , for $1 \leq i \leq n$, represents the rate of a transition from state i to the absorbing state. Since \mathbf{Q} is an infinitesimal generator matrix, $\vec{A} = -\mathbf{A}\vec{e}$.

The Markov chain is fully specified by the infinitesimal generator matrix \mathbf{Q} and an initial probability vector $\vec{\pi} = [\vec{\alpha}, \alpha_{n+1}]$, where $\vec{\alpha}$ is an n -dimensional row vector corresponding to the initial probabilities of the transient states, and α_{n+1} is the initial probability to be immediately in the absorbing state. Therefore, $\vec{\alpha}\vec{e} + \alpha_{n+1} = 1$.

Definition 2.7 ([Neu81]). A probability distribution on $\mathbb{R}_{\geq 0}$ is a phase-type (PH) distribution if and only if it is the distribution of the time until absorption in a continuous-time Markov chain described above.

A matrix of the form of \mathbf{A} is called a PH-generator. The pair $(\vec{\alpha}, \mathbf{A})$ is called the representation of the PH distribution and $\text{PH}(\vec{\alpha}, \mathbf{A})$ denotes the PH distribution of the representation $(\vec{\alpha}, \mathbf{A})$.

Example 2.8 (An absorbing CTMC representing a PH distribution). Consider the absorbing CTMC depicted in Figure 2.2. A CTMC is depicted in a similar way as a DTMC with the only difference that an edge in the graph is decorated with the rate of the transition instead of the transition probability. The CTMC models the stochastic behavior of the Hubble Space Telescope (HST) in terms of the failure behavior of its gyroscopes. A more detailed description of this problem can be found in [Her01].

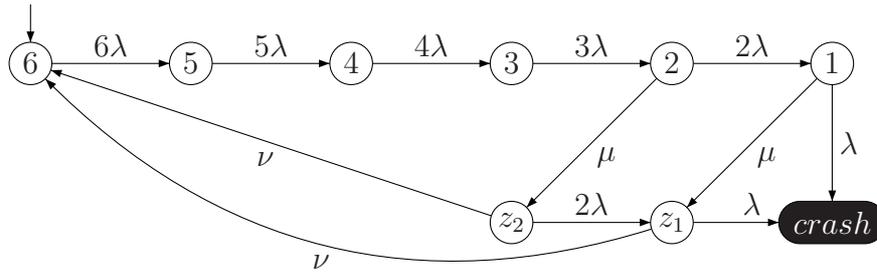


Figure 2.2: An Absorbing Continuous-Time Markov Chain

Initially, the HST had six functional gyroscopes when it was launched. Since then, however, one gyroscope after the other failed. The state labelled $i \in \{1, 2, \dots, 6\}$ represents the state where i gyroscopes are functioning properly. If there are only one or two functioning gyroscopes, the HST can be put into sleep modes (states z_1 and z_2 , respectively) and a reparation procedure is initiated. If none of the gyroscopes is operational, the HST may crash, which is represented by the black-shaded state. Note that from now onward, all absorbing states will always be depicted by a black-shaded state.

Each gyroscope is assumed to have an average lifetime of one year ($\lambda = 1$). The time it takes to bring the HST to sleep mode is around three and a half days ($\mu = 100$) and the reparation time requires about two months ($\nu = 6$). The reliability analysis of the HST boils down to computing the probability distribution of the time until the HST crashes. This is nothing more than a PH distribution with representation $(\vec{\gamma}, \mathbf{G})$, where

$$\vec{\gamma} = [1, 0, 0, 0, 0, 0, 0, 0], \text{ and}$$

$$\mathbf{G} = \begin{bmatrix} -6\lambda & 6\lambda & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -5\lambda & 5\lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -4\lambda & 4\lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3\lambda & 3\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2\lambda - \mu & 2\lambda & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & -\lambda - \mu & 0 & \mu \\ \nu & 0 & 0 & 0 & 0 & 0 & -2\lambda - \nu & 2\lambda \\ \nu & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda - \nu \end{bmatrix}.$$

Example 2.9 (Absorbing CTMCs that do not represent PH distributions). Not every absorbing CTMC can be used as a PH representation. Consider the absorbing CTMCs depicted in Figure 2.3.

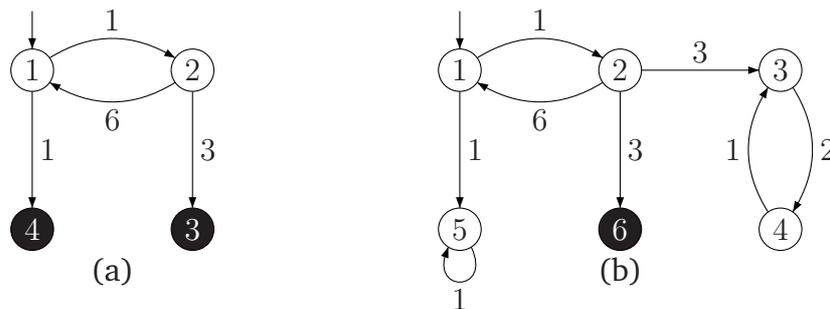


Figure 2.3: Not Phase-Type Representations

The absorbing CTMC in Figure 2.3(a) is not a PH representation because it has more than one absorbing state. The two states, of course, can always be collapsed or lumped together into one state, and thereby losing their identities; but this is not always desirable, for the identities might be important. The distribution of the time to absorption to state 4 in this CTMC, for instance, is not a probability distribution.

Figure 2.3(b), on the other hand, depicts an absorbing CTMC with a single absorbing state. However, this CTMC contains 3 non-transient states, namely states 3, 4 and 5. These are the states from which no path leading to the absorbing state exists. The probability of eventual absorption to state 6, then, is not equal to 1. The CTMC is, therefore, not a PH representation of any PH distribution.

Structural Definition The underlying absorbing CTMC of a PH representation with n transient states can be viewed as a tuple $\mathcal{M} = (\mathcal{S}, \mathbf{R}, \vec{\pi})$, where $\mathcal{S} = \{s_1, s_2, \dots, s_n, s_{n+1}\}$ is the state space, \mathbf{R} is a rate matrix $\mathbf{R} : (\mathcal{S} \times \mathcal{S}) \rightarrow \mathbb{R}_{\geq 0}$ of the underlying CTMC and $\vec{\pi} : \mathcal{S} \rightarrow [0, 1]$ is the initial probability distribution on the state space \mathcal{S} .

Now, a path σ in \mathcal{M} with the absorbing state s_a is an alternating finite sequence of states and their total outgoing rates

$$\sigma = s_1 \xrightarrow{E(s_1)} s_2 \xrightarrow{E(s_2)} s_3 \xrightarrow{E(s_3)} \dots \xrightarrow{E(s_{m-1})} s_m \xrightarrow{E(s_m)} s_{m+1} = s_a,$$

such that $\vec{\pi}(s_1) > 0$, $s_{m+1} = s_a$, and satisfying $\mathbf{R}(s_i, s_{i+1}) > 0$ for all $1 \leq i \leq m$. Let $Paths(\mathcal{M})$ denote the set of all paths in \mathcal{M} . With each path $\sigma \in Paths(\mathcal{M})$, a probability

$$P(\sigma) = \vec{\pi}(s_1) \prod_{i=1}^m \frac{\mathbf{R}(s_i, s_{i+1})}{E(s_i)},$$

is associated. This probability is called the *occurrence probability* of the path. Intuitively, the occurrence probability gives the probability of observing a particular path if the Markov process is run until it hits the absorbing state.

Example 2.10. The underlying CTMC of the PH representation $(\vec{\gamma}, \mathbf{G})$ in Example 2.8 is $\mathcal{M} = (\mathcal{S}, \mathbf{R}, \vec{\pi})$, where $\mathcal{S} = \{6, 5, 4, 3, 2, 1, z_1, z_2, crash\}$,

$$\mathbf{R} = \begin{bmatrix} 0 & 6\lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5\lambda & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4\lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3\lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\lambda & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu & \lambda \\ \nu & 0 & 0 & 0 & 0 & 0 & 0 & 2\lambda & 0 \\ \nu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and $\vec{\pi} = [\vec{\gamma}, 0]$. One of the shortest paths $\sigma_s \in Paths(\mathcal{M})$ in the CTMC is

$$\sigma_s = 6 \xrightarrow{6\lambda} 5 \xrightarrow{5\lambda} 4 \xrightarrow{4\lambda} 3 \xrightarrow{3\lambda} 2 \xrightarrow{2\lambda+\mu} 1 \xrightarrow{\lambda+\mu} crash.$$

The occurrence probability of path σ_s is

$$\begin{aligned} P(\sigma_s) &= \vec{\pi}(6) \cdot \frac{\mathbf{R}(6,5)}{E(6)} \cdot \frac{\mathbf{R}(5,4)}{E(5)} \cdot \frac{\mathbf{R}(4,3)}{E(4)} \cdot \frac{\mathbf{R}(3,2)}{E(3)} \cdot \frac{\mathbf{R}(2,1)}{E(2)} \cdot \frac{\mathbf{R}(1, \text{crash})}{E(1)}, \\ &= 1 \cdot \frac{6\lambda}{6\lambda} \cdot \frac{5\lambda}{5\lambda} \cdot \frac{4\lambda}{4\lambda} \cdot \frac{3\lambda}{3\lambda} \cdot \frac{2\lambda}{2\lambda + \mu} \cdot \frac{\lambda}{\lambda + \mu} = \frac{2\lambda^2}{2\lambda^2 + 3\lambda\mu + \mu^2} = \frac{2}{10302}. \end{aligned}$$

Let a PH-generator of the form

$$\begin{bmatrix} -\lambda_1 & \lambda_1 & 0 & \cdots & 0 \\ 0 & -\lambda_2 & \lambda_2 & \cdots & 0 \\ 0 & 0 & -\lambda_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -\lambda_n \end{bmatrix}$$

be denoted by $\mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)$. A PH representation having PH-generator of the form $\mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)$ is called a *bidiagonal representation*.

A path $\sigma \in \text{Paths}(\mathcal{M})$ can be regarded as a PH distribution with representation $(\vec{e}_1, \mathbf{Bi}(E(s_1), E(s_2), \dots, E(s_m)))$, where vector \vec{e}_1 is a unit vector at the first component of appropriate dimension. Let $\text{PH}(\sigma)$ be the PH distribution of the path, namely $\text{PH}(\sigma) = \text{PH}(\vec{e}_1, \mathbf{Bi}(E(s_1), E(s_2), \dots, E(s_m)))$. Now, we can put forward the following proposition.

Proposition 2.11. *Let the underlying absorbing continuous-time Markov chain of a phase-type representation $(\vec{\alpha}, \mathbf{A})$ be $\mathcal{M} = (\mathcal{S}, \mathbf{R}, \vec{\pi})$. Then*

$$\text{PH}(\vec{\alpha}, \mathbf{A}) = \sum_{\sigma \in \text{Paths}(\mathcal{M})} P(\sigma) \text{PH}(\sigma).$$

Intuitively, a PH representation is characterized by the collection of its paths and the probability with which each path occurs in the representation. It is straightforward that $\sum_{\sigma \in \text{Paths}(\mathcal{M})} P(\sigma) = 1$. Therefore, a PH representation is a convex combination of its paths.

2.5.1 General Notions and Concepts

Cumulative and Density Functions Let $\vec{p}(t)$ be the transient probability vector of the transient states of a Markov chain representing PH distribution $\text{PH}(\vec{\alpha}, \mathbf{A})$, namely $\vec{p}_i(t)$ is the probability that the process is in state s_i , for $1 \leq i \leq n$, at time t . Based on Equation (2.4), this transient probability vector satisfies the following system of differential equations

$$\frac{d}{dt} \vec{p}(t) = \vec{p}(t) \mathbf{A}, \quad t \in \mathbb{R}_+, \quad (2.9)$$

with initial condition $\vec{p}(0) = \vec{\alpha}$.

The solution of Equation (2.9) is $\vec{p}(t) = \vec{\alpha} e^{\mathbf{A}t}$. Since the probability that the process is in state s_{n+1} (i.e., being in the absorbing state) at time t is the same as the probability of not being in any of the states $\{s_1, s_2, \dots, s_n\}$ at time t , the distribution function of the time until absorption in the Markov chain (hence of PH distribution) is

$$F(t) = 1 - \vec{p}(t) \vec{e} = 1 - \vec{\alpha} e^{\mathbf{A}t} \vec{e}, \quad t \in \mathbb{R}_{\geq 0}. \quad (2.10)$$

The PH distribution is completely characterized by this (cumulative) distribution function.

From Equation (2.10), the probability density function of the PH distribution is

$$f(t) = \frac{d}{dt}F(t) = -\vec{\alpha}e^{\mathbf{A}t}\mathbf{A}\vec{e} = \vec{\alpha}e^{\mathbf{A}t}\vec{A}, \quad t \in \mathbb{R}_+. \quad (2.11)$$

The PH distribution has a mass of α_{n+1} at $t = 0$. This means that the CTMC starts immediately in the absorbing state with probability α_{n+1} .

Example 2.12. Figure 2.4 depicts the curve of the distribution and respectively the density functions of $\text{PH}(\vec{\gamma}, \mathbf{G})$ from Example 2.8.

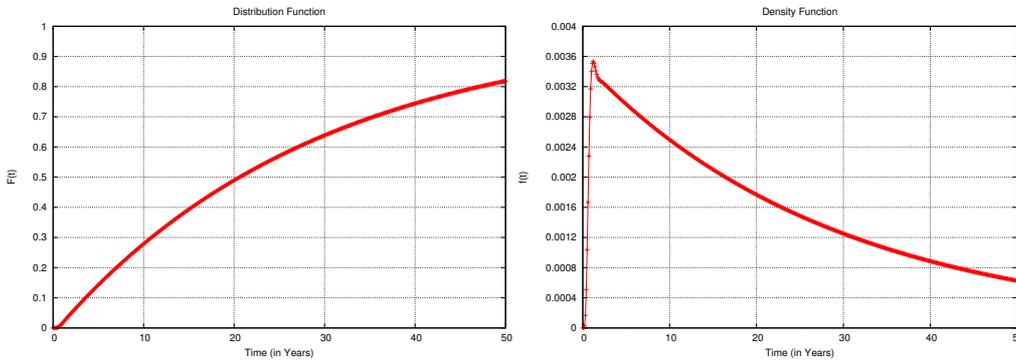


Figure 2.4: PH Distribution (Left) and Density (Right) Functions

A PH distribution has no mass at $t = 0$ if $\alpha_{n+1} = 0$. A point mass at zero, namely a probability distribution with a distribution function $F(t)$ such that $F(0) = 1$, is a trivial PH distribution, denoted by δ .

Irreducible Representations All PH representations we are dealing with in this thesis are assumed to be irreducible. An irreducible representation is characterized as the representation in which no transient state is visited with probability zero for the specified initial probability distribution.

The irreducibility of a PH representation $(\vec{\alpha}, \mathbf{A})$ can be determined as follows. Let the underlying absorbing CTMC of $(\vec{\alpha}, \mathbf{A})$ be $\mathcal{M} = (\mathcal{S}, \mathbf{R}, \vec{\pi})$. For each state $s_i \in \mathcal{S}$ such that $\vec{\pi}(s_i) > 0$, we create a new transition with an arbitrary rate $\lambda \in \mathbb{R}_+$ from the absorbing state to state s_i . Now, PH representation $(\vec{\alpha}, \mathbf{A})$ is irreducible if and only if the newly modified CTMC is strongly connected. Recall that a CTMC is strongly connected if it is always possible to reach all states from any state.

If $(\vec{\alpha}, \mathbf{A})$ is an irreducible representation, each component of vector $\vec{p}(t) = \vec{\alpha}e^{\mathbf{A}t}$ is strictly positive for $t \in \mathbb{R}_+$. On the other hand, if representation $(\vec{\alpha}, \mathbf{A})$ is not irreducible, some components of $\vec{p}(t)$ will be zero for all $t \in \mathbb{R}_{\geq 0}$. In this case, all columns of $\vec{\alpha}$ and all rows and columns of \mathbf{A} that correspond to the components of vector $\vec{p}(t)$ that are zero for all $t \in \mathbb{R}_{\geq 0}$ can be deleted to obtain the irreducible representation [Neu81].

Laplace-Stieltjes Transforms and Moments Aside from its distribution function, a PH distribution is also completely characterized by its Laplace-Stieltjes transform (LST). The LST of the PH distribution is

$$\tilde{f}(s) = \int_{-\infty}^{\infty} e^{-st} dF(t) = \vec{\alpha}(s\mathbf{I} - \mathbf{A})^{-1}\vec{A} + \alpha_{n+1}, \quad s \in \mathbb{R}_{\geq 0}, \quad (2.12)$$

where \mathbf{I} is the n -dimensional identity matrix. This transform is a rational function, *i.e.*,

$$\tilde{f}(s) = \vec{\alpha}(s\mathbf{I} - \mathbf{A})^{-1}\vec{A} + \alpha_{n+1} = \frac{p(s)}{q(s)},$$

for some polynomials $p(s)$ and $q(s) \neq 0$. When the LST is expressed in irreducible ratio, the degree of the numerator $p(s)$ is no more than the degree of the denominator $q(s)$. The degrees of the two polynomials are equal only when vector $\vec{\alpha}$ is sub-stochastic but not stochastic [O'C90].

The LST of an exponential distribution with rate $\lambda \in \mathbb{R}_+$ is given by

$$\tilde{f}(s) = \frac{\lambda}{s + \lambda}. \quad (2.13)$$

In the rest of the thesis, we sometimes refer to the LST of a PH representation. In this case, we are actually referring to the LST of the PH distribution of the representation.

Example 2.13. The LST of PH distribution $\text{PH}(\vec{\gamma}, \mathbf{G})$ described in Example 2.8 is given by $\tilde{f}(s) = \frac{p(s)}{q(s)}$, where

$$\begin{aligned} p(s) &= 720(s + 132)(s + 83), \text{ and} \\ q(s) &= s^8 + 236s^7 + 17446s^6 + 433436s^5 + 5232949s^4 + 34788584s^3 \\ &\quad + 129998724s^2 + 232835184s + 7888320. \end{aligned}$$

The rational function $\tilde{f}(s)$ shown above is expressed in irreducible ratio. The degree of its numerator is 2, while the degree of its denominator is 8.

Let $\text{PH}(\vec{\alpha}, \mathbf{A})$ be the distribution of a random variable X . The k -th non-central moment of $\text{PH}(\vec{\alpha}, \mathbf{A})$ is given by

$$m_k = E[X^k] = (-1)^k k! \vec{\alpha} \mathbf{A}^{-k} \vec{e}, \quad (2.14)$$

where $E[X]$ is the expected value of random variable X . The relationship between the LST and the moments is described by

$$m_k = (-1)^k \left. \frac{d^k \tilde{f}(s)}{ds^k} \right|_{s=0}. \quad (2.15)$$

2.5.2 Order and Degree

For a PH distribution with representation $(\vec{\alpha}, \mathbf{A})$, the *size of the representation* is defined to be the dimension of matrix \mathbf{A} . The degree of the denominator polynomial of its LST expressed in irreducible ratio—which is no more than the dimension of the matrix—is called the *algebraic degree of the distribution*. The zeros of the denominator polynomial

are called the *poles* of the LST. Sometimes we also call them the poles of the PH distribution.

It is known [Neu81, O’C90] that a given PH distribution has more than one representation. The size of a *minimal representation*—namely a representation with the least number of states—is called the *order of the phase-type distribution*. Note that in standard literature, the size of a representation is called the order of the representation. We choose to call it “size” to avoid confusion with the order of a PH distribution.

The order of a PH distribution may be different from its algebraic degree but it is no less than its algebraic degree [O’C90]. The following lemma is straightforward.

Lemma 2.14. *A phase-type representation whose size is equal to the algebraic degree of its phase-type distribution is a minimal representation.*

In this case, the order of the distribution is then simply given by the size of the representation.

Example 2.15. *We continue using the PH distribution described in Example 2.8, whose representation is $(\vec{\gamma}, \mathbf{G})$. The size of the representation is the dimension of \mathbf{G} , i.e., 8. Now, the algebraic degree of the PH distribution is given by the degree of the denominator polynomial of its LST, which in this case is also 8. The poles of the LST are given by the zeros of $q(s)$, and they are $s = -3.329344 \pm 4.307561i$, $s = -8.753829 \pm 2.893837i$, $s = -0.034539$, $s = -8.799086$, $s = -101.00005$ and $s = -101.999975$.*

Since the size of the representation is equal to the algebraic degree of the distribution, $(\vec{\gamma}, \mathbf{G})$ is a minimal representation. Therefore, the order of the distribution $\text{PH}(\vec{\gamma}, \mathbf{G})$ is also 8.

In this thesis, PH distributions whose order is equal to their algebraic degree play an important role. The following definition simplifies the way we refer to them.

Definition 2.16. *A phase-type distribution is called ideal if and only if its order is equal to its algebraic degree.*

2.5.3 Dual Representations

Let $\{X_t \mid t \in \mathbb{R}_{\geq 0}\}$ be an absorbing Markov process representing a PH distribution and let τ be a random variable denoting its absorption time.

Definition 2.17 ([Kel79]). *The dual or the time-reversal representation of the absorbing Markov process $\{X_t \mid t \in \mathbb{R}_{\geq 0}\}$ is given by an absorbing Markov process $\{X_{\tau-t} \mid t \in \mathbb{R}_{\geq 0}\}$.*

The relationship between the two processes can be described intuitively as follows: the probability of being in state s at time t in one Markov process is equal to the probability of being in state s at time $\tau - t$ in the time-reversal Markov process and vice versa.

Theorem 2.18 ([CC93, CM02]). *Given a PH representation $(\vec{\alpha}, \mathbf{A})$, then its dual representation is $(\vec{\beta}, \mathbf{B})$ such that*

$$\vec{\beta} = \vec{A}^\top \mathbf{M} \quad \text{and} \quad \mathbf{B} = \mathbf{M}^{-1} \mathbf{A}^\top \mathbf{M}, \quad (2.16)$$

where $\mathbf{M} = \text{diag}(\vec{m})$ is a diagonal matrix whose diagonal components are formed by the components of vector $\vec{m} = -\vec{\alpha} \mathbf{A}^{-1}$.

Recall that \vec{A} in Equation (2.16) is a column vector representing the rates of the transitions from all transient states to the absorbing state. From Equation (2.16), we can derive that

$$\vec{B} = \mathbf{M}^{-1}\vec{\alpha}^\top. \quad (2.17)$$

Example 2.19. Figure 2.5 depicts the dual or time-reversal of the representation shown in Figure 2.2. Several transition rates and initial probabilities are written as rational numbers rather than as expressions of λ , μ , and/or ν , because they are simply too long to be printed. Recall from Example 2.8 that $\lambda = 1$, $\mu = 100$, and $\nu = 6$. For the remaining transitions, the rates are expressed in λ , μ , and/or ν so that the correspondence between the original and the dual representations can be observed.

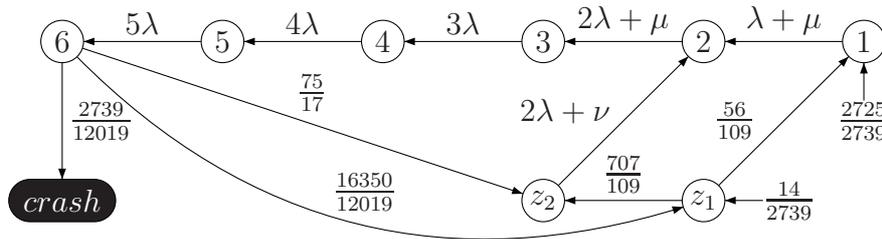


Figure 2.5: The Dual of the Representation in Figure 2.2

From the form of Equation (2.16) as well as from the example, several properties of the dual representation can be inferred: (1) the dual representation has the same number of states as the original one; (2) there is a one-to-one correspondence between the state spaces of the two representations: a state in Figure 2.2, for instance, corresponds to a state with the same identity in Figure 2.5, and these states have the same total outgoing rate; (3) the dual representation is the reverse of the original representation, *i.e.*, the direction of each transition that does not end in the absorbing state is reversed in the dual, each state with a transition to the absorbing state in the original becomes a state with nonzero initial probability in the dual, and conversely each state with nonzero initial probability in the original becomes a state with a direct transition to the absorbing state in the dual; and (4) both representations represent the same PH distribution.

2.5.4 Characterization

All PH distributions have rational Laplace-Stieltjes transforms. However, not every probability distribution with rational LST is a PH distribution. For instance, some members of matrix-exponential distributions [AB96, AO98, Fac03, HZ07b]—which we will discuss further in Section 2.8—are not PH distribution. The following characterization theorem establishes the necessary and sufficient conditions for a probability distribution to be a PH distribution.

Theorem 2.20 ([O’C90]). *A probability distribution defined on $\mathbb{R}_{\geq 0}$ is a phase-type distribution if and only if it is the point mass at zero (δ), or it satisfies the following conditions:*

1. its density function is strictly positive on \mathbb{R}_+ , and

2. its Laplace-Stieltjes transform is a rational function having a real pole whose real part is strictly larger than the real parts of all other poles.

Therefore, any probability distribution whose density function is equal to zero at some $t \in \mathbb{R}_+$ is not a PH distribution. Moreover, a PH distribution must have a rational LST with a real pole whose real part is strictly larger than the real parts of all other poles. This means that the largest real pole is unique. An LST with a complex pole whose real part is the largest is not the LST of a PH distribution, since there are then two poles having the largest real part, for its conjugate is also a pole. From Example 2.15, for instance, we know that the probability distribution associated with representation $(\vec{\gamma}, \mathbf{G})$ has an LST with a unique real pole having the largest real part, namely $s = -0.034539$. Hence, it is a PH distribution.

Example 2.21 (Not a phase-type). *A probability distribution with density function $f(t) = e^{-t} + e^{-t} \cos(t)$ is not a PH distribution. The LST of the distribution is*

$$\tilde{f}(s) = \frac{2s^2 + 4s + 3}{(s + 1)(s^2 + 2s + 2)},$$

and the poles are $s = -1$, $s = -1 + \iota$, and $s = -1 - \iota$. Hence, the LST has no unique real pole whose real part is the largest among the real parts of all poles.

Any probability distribution that fails to satisfy any of the two conditions is not a PH distribution. It is interesting to note, however, that as a PH distribution more nearly fails to satisfy the second condition—namely its LST has poles whose real part approaches the largest one—more states are required to represent it [O’C91].

2.5.5 Closure Properties

The set of all PH distributions is *closed* under a certain operation if the application of the operation on any PH distributions produces a PH distribution. The set of all such operations defines the closure properties of PH distributions. Some of these closure properties can be found in [Neu81]. Closure properties under four operations—convolution, minimum, maximum, and mixture—are required in this thesis.

Definition 2.22. *Let $p \in \mathbb{R}_{\geq 0}$ and $0 \leq p \leq 1$. For two distribution functions $F(t)$ and $G(t)$, let the distribution functions:*

- (a) $\text{con}(F(t), G(t)) = [F * G](t) = \int_0^t F(t-x)G(x)dx,$
- (b) $\text{min}(F(t), G(t)) = 1 - (1 - F(t))(1 - G(t)),$
- (c) $\text{max}(F(t), G(t)) = F(t)G(t),$ and
- (d) $\text{mix}(pF(t), (1-p)G(t)) = pF(t) + (1-p)G(t).$

We refer to these functions as the convolution, minimum, maximum, and mixture,² respectively, of the two distributions.

²Note that there is an inconsistency in our naming: convolution and mixture operations proceed on the level of distributions (and they are actually summation operations on the level of random variables), while minimum and maximum operations proceed on the level of random variables. Despite this inconsistency, we decide to use this nomenclature consistently throughout this thesis.

The following theorem establishes that the set of PH distributions is closed under the four operations. The theorem also provides the representation of the PH distribution produced by each of the operations.

Theorem 2.23 ([Neu81]). *Let $F(t)$ and $G(t)$ be two phase-type distributions with representations $(\vec{\alpha}, \mathbf{A})$ and $(\vec{\beta}, \mathbf{B})$ of size m and n , respectively. Then:*

- (a) $\text{con}(F(t), G(t))$ is a phase-type distribution with representation $(\vec{\delta}, \mathbf{D})$ of size $m+n$, where

$$\vec{\delta} = [\vec{\alpha}, \alpha_{m+1}\vec{\beta}] \quad \text{and} \quad \mathbf{D} = \begin{bmatrix} \mathbf{A} & \vec{A}\vec{\beta} \\ \vec{0} & \mathbf{B} \end{bmatrix}. \quad (2.18)$$

- (b) $\text{min}(F(t), G(t))$ is a phase-type distribution whose representation is³

$$(\vec{\alpha} \otimes \vec{\beta}, \mathbf{A} \oplus \mathbf{B}), \quad (2.19)$$

of size mn .

- (c) $\text{max}(F(t), G(t))$ is a phase-type distribution with representation $(\vec{\delta}, \mathbf{D})$ of size $mn+m+n$, where

$$\vec{\delta} = \left[\vec{\alpha} \otimes \vec{\beta}, \beta_{n+1}\vec{\alpha}, \alpha_{m+1}\vec{\beta} \right] \quad \text{and} \\ \mathbf{D} = \begin{bmatrix} \mathbf{A} \oplus \mathbf{B} & \mathbf{I}_m \otimes \vec{B} & \vec{A} \otimes \mathbf{I}_n \\ \mathbf{0} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B} \end{bmatrix}. \quad (2.20)$$

- (d) $\text{mix}(pF(t), (1-p)G(t))$ is a phase-type distribution with representation $(\vec{\delta}, \mathbf{D})$ of size $m+n$, where

$$\vec{\delta} = [p\vec{\alpha}, (1-p)\vec{\beta}] \quad \text{and} \quad \mathbf{D} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}. \quad (2.21)$$

In the following definition, we abuse the notation of convolution, minimum, maximum, and mixture functions to not only operate on PH distributions, but also to operate on the representations of PH distributions.

Definition 2.24. *For two phase-type representations $(\vec{\alpha}, \mathbf{A})$ and $(\vec{\beta}, \mathbf{B})$, let the functions:*

- (a) $\text{con}((\vec{\alpha}, \mathbf{A}), (\vec{\beta}, \mathbf{B})) = (\vec{\delta}, \mathbf{D})$ as given in Equation (2.18),
 (b) $\text{min}((\vec{\alpha}, \mathbf{A}), (\vec{\beta}, \mathbf{B})) = (\vec{\alpha} \otimes \vec{\beta}, \mathbf{A} \oplus \mathbf{B})$ as given in Equation (2.19),
 (c) $\text{max}((\vec{\alpha}, \mathbf{A}), (\vec{\beta}, \mathbf{B})) = (\vec{\delta}, \mathbf{D})$ as given in Equation (2.20), and
 (d) $\text{mix}(p(\vec{\alpha}, \mathbf{A}), (1-p)(\vec{\beta}, \mathbf{B})) = (\vec{\delta}, \mathbf{D})$ as given in Equation (2.21).

We refer to these functions as the convolution, minimum, maximum, and mixture, respectively, of the two phase-type representations.

³See Appendix A.2 for the definitions of the Kronecker product (\otimes) and sum (\oplus).

The convolution, minimum, and maximum operations on PH distributions (and also on PH representations) are commutative and associative.

Let $F(t)$ be a probability distribution on $\mathbb{R}_{\geq 0}$, and let X be a random variable governed by a geometric distribution with parameter $p \in \mathbb{R}_{\geq 0}$, $0 \leq p < 1$. We define $F(t)^{(p)}$ as the distribution of the sum of $X + 1$ independent and identically distributed random variables with distribution $F(t)$, namely

$$F(t)^{(p)} := (1 - p) (F(t) + pF(t) * F(t) + p^2 F(t) * F(t) * F(t) + \dots),$$

where $*$ denotes the convolution of two probability distributions.

Intuitively, operation $F(t)^{(p)}$ is a kind of tail-recursion on probability distributions. If $F(t)$ is a PH distribution, the operation produces another PH distribution whose underlying Markov process repeatedly performs a trial when it is about to hit the absorbing state in order to decide whether to restart the process with probability $1 - p$, or to hit the absorbing state with probability p . The set of PH distributions is also closed under this operation.

Figure 2.6 illustrates the effects of the operation on the representation of a PH distribution. Figure 2.6(a) depicts a PH representation of $F(t)$, and Figure 2.6(b) depicts a PH representation of $F(t)^{(\frac{1}{3})}$.

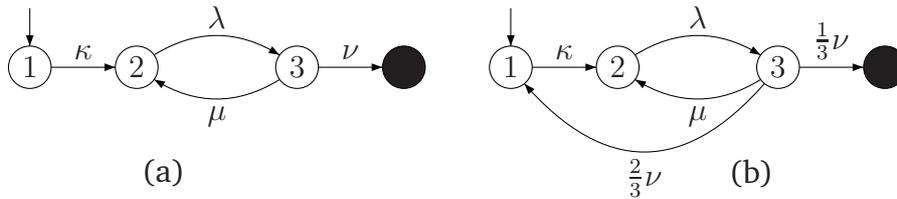


Figure 2.6: Operation $F(t)^{(p)}$ on Phase-Type Representation

Some of the closure properties are enough to characterize the set of all PH distributions [MO92], as described in the following theorem.

Theorem 2.25 ([MO92]). *The family of phase-type distributions is the smallest family of distributions on $\mathbb{R}_{\geq 0}$ that:*

1. contains the point mass at zero (δ) and all exponential distributions,
2. is closed under finite mixture and convolution,
3. is closed under the operation $F(t)^{(p)}$, for $0 \leq p < 1$.

The theorem establishes that the whole family of PH distributions can be generated by the point mass at zero and exponential distributions together with the finite mixture, convolution, and recursion (restarting) operations.

2.6 Acyclic Phase-Type Distributions

An interesting subset of PH distributions is the family of acyclic phase-type (APH) distributions. The family can be identified by the form of their representation matrices.

An APH distribution must have at least one representation that, under some permutation of its state space, has a triangular representation matrix. In other words, an APH distribution must have at least one representation whose associated graph contains no cycle or is acyclic. Such representations are called APH representations.

A *triangular minimal representation* of an APH distribution is an APH representation with the least number of states. In a similar way we defined the order of a PH distribution, we can also define the triangular order of an APH distribution [O’C93]. The *triangular order* of an APH distribution is the size of its triangular minimal representation. As shown in [BHM87], the triangular order and the order of an APH distribution are not always the same. The triangular order of an APH distribution may exceed, but cannot be smaller than, its order.

Lemma 2.26. *An acyclic phase-type representation whose size is equal to the algebraic degree of its phase-type distribution is a triangular minimal representation.*

In this case, the triangular order of the distribution is then simply given by the size of the representation. Also, the order of the distribution is then equal to the triangular order of the distribution.

The following definition simplifies the way we refer to APH distributions whose triangular order is equal to their algebraic degree.

Definition 2.27. *An acyclic phase-type distribution is called triangular ideal if and only if its triangular order is equal to its algebraic degree.*

2.6.1 Characterization

Similar to the case of the general PH distributions, APH distributions can be characterized by the properties of their density functions and LSTs.

Theorem 2.28 ([Cum82, O’C91]). *A probability distribution defined on $\mathbb{R}_{\geq 0}$ is an acyclic phase-type distribution if and only if it is the point mass at zero (δ), or it satisfies the following conditions:*

1. *its density function is strictly positive on \mathbb{R}_+ , and*
2. *its Laplace-Stieltjes transform is a rational function having only real poles.*

Actually, [Cum82] and [O’C91] proved a stronger result than Theorem 2.28, namely that PH distributions satisfying those conditions are not only acyclic but also have bidagonal PH representations. This will be discussed in more detail in Section 3.2.

Example 2.29 (An acyclic phase-type distribution). *Consider a PH distribution with representation $(\vec{\gamma}, \mathbf{G})$, where $\vec{\gamma} = [\frac{1}{2}, \frac{1}{2}, 0, 0]$ and*

$$\mathbf{G} = \begin{bmatrix} -4 & 2 & 1 & 0 \\ 0 & -3 & 0 & 2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

$\text{PH}(\vec{\gamma}, \mathbf{G})$ is an APH distribution, because its LST

$$\tilde{f}(s) = \frac{s^2 + 5.5s + 8}{(s + 4)(s + 2)(s + 1)}, \quad (2.22)$$

is a rational function, and has only real poles. The poles are $s = -1$, $s = -2$ and $s = -4$. One of the APH representations of the APH distribution is $(\bar{\gamma}', \mathbf{G}')$, where $\bar{\gamma}' = [\frac{1}{4}, \frac{5}{16}, \frac{7}{16}]$, and

$$\mathbf{G}' = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -2 & 2 \\ 0 & 0 & -4 \end{bmatrix}.$$

2.6.2 Closure Properties

The following theorem describes the closure characterization of the set of APH distributions.

Theorem 2.30 ([AL82]). *The family of acyclic phase-type distributions is the smallest family of distributions on $\mathbb{R}_{\geq 0}$ that:*

1. contains the point mass at zero (δ) and all exponential distributions,
2. is closed under finite mixture and convolution.

Therefore, the family of APH distributions can be generated by the point mass at zero and exponential distributions together with the finite mixture and convolution operations. This result will be clearer in the next chapter when we introduce several canonical forms of APH representations. The family of APH distributions is also closed under the minimum and maximum operations.

2.6.3 Erlang and Hypoexponential Distributions

Two of the most frequently used APH distributions in this thesis are Erlang distributions and hypoexponential distributions.

Erlang Distributions Erlang distributions are formed by convolutions of several exponential distributions with the same rate. They are named after AGNER K. ERLANG (1878-1929), a Danish mathematician and a pioneer in traffic engineering.

Definition 2.31. *A random variable X is distributed according to an Erlang distribution with rate $\lambda \in \mathbb{R}_+$ and phase $k \in \mathbb{Z}_+$ if its distribution function is given by*

$$F(t) = \Pr(X \leq t) = \begin{cases} 1 - \sum_{i=0}^{k-1} \frac{(\lambda t)^i}{i!} e^{-\lambda t}, & t \in \mathbb{R}_{\geq 0}, \\ 0, & \text{otherwise.} \end{cases} \quad (2.23)$$

The probability density function of the Erlang distribution is then

$$f(t) = \frac{\lambda^k t^{k-1} e^{-\lambda t}}{(k-1)!}, \quad t \in \mathbb{R}_+.$$

An Erlang distribution with rate λ and phase k is denoted by $\text{Erl}(\lambda, k)$.

A representation of $\text{Erl}(\lambda, k)$ is simply a concatenation of k states, each with total outgoing rate λ as depicted in Figure 2.7(a). Such a representation is called an *Erlang representation*.

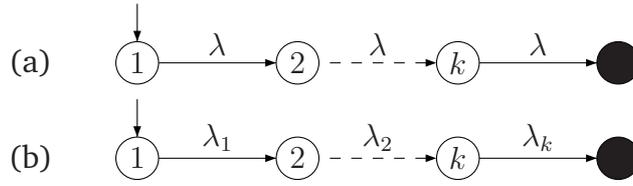


Figure 2.7: Erlang and Hypoexponential Representations

Hypoexponential Distributions Hypoexponential distributions are formed by convolutions of several exponential distributions with possibly different rates. Hence the set of Erlang distributions is a subset of hypoexponential distributions.

Let λ_i , for $1 \leq i \leq k$ and $k \in \mathbb{Z}_+$, be the rates of exponential distributions forming a hypoexponential distribution. The rates are not necessarily distinct from each other. A representation of the hypoexponential distribution is simply a concatenation of k states, each having total outgoing rate λ_i , for $1 \leq i \leq k$, as depicted in Figure 2.7(b). Such a representation is called a *hypoexponential representation*. A hypoexponential representation is a bidiagonal representation that starts only from the first state, *i.e.*, $(\vec{e}_1, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_k))$.

Lemma 2.32. *Every hypoexponential representation (hence, also every Erlang representation) is a minimal representation.*

Proof. Consider the representation in Figure 2.7(b). The LST of the distribution of the hypoexponential representation is

$$\tilde{f}(s) = \frac{\lambda_1}{(s + \lambda_1)} \frac{\lambda_2}{(s + \lambda_2)} \cdots \frac{\lambda_k}{(s + \lambda_k)},$$

namely it is the multiplication of the LSTs of all exponential distributions. The LST is in irreducible ratio form. We can then conclude that the algebraic degree of the hypoexponential representation is k . This means that the hypoexponential representation is minimal, because its size is equal to its algebraic degree. \square

2.7 The Polytope of Phase-Type Distributions

Let $\text{PH}(\mathbf{A})$ be the set of all PH distributions with representation $(\vec{\alpha}, \mathbf{A})$, where $\vec{\alpha}$ ranges over all sub-stochastic vectors of dimension n , *i.e.*,

$$\text{PH}(\mathbf{A}) = \{\text{PH}(\vec{\alpha}, \mathbf{A}) \mid \vec{\alpha} \in \mathbb{R}_{\geq 0}^n, \vec{\alpha} \vec{e} \leq 1\}.$$

The point mass at zero (δ) always resides in any $\text{PH}(\mathbf{A})$, having representation $(\vec{0}, \mathbf{A})$.

$\text{PH}(\mathbf{A})$ can be regarded as a subset of a vector space of *signed measures*⁴ on $\mathbb{R}_{\geq 0}$ having rational LSTs [O'C90]. Using \vec{e}_i to denote the i -th unit vector of \mathbb{R}^n , $\text{PH}(\mathbf{A})$ is the convex hull of vectors δ and $\text{PH}(\vec{e}_i, \mathbf{A})$, for $1 \leq i \leq n$. Hence each of the $n + 1$ distributions is a vertex of $\text{PH}(\mathbf{A})$. We refer to this convex hull as the *polytope*⁵ of the PH-generator \mathbf{A} .

⁴Signed measure is a measure that can have negative values.

⁵See Appendix A.3 for a brief description of some required concepts from the field of convex analysis.

Example 2.33. Let $(\vec{\gamma}, \mathbf{G})$ be a PH representation, where $\vec{\gamma} = [\frac{1}{2}, \frac{1}{4}, \frac{1}{4}]$ and

$$\mathbf{G} = \begin{bmatrix} -21 & 11 & 7 \\ 0 & -16 & 10 \\ 0 & 0 & -2 \end{bmatrix}.$$

We illustrate the polytope of PH-generator \mathbf{G} and several concepts related to it in Figure 2.8. Sub-figure (a) depicts the polytope in a 3-dimensional coordinate system.

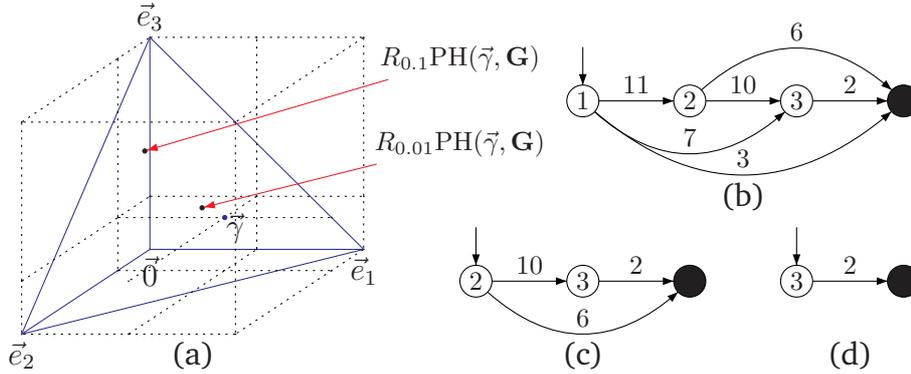


Figure 2.8: The Polytope $\text{PH}(\mathbf{G})$ and Its Unit Components

Each point in the coordinate system represents an initial vector—which can be a non-valid initial probability vector—of PH-generator \mathbf{G} . The origin corresponds to the point mass at zero (δ) and the unit vectors on the axes x , y and z correspond to PH distributions $\text{PH}(\vec{e}_1, \mathbf{G})$, $\text{PH}(\vec{e}_2, \mathbf{G})$ and $\text{PH}(\vec{e}_3, \mathbf{G})$, respectively. The convex hull of these four vectors forms a polyhedron $0\vec{e}_1\vec{e}_2\vec{e}_3$. Every point inside the polyhedron represents a valid initial probability vector (namely a sub-stochastic vector) of PH-generator \mathbf{G} , and it is a convex combination of vectors 0 , \vec{e}_1 , \vec{e}_2 and \vec{e}_3 . The plane $\vec{e}_1\vec{e}_2\vec{e}_3$ is called the face of the polytope. (We can generalize this for the n -dimensional space: the hyperplane $\vec{e}_1\vec{e}_2 \cdots \vec{e}_n$ is also called the face of the corresponding polytope). Every point on this plane represents a stochastic vector, hence corresponds to a PH representation that starts immediately in the absorbing state with probability 0. The PH distribution $\text{PH}(\vec{\gamma}, \mathbf{G})$ is on this plane, represented by point $\vec{\gamma}$ in the sub-figure.

Sub-figures (b), (c) and (d) depict the PH representations of $\text{PH}(\vec{e}_1, \mathbf{G})$, $\text{PH}(\vec{e}_2, \mathbf{G})$ and $\text{PH}(\vec{e}_3, \mathbf{G})$, respectively.

2.7.1 Residual-Life Operator

Intuitively, the residual-life operator R_t is an operator whose application on a PH distribution amounts to running the underlying Markov process of the PH distribution t time units. At time t , a new PH distribution is produced, which reflects the current standing of the underlying Markov process. The operator R_t , then, describes the trajectory that is traversed by the PH distribution in its polytope as t changes.

Definition 2.34 ([O’C90]). Let Z be the vector space of all Borel signed measures.⁶ For

⁶The σ -algebra generated by the open sets of \mathbb{R} is called the Borel σ -algebra, denoted by \mathcal{B} . A measure defined on the Borel σ -algebra is called a Borel measure. A set is a Borel subset if $A \subseteq \Omega$ and $A \in \mathcal{B}$ [Rud87]. A signed measure on \mathcal{B} is called a Borel signed measure.

all $\mu \in Z$, the residual-life operator $R_t, t \in \mathbb{R}_{\geq 0}$, is defined by

$$R_t\mu(0) = \mu([0, t]) \quad \text{and} \quad R_t\mu(E) = \mu(E + t),$$

where E is a Borel subset of $[0, +\infty)$ and $E + t$ is the translation of E by t units to the right.

The residual-life operator translates the mass associated with the given measure by t units to the left of real line. The mass that because of the translation now resides in the interval $(-\infty, 0)$ is accumulated on point $t = 0$. Given a PH distribution $\text{PH}(\vec{\alpha}, \mathbf{A})$, the application of the residual-life operator R_t on the distribution is

$$R_t\text{PH}(\vec{\alpha}, \mathbf{A}) = \text{PH}(\vec{\alpha}e^{\mathbf{A}t}, \mathbf{A}).$$

For each sub-stochastic vector $\vec{\alpha}$ and $t \in \mathbb{R}_{\geq 0}$, vector $\vec{\alpha}' := \vec{\alpha}e^{\mathbf{A}t}$ is sub-stochastic. Hence all possible initial probability distributions of a PH-generator \mathbf{A} and their trajectories towards absorption (namely to the point mass at zero δ) reside inside the polytope \mathbf{A} . The dimension of the smallest affine subspace containing the whole trajectory starting at point $\vec{\alpha}$ is equal to the algebraic degree of $\text{PH}(\vec{\alpha}, \mathbf{A})$ [O'C90].

Example 2.35. We refer back to Figure 2.8 in Example 2.33. As mentioned before, point $\vec{\gamma}$ in the figure represents PH distribution $\text{PH}(\vec{\gamma}, \mathbf{G})$. The application of the residual-life operator R_t forms the trajectory that is traversed by the distribution as t advances. In the figure, we depict two points that correspond to $R_{0.01}\text{PH}(\vec{\gamma}, \mathbf{G})$ and $R_{0.1}\text{PH}(\vec{\gamma}, \mathbf{G})$, and whose coordinates are approximately $[0.405, 0.258, 0.302]$ and $[0.061, 0.138, 0.525]$, respectively. $R_{0.01}\text{PH}(\vec{\gamma}, \mathbf{G})$ and $R_{0.1}\text{PH}(\vec{\gamma}, \mathbf{G})$ also reflect the transient probabilities of the underlying CTMC at time $t = 0.01$ and $t = 0.1$, respectively.

Previously, we mentioned that the dimension of the smallest affine subspace containing the whole trajectory starting at point $\vec{\gamma}$ is equal to the algebraic degree of $\text{PH}(\vec{\gamma}, \mathbf{G})$. We can conclude that the algebraic degree of the PH distribution in our example is no less than 3 because the whole trajectory of $\text{PH}(\vec{\gamma}, \mathbf{G})$ is contained in a 3-dimensional affine subspace. We know that this is the smallest affine subspace, because points $\vec{0}$, $\vec{\gamma}$, $R_{0.01}\text{PH}(\vec{\gamma}, \mathbf{G})$ and $R_{0.1}\text{PH}(\vec{\gamma}, \mathbf{G})$ do not reside in the same plane.

If point $\vec{\gamma}$ were to be on the plane $\vec{0}\vec{e}_1\vec{e}_2$, for instance, and its trajectory for all $t \in \mathbb{R}_{\geq 0}$ were to be confined in this plane, then we would be able to conclude that the algebraic degree of $\text{PH}(\vec{\gamma}, \mathbf{G})$ was 2.

2.7.2 Simplicity and Majorization

The notion of PH-simplicity was first formalized in [O'C89] and it is closely related to the notion of simplicity in convex analysis.

Definition 2.36. A PH-generator \mathbf{A} (of dimension n) is PH-simple if and only if for any two n -dimensional sub-stochastic vectors $\vec{\alpha}_1$ and $\vec{\alpha}_2$, where $\vec{\alpha}_1 \neq \vec{\alpha}_2$, $\text{PH}(\vec{\alpha}_1, \mathbf{A}) \neq \text{PH}(\vec{\alpha}_2, \mathbf{A})$.

A PH-generator \mathbf{A} is PH-simple if all PH distributions in $\text{PH}(\mathbf{A})$ are pairwise distinct. This is equivalent to saying that each distribution in $\text{PH}(\mathbf{A})$ is a unique convex combination of the $n + 1$ distributions δ and $\text{PH}(\vec{e}_i, \mathbf{A})$, for $1 \leq i \leq n$.

The term ‘‘simple’’ comes from the fact that if PH-generator \mathbf{A} is PH-simple, then the $n + 1$ points δ and $\text{PH}(\vec{e}_i, \mathbf{A})$, for $1 \leq i \leq n$, form an n -dimensional simplex (see

Appendix A.3). It follows that a PH-generator is not PH-simple if the $n + 1$ points are not linearly independent of each other, or equivalently, if its polytope is a subset of an $(n - 1)$ -dimensional affine subspace [Roc70].

Example 2.37. *Returning to Example 2.33, we can determine whether PH-generator \mathbf{G} is PH-simple by showing that δ and $\text{PH}(\vec{e}_i, \mathbf{G})$, for $1 \leq i \leq 3$, are linearly independent of each other. One of the ways to do this is in the LST domain. Using the representations depicted in Figure 2.8(b)–(d), the LSTs of δ , $\text{PH}(\vec{e}_1, \mathbf{G})$, $\text{PH}(\vec{e}_2, \mathbf{G})$, and $\text{PH}(\vec{e}_3, \mathbf{G})$ are calculated as follows*

$$1, \frac{3s^2 + 134s + 672}{(s + 2)(s + 16)(s + 21)}, \frac{6s + 32}{(s + 2)(s + 16)}, \text{ and } \frac{2}{(s + 2)}.$$

Since the LSTs are linearly independent, PH-generator \mathbf{G} is PH-simple.

Example 2.38 (Not PH-simple). PH-generator

$$\mathbf{G} = \begin{bmatrix} -3 & 1 \\ 0 & -2 \end{bmatrix}$$

is not PH-simple because $\text{PH}([1, 0], \mathbf{G}) = \text{PH}([0, 1], \mathbf{G})$. Both representations represents an exponential distribution with rate 2.

A simpler and more efficient method to determine the PH-simplicity of a PH-generator is described in the following theorem.

Theorem 2.39 ([O’C89]). *A PH-generator \mathbf{A} of dimension n is PH-simple if and only if vectors $\mathbf{A}^i \vec{e}$, for $0 \leq i \leq n - 1$, span (or equivalently form a basis of) the vector space \mathbb{R}^n .*

Therefore, a PH-generator \mathbf{A} of dimension n is PH-simple if and only if matrix

$$[\vec{A} \quad \mathbf{A}\vec{A} \quad \mathbf{A}^2\vec{A} \quad \dots \quad \mathbf{A}^{n-1}\vec{A}], \quad (2.24)$$

has rank n . Recall that \vec{A} in Equation (2.24) is a column vector representing the transition rates from all transient states to the absorbing state.

Since a PH distribution has many representations, it is of interest to obtain and use a representation that has a simple form or some desirable properties. In the following, we define a relation between two PH-generators based on the sets of PH distributions they can generate.

Definition 2.40. *Let \mathbf{A} and \mathbf{B} be two PH-generators. Then \mathbf{B} PH-majorizes \mathbf{A} if and only if $\text{PH}(\mathbf{A}) \subseteq \text{PH}(\mathbf{B})$.*

The following theorem gives the necessary conditions for PH-majorization when PH-generator \mathbf{B} is PH-simple.

Theorem 2.41 ([O’C89]). *Let \mathbf{A} and \mathbf{B} be two PH-generators and let \mathbf{B} be PH-simple. Then*

1. \mathbf{B} PH-majorizes \mathbf{A} if and only if there exists a nonnegative matrix \mathbf{P} with unit row-sums (i.e., $\mathbf{P}\vec{e} = \vec{e}$) such that $\mathbf{A}\mathbf{P} = \mathbf{P}\mathbf{B}$, and
2. If \mathbf{B} PH-majorizes \mathbf{A} , then there exists exactly one matrix \mathbf{P} with unit row-sums such that $\mathbf{A}\mathbf{P} = \mathbf{P}\mathbf{B}$. Furthermore, \mathbf{P} is of full rank if and only if \mathbf{A} is also PH-simple.

2.7.3 Geometrical View of APH Representations

Let $(\vec{\alpha}, \mathbf{A})$ be an APH representation of size n . Moreover, let $-\lambda_1, -\lambda_2, \dots, -\lambda_n$ be the eigenvalues of PH-generator \mathbf{A} . Since PH-generator \mathbf{A} is a triangular matrix, its eigenvalues are simply given by its diagonal components. Furthermore, by Theorem 2.28, the poles of $\text{PH}(\vec{\alpha}, \mathbf{A})$ are among these eigenvalues. Without loss of generality, assume that $\lambda_n > \lambda_{n-1} > \dots > \lambda_1 > 0$.

Diagonal Representations An important observation [HZ06a] is that an eigenvector $\vec{b}^{[i]}$ of \mathbf{A} that corresponds to an eigenvalue $-\lambda_i$, which means that $\vec{b}^{[i]}\mathbf{A} = -\lambda_i\vec{b}^{[i]}$, can be associated with an exponential distribution with rate λ_i as follows. If $\vec{b}^{[i]}\vec{e} \neq 0$, we normalize $\vec{b}^{[i]}$ to have a unit sum (i.e., $\vec{b}^{[i]}\vec{e} = 1$), namely, for all $1 \leq i \leq n$, let

$$\vec{\beta}^{[i]} = \begin{cases} \frac{\vec{b}^{[i]}}{\vec{b}^{[i]}\vec{e}}, & \text{if } \vec{b}^{[i]}\vec{e} \neq 0, \\ \vec{b}^{[i]}, & \text{otherwise.} \end{cases}$$

Now, since $\vec{\beta}^{[i]}$ is an eigenvector of \mathbf{A} with eigenvalue $-\lambda_i$, $\vec{\beta}^{[i]}e^{\mathbf{A}t}\vec{e} = e^{-\lambda_i t}$. Therefore, eigenvector $\vec{\beta}^{[i]}$ represents the exponential distribution with rate λ_i in the polytope $\text{PH}(\mathbf{A})$, even though it is not a sub-stochastic vector.

In general, we have n eigenvectors $\vec{\beta}^{[i]}$, for $1 \leq i \leq n$, each corresponding to the exponential distribution with rate λ_i . These n points lie in the same hyperplane as the face of polytope $\text{PH}(\mathbf{A})$. The convex hull of $\vec{0}$ and these n points form a new polytope $\text{PH}(\mathbf{D})$, where \mathbf{D} is a PH-generator whose diagonal components are $-\lambda_i$, for $1 \leq i \leq n$, and all other components are equal to zero. A PH-generator of this form is called a *diagonal* PH-generator. A PH representation with a diagonal PH-generator is a mixture of exponential distributions, and it is called a *hyperexponential* representation.

Example 2.42. We continue using Example 2.33. Figure 2.8(a) depicts the polytope of PH-generator \mathbf{G} . The eigenvectors of \mathbf{G} are $\vec{\gamma}^{[1]} = [0, 0, 1]$ for eigenvalue -2 , $\vec{\gamma}^{[2]} = [0, \frac{-7}{5}, 1]$ for eigenvalue -16 , and $\vec{\gamma}^{[3]} = [\frac{19}{15}, \frac{-209}{75}, 1]$ for eigenvalue -21 . Normalized, they become

$$\vec{\gamma}^{[1]} = [0, 0, 1], \vec{\gamma}^{[2]} = [0, \frac{7}{2}, \frac{-5}{2}], \text{ and } \vec{\gamma}^{[3]} = [\frac{-95}{39}, \frac{209}{39}, \frac{-75}{39}].$$

Figure 2.9 depicts these three eigenvectors in the coordinate system of $\text{PH}(\mathbf{G})$.

The convex hull of $\vec{0}$, $\vec{\gamma}^{[1]}$, $\vec{\gamma}^{[2]}$, and $\vec{\gamma}^{[3]}$ forms a polytope $\text{PH}(\mathbf{D})$, where

$$\mathbf{D} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & -16 & 0 \\ 0 & 0 & -21 \end{bmatrix}.$$

If some vector $\vec{\gamma}' = a_1\vec{\gamma}^{[1]} + a_2\vec{\gamma}^{[2]} + a_3\vec{\gamma}^{[3]}$, where $a_i \geq 0$, and $0 < \sum_{i=1}^3 a_i \leq 1$, then $\text{PH}(\vec{\gamma}', \mathbf{G})$ has a diagonal PH representation $([a_1, a_2, a_3], \mathbf{D})$. Equivalently, the intersection of the two polytopes represents the set of the APH distributions that can be represented by using both PH-generators. In our example, the intersection forms the plane $\vec{0}\vec{e}_2\vec{e}_3$. Hence, our special point $\vec{\gamma}$ does not have a representation in polytope $\text{PH}(\mathbf{D})$.

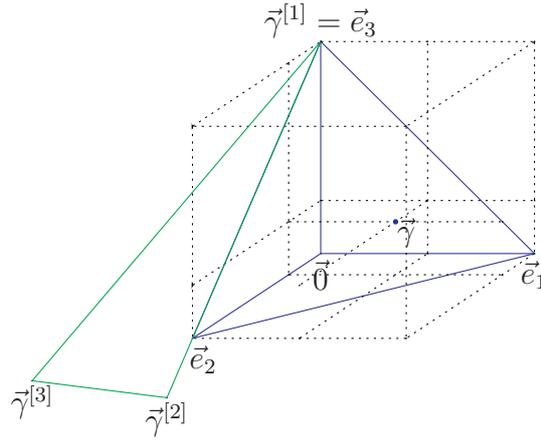


Figure 2.9: The Polytopes PH(G) and PH(D)

Bidiagonal Representations In the following, we describe a method to expand the face of polytope PH(D) to include linear combinations of eigenvectors $\vec{\beta}^{[i]}$, for $1 \leq i \leq n$, that still represent probability distributions [DL82]. Recall that eigenvector $\vec{\beta}^{[i]}$ represents an exponential distribution with rate λ_i . Let \mathcal{V} be an n -dimensional real vector space generated by $\vec{\beta}^{[i]}$, for $1 \leq i \leq n$. We work with $\{\vec{\beta}^{[1]}, \vec{\beta}^{[2]}, \dots, \vec{\beta}^{[n]}\}$ as the basis of \mathcal{V} and we denote the subset of \mathcal{V} that are probability distributions by \mathcal{C}_n .

The first step is recognizing that all convex combinations of two arbitrary vectors $\vec{\beta}^{[i]}$ and $\vec{\beta}^{[j]}$, for $1 \leq i, j \leq n$ —namely those vectors that lie in the line segment between the two vectors—are also probability distributions. Moreover, certain parts of the extension of this line are still probability distributions as described in the following theorem.

Theorem 2.43 ([DL82]). *For $i, j \in \{1, 2, \dots, n\}$ and $i > j$, vector $\kappa\vec{\beta}^{[j]} + (1 - \kappa)\vec{\beta}^{[i]}$ is an element of \mathcal{C}_n if and only if $\frac{\lambda_i}{\lambda_i - \lambda_j} \leq \kappa \leq 1$.*

Consider Figure 2.10, which is based on the discussion in Example 2.42, and will be used as a running example for the rest of the section.

The line segment between vectors $\vec{\gamma}^{[3]}$ and $\vec{\gamma}^{[2]}$, for instance, can be extended and the extension still represents probability distributions. The extreme point of the extension, namely when $\kappa = \frac{2}{16-2} = \frac{1}{7}$, corresponds to the vector $\vec{\gamma}^{[2,3]}$. In a similar fashion, we can obtain vectors $\vec{\gamma}^{[1,2]}$ and $\vec{\gamma}^{[1,3]}$.

In general, let vector $\kappa\vec{\beta}^{[j]} + (1 - \kappa)\vec{\beta}^{[i]}$ in Theorem 2.43 when $\kappa = \frac{\lambda_i}{\lambda_i - \lambda_j}$ be denoted by $\vec{\beta}^{[j,i]}$. In terms of probability distributions, vector $\vec{\beta}^{[j,i]}$ corresponds to the convolution of two exponential distributions with rates λ_j and λ_i , respectively (we know this by looking at the distribution function). We generalize this notation: Let $\psi \subseteq \{1, 2, \dots, n\}$ and $\psi \neq \emptyset$. Let vector $\vec{\beta}^{[\psi]}$ ⁷ denote the vector that represents the convolution of $|\psi|$ exponential distributions with rates λ_i , for each $i \in \psi$, respectively, in the vector space \mathcal{V} . Let Ψ denote the collection of all such ψ 's. The following theorem helps us locating these vectors.

Theorem 2.44 ([DL82]). *For $i, j \in \{1, 2, \dots, n\}$ and $i > j$*

$$\vec{\beta}^{[j, \dots, i-1, i+1, \dots, n]} \in \text{conv}(\{\vec{\beta}^{[j+1, \dots, n]}, \vec{\beta}^{[j, \dots, n]}\}).$$

⁷We abuse the vector index notation also to be able to indicate a set notation.

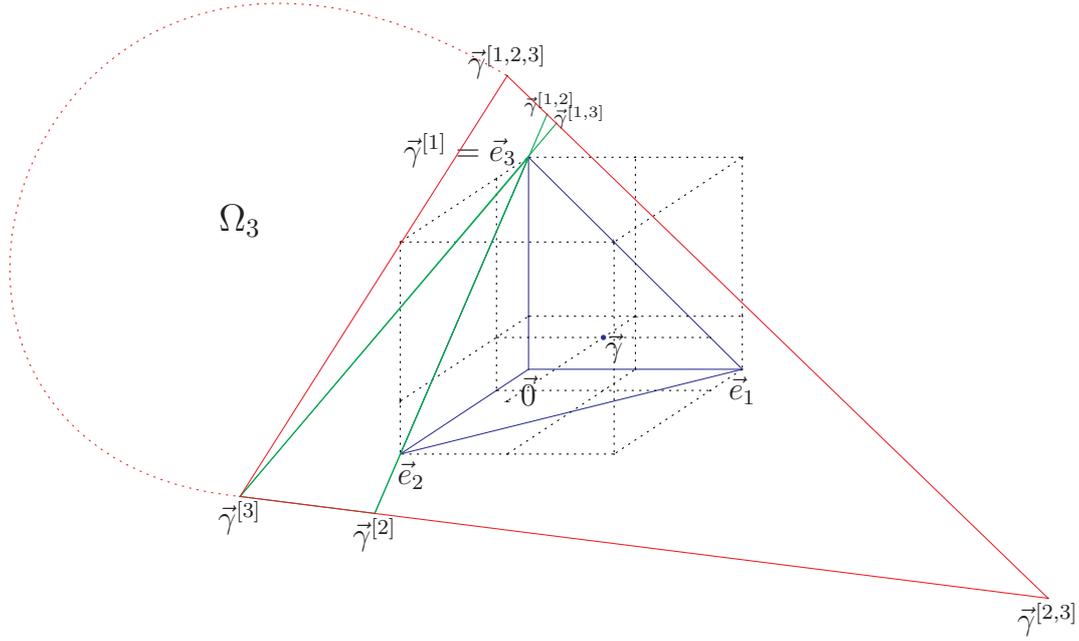


Figure 2.10: The Polytopes PH(**G**), PH(**D**) and PH(**Bi**)

Set $\text{conv}(A)$ is the set of all convex combinations of all members of set A (see Appendix A.3.2). In the case of Theorem 2.44, this set forms the line between the two vectors. Based on the theorem, we know for certain that vector $\vec{\gamma}^{[1,3]}$, in our running example, lies on the line between vectors $\vec{\gamma}^{[2,3]}$ and $\vec{\gamma}^{[1,2,3]}$.

The following theorem summarizes the result of the expansion and describes the structure of the subset \mathcal{C}_n .

Theorem 2.45 ([DL82]). *For each $\psi \in \Psi$, vector $\vec{\beta}^{[\psi]}$ is on the boundary of \mathcal{C}_n , except for when $\psi = \{1\}$, which resides inside it. Furthermore, vectors*

$$\vec{\beta}^{[n]}, \vec{\beta}^{[n-1,n]}, \dots, \vec{\beta}^{[2,\dots,n]}, \vec{\beta}^{[1,\dots,n]}$$

are the extreme points of \mathcal{C}_n .

In conclusion, we have n vectors $\vec{\beta}^{[n]}, \vec{\beta}^{[n-1,n]}, \dots, \vec{\beta}^{[2,\dots,n]}, \vec{\beta}^{[1,\dots,n]}$, each corresponding to an extreme point of \mathcal{C}_n , and representing a convolution of several exponential distributions. These n vectors lie in the same hyperplane as the face of the polytope PH(**D**). Now, the convex hull of $\vec{0}$ and these n vectors form a new polytope PH(**Bi**), where **Bi** is a bidiagonal PH-generator.

Example 2.46. *The convex hull of $\vec{0}$, $\vec{\gamma}^{[3]}$, $\vec{\gamma}^{[2,3]}$, and $\vec{\gamma}^{[1,2,3]}$ forms a polytope PH(**Bi**), where*

$$\mathbf{Bi} = \begin{bmatrix} -2 & 2 & 0 \\ 0 & -16 & 16 \\ 0 & 0 & -21 \end{bmatrix}.$$

The polytope is depicted in Figure 2.10.

*If some vector $\vec{\gamma}' = a_1\vec{\gamma}^{[1,2,3]} + a_2\vec{\gamma}^{[2,3]} + a_3\vec{\gamma}^{[3]}$, where $a_i \geq 0$, and $0 < \sum_{i=1}^3 a_i \leq 1$, then PH($\vec{\gamma}'$, **G**) has a bidiagonal PH representation $([a_1, a_2, a_3], \mathbf{Bi})$. In the example, polytope PH(**G**) is a subset of PH(**Bi**). Thus, our special point $\vec{\gamma}$ must have a representation in polytope PH(**Bi**). Indeed, vector $\vec{\gamma}$ is equal to vector $[\frac{109}{168}, \frac{31}{168}, \frac{1}{6}]$ in PH(**Bi**).*

Actually, the structure of \mathcal{C}_n is not completely described by Theorem 2.45 and Theorem 2.43. In Figure 2.10, we also depict the region Ω_3 , which is part of \mathcal{C}_3 , but lies outside the polytope $\vec{0}\vec{\gamma}^{[1,2,3]}\vec{\gamma}^{[2,3]}\vec{\gamma}^{[3]}$ (we call this polytope Θ_3). The construction of the curve delimiting the region Ω_3 is described in [DL82]. Each vector in this region is formed by a linear combination of a vector in Θ_3 and a vector on the plane $\vec{0}\vec{\gamma}^{[1,2,3]}\vec{\gamma}^{[3]}$, such that the resulting vector still represents a probability distribution, even though it must now have negative entries. Although the vector cannot have an APH representation with PH-generator \mathbf{Bi} , it can be represented by a PH-generator of larger sizes [O’C91].

In a similar fashion, we can also describe the region Ω_n and Θ_n in \mathcal{C}_n . Every PH distribution in Θ_n is of algebraic degree and order of no more than n . On the other hand, every PH distribution in Ω_n is of algebraic degree n , but of order more than n .

2.8 Matrix-Exponential Distributions

As mentioned in Section 2.5.4, not every probability distribution with rational LST is a PH distribution. Some matrix-exponential (ME) distributions, which have rational LSTs, for instance, are not PH distributions. The set of all ME distributions is described in the following definition. An excellent introduction to ME distributions is available in [AO98].

Definition 2.47 ([AB96, AO98, Fac03, HZ07b]). *A random variable X is distributed according to a matrix-exponential distribution if its distribution function is given by*

$$F(t) = \Pr(X \leq t) = \begin{cases} 1 - \vec{\alpha}e^{\mathbf{A}t}\vec{\omega}, & t \in \mathbb{R}_{\geq 0}, \\ 0, & \text{otherwise,} \end{cases} \quad (2.25)$$

where $\vec{\alpha}$ and $\vec{\omega}$ are vectors of dimension n , \mathbf{A} is a square matrix of dimension n , and they all may have complex components.

The tuple $(\vec{\alpha}, \mathbf{A}, \vec{\omega})$ is called the representation of the matrix-exponential distribution, and $\text{ME}(\vec{\alpha}, \mathbf{A}, \vec{\omega})$ denotes the matrix-exponential distribution of the representation.

The notions of the size of the representation, the order of the distribution, the algebraic degree, and the poles are defined in a similar fashion for ME distributions as for PH distributions. Unlike PH distributions, the problem of the order of ME distributions, and hence also the problem of their minimal representations, have been solved in [AB96].

Example 2.48. $([\frac{27}{55}, -\frac{4}{55}, \frac{32}{55}], \mathbf{Bi}(2, 16, 21), \vec{e})$ is an ME representation of size 3. As will be shown in Chapter 5, this ME distribution also has APH representations of size 4 and 5. This ME distribution is then also a PH distribution.

From Equation (2.25), we are certain that the set of ME distributions is a superset of PH distributions. The set of ME distributions is also precisely the set of probability distributions whose Laplace-Stieltjes transforms are rational [LR89, AB96]. An interesting question arises: how much “larger” is the family of ME distributions compared to the family of PH distributions?

Theorem 2.49 ([Fac03]). *In the set of all matrix-exponential distributions of algebraic degree n , the set of matrix-exponential distributions that are not phase-type distributions has measure zero.*

Thus, the set of all ME distributions is only “slightly larger” than the set of all PH distributions. Or equivalently, almost all ME distributions of a particular algebraic degree are actually PH distributions.

Chapter 3

Reducing APH Representations

In this chapter, we propose an algorithm to reduce the size of the matrix representation of an APH distribution without altering the distribution.

Related Work The study of APH representations begins with the canonical forms of CUMANI's [Cum82] in 1982. He recognized the importance of the ordered bidiagonal representations both for their simplicity and canonicity. In the same paper, he also sketched an algorithm to transform an APH representation to its ordered bidiagonal representation. The transformations to other canonical forms can easily be accomplished once the ordered bidiagonal representation is obtained. Although the algorithm performs poorly, it opens a way to future improvements. O'CONNOR proposed such an improvement in [O'C89]. His algorithm is inspired by the notions of PH-simplicity and PH-majorization, which he put forward in the same paper.

In 1991, O'CONNOR characterized the class of APH distributions in terms of the density functions and LSTs [O'C91]. He also reproved CUMANI's results using the invariant polytopes method [O'C90]. In 2006, HE and ZHANG introduced an improved algorithm for the transformation, called spectral polynomial algorithm [HZ06b]. Their new algorithm avoids several weak points of the previous ones, such as the need for matrix inversions and the restriction on the PH-simplicity of the original APH representation.

O'CONNOR in [O'C93] presented the notion and theory of triangular order, namely the order of PH distributions having the smallest triangular matrix representations. Although the theory enables us to reason about the minimal representations and their characteristics, it is not algorithmic. In [HZ07a], HE and ZHANG provided an algorithm for computing minimal ordered bidiagonal representations of APH distributions.

The algorithm of HE and ZHANG starts by immediately transforming a given APH distribution to a representation that only contains states that represent the poles of the LST of the distribution. This representation is not necessarily a PH distribution, but certainly a matrix-exponential distribution. If this is the case, another state and its total outgoing rate are determined and appended to the representation. This is performed one by one until a PH representation is obtained. The first PH representation found is a minimal representation.

The algorithm of HE and ZHANG involves solving systems of non-linear equations when additional states and their total outgoing rates are to be determined. Since non-linear programming is difficult, the practicality of this algorithm for large models is not obvious, and has not been investigated so far.

Contribution The algorithm developed in this chapter, on the other hand, is of cubic complexity in the size of the state space, and only involves standard numerical computations. The goal is to reduce the state space of the original representation one state by one state. The algorithm returns a smaller or equal size representation than the original one. However, unlike the algorithm of HE and ZHANG, the result is not guaranteed to be minimal.

The algorithm starts by transforming a given APH representation to its ordered bidiagonal representation. This transformation does not increase the number of states. It then proceeds by removing “unnecessary” states while maintaining the resulting representation to be phase-type. The removal of a state involves solving a system of linear equations. This removal is repeated until no more removal is possible.

This chapter contributes an efficient algorithm to reduce the size of APH representations. The algorithm is easy to implement and straightforward to parallelize. It only consists of vector-matrix multiplications and the solutions of well-conditioned systems of linear equations. (A system of linear equations is well-conditioned if small perturbations in the system do not produce relatively large changes in the exact solution [Mey04]). Furthermore, because we are dealing with bidiagonal representations, these operations can be carried out even more efficiently. To illustrate the effectiveness of our approach, we study a dynamic fault tree [MDCS98] scenario with a prototype implementation of the algorithm. As we will discuss, the results are promising. Aside from this, we provide insight into the relation between lumping on absorbing CTMCs and our reduction algorithm. We show that on some special cases, our reduction algorithm is contained in the weak-bisimulation lumping on the dual of ordered bidiagonal representations.

Structure The chapter is organized as follows: In Section 3.1, we discuss two canonical forms of APH representations. In Section 3.2, we employ a small example to show step by step how to transform an APH representation to its ordered bidiagonal representation in order to clarify and emphasize the main idea behind the transformation. Three existing algorithms for the transformation are discussed in Section 3.3. Section 3.4 is the core of the chapter, where we present the reduction algorithm for APH representations. An example illustrating the use of the reduction algorithm is presented in Section 3.5. In Section 3.6, we discuss how our reduction algorithm relates to the weak-bisimulation-based lumping algorithm for absorbing CTMCs. In Section 3.7, we summarize and conclude the chapter.

3.1 Acyclic Canonical Forms

Three canonical forms of APH representations are presented in [Cum82]. They are called “forms” because every APH representation can be transformed into any of them without altering its stochastic behavior, *i.e.*, its distribution. The term “canonical” suggests uniqueness and simpleness. They are unique since in each of the forms, there is only one way (up to the permutations of states with similar total outgoing rate) to represent a given APH distribution. They are simple because the forms have straightforward and easy-to-understand structure, and, moreover, they may have fewer transitions than the original representation.

Example 3.1 (Acyclic Representation). Consider an APH representation depicted in Figure 3.1. The representation has 3 states, named 1, 2, and 3 with total outgoing rates 4, 3, and 1, respectively. The figure also shows a fourth state, the absorbing one.

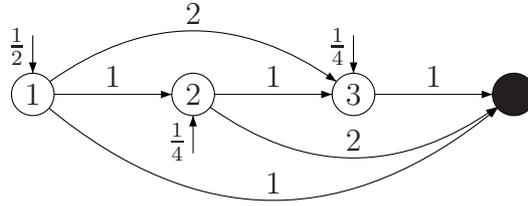


Figure 3.1: An Acyclic Phase-Type Representation

We will use this as the running example throughout our description of the canonical forms.

In this section, we describe two canonical forms: the ordered bidiagonal and the Cox representations.

3.1.1 Ordered Bidiagonal Representation

The first canonical form is specified in the following theorem.

Theorem 3.2. Let $(\vec{\alpha}, \mathbf{A})$ be an acyclic phase-type representation. Let $-\lambda_1, -\lambda_2, \dots, -\lambda_n$ be the eigenvalues of \mathbf{A} , and, without loss of generality, assume $\lambda_n \geq \lambda_{n-1} \geq \dots \geq \lambda_1 > 0$. Then there exists a unique ordered bidiagonal representation $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$ such that

$$\text{PH}(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)) = \text{PH}(\vec{\alpha}, \mathbf{A}).$$

This is a combined restatement of Theorem 1 and Lemma 1 in [Cum82].

Since the total outgoing rates of all states in $\mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)$ are of a particular order (namely ascending), this canonical form is called *ordered bidiagonal representation*.

Example 3.3 (Ordered Bidiagonal Representation). Figure 3.2 depicts the ordered bidiagonal representation of the APH representation shown in Figure 3.1. This representation has a nice and simple structure: the states are ordered by their total outgoing rates, and each of the states has only one outgoing transition. The branching structure and the initial probability distribution of the original representation are now contained in the initial probability distribution of the ordered bidiagonal representation.

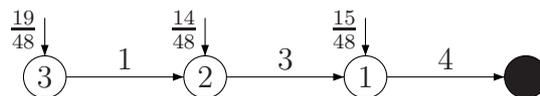


Figure 3.2: The Ordered Bidiagonal Representation of Figure 3.1

An important property of ordered bidiagonal representations is the PH-simplicity of their PH-generators. Let PH-generator $\mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)$ be an ordered bidiagonal representation. For $1 \leq i \leq n$, let X_i be a random variable distributed according to an exponential distribution with rate λ_i .

Theorem 3.4 ([O'C89]). *The polytope of PH-generator $\mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)$ is an n -simplex, i.e., a simplex of dimension n (see Appendix A.3). Therefore, the PH-generator is PH-simple. PH distributions $\text{PH}(\vec{e}_i, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$, for $1 \leq i \leq n$, and the point mass at zero (δ) are the vertices of the simplex. The distributions of random variables*

$$\sum_{i=j}^n X_i, \quad 1 \leq j \leq n,$$

correspond to the n PH distributions, respectively.

3.1.2 Cox Representation

For $1 \leq i \leq n-1$, let $0 \leq p_i < 1$. Let a PH-generator of the form

$$\begin{bmatrix} -\lambda_1 & p_1\lambda_1 & 0 & \cdots & 0 \\ 0 & -\lambda_2 & p_2\lambda_2 & \cdots & 0 \\ 0 & 0 & -\lambda_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -\lambda_n \end{bmatrix}$$

be denoted by $\mathbf{Cx}([\lambda_1, p_1], [\lambda_2, p_2], \dots, \lambda_n)$. We call a PH representation with a PH-generator $\mathbf{Cx}([\lambda_1, p_1], [\lambda_2, p_2], \dots, \lambda_n)$ a *Cox representation*. The following theorem describes the transformation from an ordered bidiagonal to a Cox representation.

Theorem 3.5 ([Cum82]). *Let $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$ be an ordered bidiagonal representation, and let the initial probability vector $\vec{\delta} = [\vec{\beta}\vec{e}, 0, \dots, 0]$. Then*

$$\text{PH}(\vec{\delta}, \mathbf{Cx}([\lambda_n, x_n], [\lambda_{n-1}, x_{n-1}], \dots, \lambda_1)) = \text{PH}(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$$

if and only if

$$x_i = 1 - \beta_i \prod_{j=i+1}^n \frac{1}{x_j}, \quad \text{for } 2 \leq i \leq n. \quad (3.1)$$

Example 3.6 (Cox Representation). *Figure 3.3 depicts the Cox representation of the APH representation shown in Figure 3.1. The name is due to DAVID R. COX who, in [Cox55], defined a probability distribution with close resemblance to the representation.*

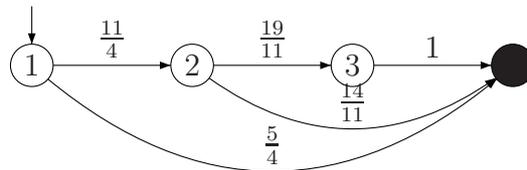


Figure 3.3: The Cox Representation of Figure 3.1

There is exactly one initial state in every Cox representation. Every state, apart from the absorbing state, has a transition to the next state, possibly a transition to the absorbing state, and no other transitions. The states are ordered descendingly by their total outgoing rates.

Both canonical representations require at most $2n$ parameters (if the distribution has no mass at $t = 0$, then $2n - 1$ parameters suffice) to characterize. However, the ordered bidiagonal representation has the most compact and straightforward structure with the least number of transitions. This compactness has many important consequences, especially in the field of fitting [BC92, BT94, ANO96]. In fitting a PH distribution to an empirical distribution, the performance of the algorithm depends on the form of the chosen PH representation and the interdependency among its estimated parameters.

The ordered bidiagonal and the Cox representations are dual representations of each other. This is especially important, because this means that when the need arises, their duals can be computed in linear time (viz. Theorem 3.5) rather than by employing Equation (2.16), which involves the inversion of a triangular matrix.

Throughout this thesis, we will mostly work with ordered bidiagonal representations, because the structure of the representations allows us to devise more efficient numerical algorithms. However, in some parts, we will use Cox representations. The fact that all Cox representations have exactly one initial state is important when we are developing a process calculus as we will do in Chapter 6.

3.2 Transformation to Ordered Bidiagonal

Theorem 3.2 does not provide a mechanism for transforming a given APH representation to its ordered bidiagonal representation. In this section, we continue using the running example to illustrate the main idea behind the transformation. This illustration does not only clarify the underlying mechanism of the algorithms later described in Section 3.3, but also provides a better insight into ordered bidiagonal representations.

Let $(\vec{\alpha}, \mathbf{A})$ be the APH representation depicted in Figure 3.1. The LST of the APH distribution is

$$\tilde{f}(s) = \frac{7s^2 + 42s + 54}{4(s+4)(s+3)(s+1)}.$$

The underlying CTMC of a PH distribution can be regarded as a transition system, in which each state is associated with some sojourn time, which is governed by some probability distribution. Thus, starting from a particular state in the transition system, the system evolves by making transitions from one state after some sojourn time to another state, and so on, until the absorbing state is reached.

In our example, the system may start in state 1 with probability $\frac{1}{2}$, and spend some time there (which is distributed according to an exponential distribution with rate 4), and then transition to state 2 with probability $\frac{1}{4}$. In state 2 the system sojourns for an exponentially distributed time with rate 3, and then transitions to state 3 with probability $\frac{1}{3}$. After an exponential delay with rate 1, the system transitions from state 3 to the absorbing state. Such a trajectory from an initial state to the absorbing state is called an *elementary series*.

Each of the elementary series is a hypoexponential representation. Recall that a hypoexponential distribution is the distribution of a convolution of several exponential distributions with possibly different rates. The elementary series described above occurs with probability $\frac{1}{2} \cdot \frac{1}{4} \cdot \frac{1}{3} = \frac{1}{24}$. Note that the sojourn times are determined by the total outgoing rates of the states in the representation.

Let the underlying CTMC of the APH representation be $\mathcal{M} = (\mathcal{S}, \mathbf{R}, \vec{\pi})$. We observe that each path $\sigma \in Paths(\mathcal{M})$ corresponds to an elementary series, and the probability with which the elementary series occurs corresponds to the occurrence probability $P(\sigma)$ of the path. Since the APH representation is finite and acyclic, the set $Paths(\mathcal{M})$ is finite, and each path $\sigma \in Paths(\mathcal{M})$ is also of finite length.

Figure 3.4 shows all elementary series of $(\vec{\alpha}, \mathbf{A})$. Each of the elementary series is depicted with its occurrence probability and LST. The representation $(\vec{\alpha}, \mathbf{A})$ is equivalent to the convex combination of all its elementary series, in which each series is weighted by its occurrence probability.

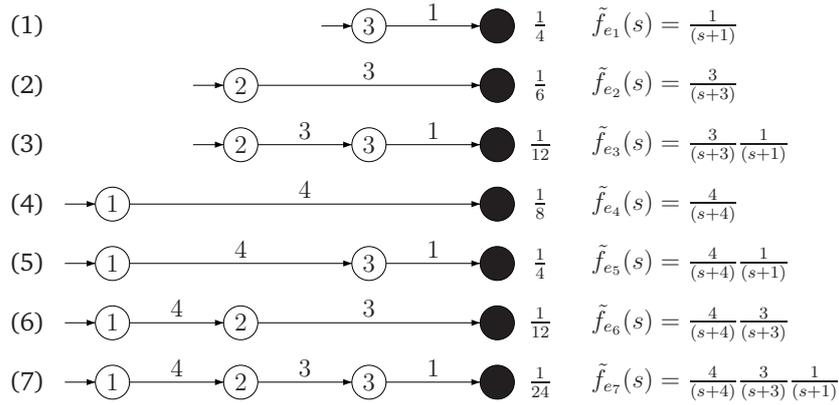


Figure 3.4: All Elementary Series of the APH Representation in Figure 3.1

From the LST of hypoexponential representations (and hence of the elementary series), it can be observed that exponential distributions forming the convolution can always be reordered. Here, we can define the set of the *basic series*. It is the set of hypoexponential representations with decreasing rates viewed from the absorbing state to each initial states. Figure 3.5 depicts all basic series of our example together with their LSTs.

Remark: We will provide formal definitions of the elementary and the basic series in Chapter 4.

Each elementary series is a convex combination of several basic series [Cum82]. In the example, this should be clear for elementary series (4), (6) and (7) (cf. Figure 3.4), but may not be obvious for the rest of them. In the following, we show that the elementary series (3) is a convex combination of the basic series (2) and (3) (cf. Figure 3.5). Consider the LSTs of elementary series (3), basic series (2) and (3), and the following holds

$$\frac{1}{s+1} \frac{3}{s+3} = \frac{1}{4} \frac{3}{s+3} \frac{4}{s+4} + \frac{3}{4} \frac{1}{s+1} \frac{3}{s+3} \frac{4}{s+4}.$$

We rely on the following identity to obtain the right hand side of the previous equation from the left hand side: Given $0 < \lambda \leq \mu$ and $p = \frac{\lambda}{\mu}$ then

$$\frac{\lambda}{s+\lambda} = p \frac{\mu}{s+\mu} + (1-p) \frac{\lambda}{s+\lambda} \frac{\mu}{s+\mu}. \quad (3.2)$$

In a similar fashion, the other elementary series can be expressed as the convex combinations of several basic series. Figure 3.6 shows the new expressions of all elementary series.

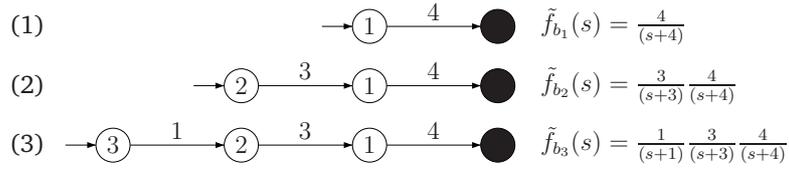


Figure 3.5: All Basic Series of the APH Representation in Figure 3.1

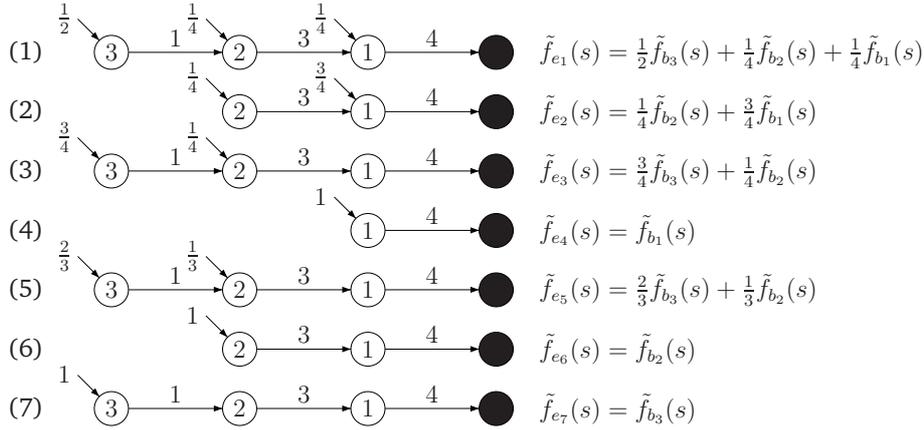


Figure 3.6: Elementary Series as Convex Combination of the Basic Series

Now, each of the new expressions of the elementary series is weighted by its occurrence probability to obtain the initial probabilities of all states forming the elementary series. The elementary series can then be combined to form a single ordered bidiagonal representation by summing up all the initial probabilities of each state of similar identity. As a result, an APH distribution with an ordered bidiagonal representation $(\vec{\beta}, \text{Bi}(1, 3, 4))$ is obtained, where $\vec{\beta} = [\frac{19}{48}, \frac{14}{48}, \frac{15}{48}]$. The ordered bidiagonal representation is depicted in Figure 3.2.

3.3 Transformation Algorithms

In this section, we present three algorithms for transforming APH representations to their ordered bidiagonal representations. The first algorithm formalizes the procedure described in the previous section. The second and the third algorithms rely on the concept of PH-majorization to perform the transformation. The algorithms are presented in chronological order, thus we should expect improvements from one to the next.

3.3.1 Cumani's Algorithm

The first algorithm is due to CUMANI [Cum82], and it works in the LST domain.

Let $(\vec{\alpha}, \mathbf{A})$ be an APH representation, and $\tilde{f}(s)$ be the LST of its distribution. Let $-\lambda_1, -\lambda_2, \dots, -\lambda_n$ be the eigenvalues of \mathbf{A} , and, without loss of generality, assume that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > 0$. (Note that the order of the eigenvalues here is different from that of Theorem 3.2 for the convenience of the presentation). Then the LST can

be written as

$$\tilde{f}(s) = \frac{P(s)}{Q(s)} = \sum_{i=1}^n \vec{\beta}_i \tilde{f}_i(s), \quad (3.3)$$

where $\tilde{f}_i(s)$ is the LST of the i -th basic series of $(\vec{\alpha}, \mathbf{A})$, namely

$$\tilde{f}_i(s) = \prod_{j=1}^i \frac{\lambda_j}{s + \lambda_j} = \left(\prod_{j=1}^i \frac{\lambda_j}{s + \lambda_j} \right) \cdot \left(\prod_{j=i+1}^n \frac{s + \lambda_j}{s + \lambda_j} \right) = \frac{P_i(s)}{Q(s)}.$$

By removing the polynomial $Q(s)$ from all $\tilde{f}_i(s)$, for $1 \leq i \leq n$ (cf. the last equation in (3.3)), we obtain

$$P(s) = \sum_{i=1}^n \vec{\beta}_i P_i(s).$$

This can be converted into a system of linear equations in the components of vector $\vec{\beta}$ by equating the coefficients of s^i , for $0 \leq i \leq n-1$, in both sides.

By solving the system of equations, the values of $\vec{\beta}_i$, for $1 \leq i \leq n$, can be obtained. The resulting canonical ordered bidiagonal representation is then given by $(\vec{\beta}, \mathbf{Bi}(\lambda_n, \lambda_{n-1}, \dots, \lambda_1))$, where $\vec{\beta} = [\vec{\beta}_n, \vec{\beta}_{n-1}, \dots, \vec{\beta}_1]$.

3.3.2 O'Kinneide's Algorithm

The second algorithm is due to O'KINNEIDE [O'C89], and it is based on the concept of PH-simplicity and PH-majorization.

We recall Theorem 2.41(2): if \mathbf{B} is PH-simple and \mathbf{B} PH-majorizes \mathbf{A} , then there is exactly one matrix \mathbf{P} with unit row-sums (i.e., $\mathbf{P}\vec{e} = \vec{e}$) such that $\mathbf{A}\mathbf{P} = \mathbf{P}\mathbf{B}$, and \mathbf{P} is of full rank if and only if \mathbf{A} is also PH-simple.

Now, let $(\vec{\alpha}, \mathbf{A})$ be an APH representation, and let \mathbf{A} be PH-simple. By Theorem 3.2, if $-\lambda_1, -\lambda_2, \dots, -\lambda_n$ are the eigenvalues of \mathbf{A} , and assuming that $\lambda_n \geq \lambda_{n-1} \geq \dots \geq \lambda_1 > 0$, then $\mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)$ PH-majorizes \mathbf{A} . Since \mathbf{A} is PH-simple, matrix \mathbf{P} is of full rank, and thus invertible. Therefore, $\mathbf{P}^{-1}\mathbf{A} = \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)\mathbf{P}^{-1}$. Letting $\mathbf{Q} = \mathbf{P}^{-1}$, and denoting the i -th row of matrix \mathbf{Q} by $\mathbf{Q}(i, *)$, we obtain the system of equations

$$\begin{aligned} \mathbf{Q}(i, *)\vec{e} &= 1, & \mathbf{Q}(i, *)\mathbf{A} &= -\lambda_i \mathbf{Q}(i, *) + \lambda_i \mathbf{Q}(i+1, *), & 1 \leq i \leq n, & \text{ or} \\ & & \mathbf{Q}(i, *)[\mathbf{A} + \lambda_i \mathbf{I}, \vec{e}] &= [\lambda_i \mathbf{Q}(i+1, *), 1], & 1 \leq i \leq n, & \end{aligned} \quad (3.4)$$

where $\mathbf{Q}(n+1, *)$ is defined to be the zero row vector, and \mathbf{I} is an identity matrix of appropriate dimension.

By solving the system of equations (3.4), matrix \mathbf{Q} is obtained, and then by inverting it, we obtain matrix \mathbf{P} . The canonical ordered bidiagonal representation is then given by $(\vec{\alpha}\mathbf{P}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$.

Note that the system of equations (3.4) has a unique solution only if matrices $[\mathbf{A} + \lambda_i \mathbf{I}, \vec{e}]$ are all of rank n , which is guaranteed if PH-generator \mathbf{A} is PH-simple. This requirement, however, restricts the practical use of this algorithm.

3.3.3 Spectral Polynomial Algorithm

The third algorithm is due to HE and ZHANG [HZ06b], and it is called the *spectral polynomial algorithm* (SPA). The algorithm does not make any assumption with respect to the PH-simplicity of the given PH-generator \mathbf{A} , and may result in an ordered bidiagonal representation of smaller size than the given APH representation.

Let $(\vec{\alpha}, \mathbf{A})$ be an APH representation, and let $-\lambda_1, -\lambda_2, \dots, -\lambda_n$ be the eigenvalues of \mathbf{A} . Let \mathbf{P} be a square matrix of the same dimension as that of \mathbf{A} , and denote the i -th column of matrix \mathbf{P} by $\mathbf{P}(*, i)$. Then the matrix equation $\mathbf{A}\mathbf{P} = \mathbf{P}\mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)$ can be expressed by

$$\mathbf{A}\mathbf{P}(*, i) = -\lambda_i\mathbf{P}(*, i) + \lambda_{i-1}\mathbf{P}(*, i-1), \quad 1 \leq i \leq n,$$

where we define $\lambda_0 = 1$ and $\mathbf{P}(*, 0) = \vec{0}$. This system of equations can be rewritten as

$$\begin{aligned} \mathbf{P}(*, i) &= \frac{1}{\lambda_i}(\mathbf{A} + \lambda_{i+1}\mathbf{I})\mathbf{P}(*, i+1), \quad 1 \leq i \leq n-1, \text{ or} \\ \mathbf{P}(*, i) &= \left(\prod_{j=i}^{n-1} \frac{1}{\lambda_j}(\mathbf{A} + \lambda_{j+1}\mathbf{I}) \right) \mathbf{P}(*, n), \quad 1 \leq i \leq n-1, \end{aligned} \quad (3.5)$$

and thus removing the need to define λ_0 and $\mathbf{P}(*, 0)$.

Matrices $\{(\mathbf{A} + \lambda_1\mathbf{I}) \cdots (\mathbf{A} + \lambda_i\mathbf{I}) \mid 2 \leq i \leq n\}$ are called the spectral polynomials of \mathbf{A} [HZ06b], hence the name of the algorithm. From the Cayley-Hamilton theorem [Mey04], we have

$$p(\mathbf{A}) = (\mathbf{A} + \lambda_1\mathbf{I})(\mathbf{A} + \lambda_2\mathbf{I}) \cdots (\mathbf{A} + \lambda_n\mathbf{I}) = 0, \quad (3.6)$$

where $p(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I})$ is the characteristic polynomial of matrix \mathbf{A} .

For the matrix \mathbf{P} to be of unit row-sums (i.e., $\mathbf{P}\vec{e} = \vec{e}$), we need to choose $\mathbf{P}(*, n)$ appropriately. Equation (3.6) can be evaluated as follows

$$\begin{aligned} &(\mathbf{A} + \lambda_1\mathbf{I})(\mathbf{A} + \lambda_2\mathbf{I}) \cdots (\mathbf{A} + \lambda_n\mathbf{I}) = 0, \\ &\mathbf{A}^n + \mathbf{A}^{n-1}\lambda_n\mathbf{I} + \mathbf{A}^{n-1}\lambda_{n-1}\mathbf{I} + \cdots + \left(\prod_{i=1}^{n-1} \lambda_i \right) \mathbf{A} + \left(\prod_{i=1}^n \lambda_i \right) \mathbf{I} = 0, \\ &\sum_{i=2}^n \left(\prod_{j=1}^{i-2} \lambda_j \right) \left(\prod_{j=i}^n (\mathbf{A} + \lambda_j\mathbf{I}) \right) \mathbf{A} + \mathbf{A} \prod_{i=1}^{n-1} \lambda_i + \mathbf{I} \prod_{i=1}^n \lambda_i = 0. \end{aligned} \quad (3.7)$$

Multiplying both sides of Equation (3.7) by $\vec{e}/(\lambda_1\lambda_2 \cdots \lambda_n)$, we obtain

$$\frac{\mathbf{P}(*, 1)\mathbf{A}\vec{e}}{\mathbf{P}(*, n)\lambda_n} + \frac{\mathbf{P}(*, 2)\mathbf{A}\vec{e}}{\mathbf{P}(*, n)\lambda_n} + \cdots + \frac{\mathbf{P}(*, n-1)\mathbf{A}\vec{e}}{\mathbf{P}(*, n)\lambda_n} + \frac{\mathbf{A}\vec{e}}{\lambda_n} + \vec{e} = 0.$$

Now, if we choose $\mathbf{P}(*, n) = -\frac{\mathbf{A}\vec{e}}{\lambda_n}$, then we obtain

$$-\mathbf{P}(*, 1) - \mathbf{P}(*, 2) - \cdots - \mathbf{P}(*, n-1) - \mathbf{P}(*, n) + \vec{e} = 0,$$

thus ensuring $\mathbf{P}\vec{e} = \vec{e}$.

If there exists a sub-stochastic vector $\vec{\beta}$ such that $\vec{\beta} = \vec{\alpha}\mathbf{P}$, then we have $\text{PH}(\vec{\alpha}, \mathbf{A}) = \text{PH}(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$. When the bidiagonal PH-generator is in canonical form,

namely if $\lambda_n \geq \lambda_{n-1} \geq \dots \geq \lambda_1$, then such a sub-stochastic vector always exists because $\mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)$ PH-majorizes \mathbf{A} .

The spectral polynomial algorithm uses $n - 1$ vector-matrix multiplications of vectors and matrices of dimension n , where n is the size of the given APH representation. Therefore its complexity is $\mathcal{O}(n^3)$.

3.4 Reducing the Representations

In this section, we propose an algorithm to reduce the size of APH representations. The algorithm is roughly as follows: Firstly, given an APH distribution with representation $(\vec{\alpha}, \mathbf{A})$, the representation is transformed into an ordered bidiagonal representation $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$ by using the spectral polynomial algorithm. The new representation is at most equal to the size of the original one. Secondly, once an ordered bidiagonal representation is obtained, some of its states can possibly be removed without affecting its distribution function. In the rest of the section, we will show that a smaller representation can be obtained by removing some “unnecessary” states from the ordered bidiagonal representation. A procedure for identifying and removing those unnecessary states will be provided. The resulting representation is an ordered bidiagonal representation having less states.

3.4.1 The L -terms

Recall that the LST of an exponential distribution with rate λ is $\tilde{f}(s) = \frac{\lambda}{s+\lambda}$. Let $L(\lambda) = \frac{s+\lambda}{\lambda}$, i.e., the reciprocal of the LST of the exponential distribution. We call a single expression of $L(\cdot)$ an L -term. The LST of an ordered bidiagonal representation $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$ can be written as

$$\begin{aligned} \tilde{f}(s) &= \frac{\vec{\beta}_1}{L(\lambda_1) \cdots L(\lambda_n)} + \frac{\vec{\beta}_2}{L(\lambda_2) \cdots L(\lambda_n)} + \cdots + \frac{\vec{\beta}_n}{L(\lambda_n)}, \\ &= \frac{\vec{\beta}_1 + \vec{\beta}_2 L(\lambda_1) + \cdots + \vec{\beta}_n L(\lambda_1) L(\lambda_2) \cdots L(\lambda_{n-1})}{L(\lambda_1) L(\lambda_2) \cdots L(\lambda_n)}. \end{aligned} \quad (3.8)$$

Note that the LST expression in Equation (3.8) may not be in irreducible ratio form. The LST is produced in such a way that the denominator polynomial corresponds exactly to the sequence of the total outgoing rates of the ordered bidiagonal representation. Hence, the degree of the denominator polynomial is equal to the size of the ordered bidiagonal representation.

Example 3.7 (Laplace-Stieltjes Transform in L -terms). *Figure 3.7 depicts an ordered bidiagonal representation. The LST of the representation, expressed in its L -terms, is*

$$\tilde{f}(s) = \frac{\frac{1}{5} + \frac{1}{5}L(1) + \frac{2}{5}L(1)L(2) + \frac{1}{5}L(1)L(2)L(2)}{L(1)L(2)L(2)L(3)L(4)L(5)L(5)}. \quad (3.9)$$

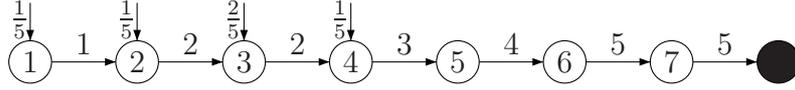


Figure 3.7: An Ordered Bidiagonal Representation

3.4.2 Reduction

Observing Equation (3.8), we see that in order to remove a state from the ordered bidiagonal representation, we have to find a common L -term in both the numerator and denominator polynomials in Equation (3.8). Removing such a common L -term from the numerator and denominator polynomials means removing the corresponding state from the representation.

However, such a removal of a state can only be carried out if after the removal, the initial probability distribution $\vec{\beta}$ is redistributed in a correct way, namely the new initial probability distribution $\vec{\delta}$ is a sub-stochastic vector. The procedure for identifying and properly removing a state from an ordered bidiagonal representation is described formally in the following lemmas.

Lemma 3.8. *Let $\vec{\alpha}_1 \neq 0$. If $\vec{\alpha}_1 + \vec{\alpha}_2 L(\lambda_1) + \cdots + \vec{\alpha}_i L(\lambda_1) \cdots L(\lambda_{i-1})$ is divisible by $L(\lambda)$, then there are $\vec{\alpha}'_j$, for $1 \leq j \leq i - 1$, and $\sum_{j=1}^{i-1} \vec{\alpha}'_j = \sum_{j=1}^i \vec{\alpha}_j$ such that*

$$\begin{aligned} L(\lambda)[\vec{\alpha}'_1 + \vec{\alpha}'_2 L(\lambda_1) + \cdots + \vec{\alpha}'_{i-1} L(\lambda_1) \cdots L(\lambda_{i-2})] \\ = \vec{\alpha}_1 + \vec{\alpha}_2 L(\lambda_1) + \cdots + \vec{\alpha}_i L(\lambda_1) \cdots L(\lambda_{i-1}). \end{aligned} \quad (3.10)$$

Proof. Evaluating the left hand side of Equation (3.10) shows that the coefficient of s^{i-1} is an expression of $\vec{\alpha}'_{i-1}$, the coefficient of s^{i-2} is an expression of $\vec{\alpha}'_{i-2}$ and $\vec{\alpha}'_{i-1}$, and \dots , and the coefficient of s is an expression of $\vec{\alpha}'_j$, for all $1 \leq j \leq i - 1$. Therefore, we can create a system of $i - 1$ linear equations in $i - 1$ variables $\vec{\alpha}'_j$ from these expressions, such that

$$\mathbf{A}[\vec{\alpha}'_1, \vec{\alpha}'_2, \dots, \vec{\alpha}'_{i-1}]^\top = \vec{b},$$

where \vec{b} is a vector obtained by evaluating the right hand side of Equation (3.10), and collecting the coefficients of s, s^2, \dots, s^{i-1} . This vector is nonzero, since at least $\vec{\alpha}_1 \neq 0$.

Matrix \mathbf{A} is clearly triangular. Moreover, all its diagonal components are nonzero; they are $\frac{1}{\lambda}, \frac{1}{\lambda_1 \lambda}, \frac{1}{\lambda_1 \lambda_2 \lambda}, \dots, \frac{1}{\lambda_1 \cdots \lambda_{i-2} \lambda}$. Therefore, matrix \mathbf{A} is nonsingular, and the system of equations has a unique solution [Mey04].

By evaluating the left and the right hand sides of Equation (3.10), and then collecting the coefficients of s^0 on both sides, we can conclude that $\sum_{j=1}^{i-1} \vec{\alpha}'_j = \sum_{j=1}^i \vec{\alpha}_j$. \square

Lemma 3.9. *Let $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$ be an ordered bidiagonal representation of size n . If for some $1 \leq i \leq n$,*

$$\vec{\beta}_1 + \vec{\beta}_2 L(\lambda_1) + \cdots + \vec{\beta}_i L(\lambda_1) \cdots L(\lambda_{i-1})$$

is divisible by $L(\lambda_i)$ then there exists a vector $\vec{\delta}$ such that

$$\text{PH}(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)) = \text{ME}(\vec{\delta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots, \lambda_n), \vec{e}).$$

If $\vec{\delta}$ is a sub-stochastic vector, then $(\vec{\delta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots, \lambda_n))$ is an ordered bidiagonal representation of size $n - 1$ and

$$\text{PH}(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)) = \text{PH}(\vec{\delta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots, \lambda_n)).$$

Proof. Assume that for some $1 \leq i \leq n$, $\vec{\beta}_1 + \vec{\beta}_2 L(\lambda_1) + \cdots + \vec{\beta}_i L(\lambda_1) \cdots L(\lambda_{i-1})$ is divisible by $L(\lambda_i)$. Then

$$p'(s)L(\lambda_i) = \vec{\beta}_1 + \vec{\beta}_2 L(\lambda_1) + \cdots + \vec{\beta}_i L(\lambda_1) \cdots L(\lambda_{i-1})$$

for some polynomial $p'(s)$ of degree $i - 2$.

Now, the LST of $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$ can be expressed in its L -terms by $\frac{p(s)}{q(s)}$, where

$$\begin{aligned} p(s) &= p'(s)L(\lambda_i) + \vec{\beta}_{i+1}L(\lambda_1) \cdots L(\lambda_{i-1})L(\lambda_i) + \vec{\beta}_{i+2}L(\lambda_1) \cdots L(\lambda_{i-1})L(\lambda_i)L(\lambda_{i+1}) \\ &\quad + \cdots + \vec{\beta}_n L(\lambda_1) \cdots L(\lambda_{i-1})L(\lambda_i)L(\lambda_{i+1}) \cdots L(\lambda_{n-1}), \\ &= L(\lambda_i)[p'(s) + \vec{\beta}_{i+1}L(\lambda_1) \cdots L(\lambda_{i-1}) + \vec{\beta}_{i+2}L(\lambda_1) \cdots L(\lambda_{i-1})L(\lambda_{i+1}) \\ &\quad + \cdots + \vec{\beta}_n L(\lambda_1) \cdots L(\lambda_{i-1})L(\lambda_{i+1}) \cdots L(\lambda_{n-1})], \text{ and} \\ q(s) &= L(\lambda_1) \cdots L(\lambda_{i-1})L(\lambda_i)L(\lambda_{i+1}) \cdots L(\lambda_n), \\ &= L(\lambda_i)[L(\lambda_1) \cdots L(\lambda_{i-1})L(\lambda_{i+1}) \cdots L(\lambda_n)]. \end{aligned}$$

Removing the common L -term $L(\lambda_i)$, we obtain $\frac{p(s)}{q(s)} = \frac{p''(s)}{q''(s)}$, where

$$\begin{aligned} p''(s) &= p'(s) + \vec{\beta}_{i+1}L(\lambda_1) \cdots L(\lambda_{i-1}) + \vec{\beta}_{i+2}L(\lambda_1) \cdots L(\lambda_{i-1})L(\lambda_{i+1}) \\ &\quad + \cdots + \vec{\beta}_n L(\lambda_1) \cdots L(\lambda_{i-1})L(\lambda_{i+1}) \cdots L(\lambda_{n-1}), \text{ and} \\ q''(s) &= L(\lambda_1) \cdots L(\lambda_{i-1})L(\lambda_{i+1}) \cdots L(\lambda_n). \end{aligned}$$

By Lemma 3.8, $p'(s)$ can be expressed as $\vec{\beta}'_1 + \vec{\beta}'_2 L(\lambda_1) + \cdots + \vec{\beta}'_{i-1} L(\lambda_1) \cdots L(\lambda_{i-2})$, for some $\vec{\beta}'_j$, for $1 \leq j \leq i - 1$, and $\sum_{j=1}^{i-1} \vec{\beta}'_j = \sum_{j=1}^i \vec{\beta}_j$. Let $\vec{\delta} = [\vec{\beta}'_1, \dots, \vec{\beta}'_{i-1}, \vec{\beta}_{i+1}, \dots, \vec{\beta}_n]$. Since $\vec{\beta}\vec{e} = 1$, also $\vec{\delta}\vec{e} = 1$, and by inspecting the structure of $\frac{p''(s)}{q''(s)}$, we can conclude that it is an LST of an ME distribution $\text{ME}(\vec{\delta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots, \lambda_n), \vec{e})$, whose representation is depicted in Figure 3.8. Since this ME distribution and PH distribution $\text{PH}(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$ arise from the same LST, they are both one and the same probability distribution.

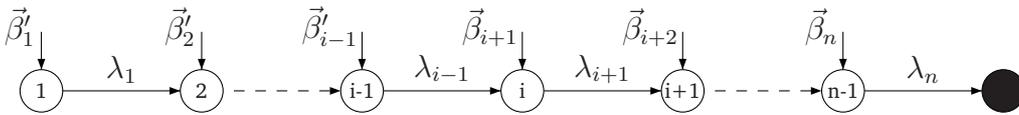


Figure 3.8: Ordered Bidiagonal Representation of $\frac{p''(s)}{q''(s)}$

From the parameterization definition of PH distributions (cf. Section 2.5), if $\vec{\delta}_i \geq 0$, for all $1 \leq i \leq n - 1$, then the representation in Figure 3.8 is an ordered bidiagonal representation of size $n - 1$ representing $\text{PH}(\vec{\delta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots, \lambda_n))$, which then agrees with $\text{PH}(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$. \square

A simple observation regarding the divisibility of the numerator polynomial by L -terms is as follows. By inspecting the form of Equation (3.8), we can observe that $L(\lambda_1)$ is not a divisor of the numerator polynomial if $\vec{\beta}_1 \neq 0$. Therefore, the first state of any ordered bidiagonal representation cannot be removed. The same case applies to all other states in the representation whose total outgoing rate is equal to

that of the first state. Hence, Erlang distributions and convex combinations of Erlang distributions with the same rate are not reducible in this way, because all states in Erlang distributions have the same total outgoing rate, λ_1 .

To reduce the size of an ordered bidiagonal representation, we need to check two conditions in Lemma 3.9, namely the divisibility of the numerator polynomial and the sub-stochasticity of the resulting initial probability vector. For this, let polynomial

$$R(s) = \vec{\beta}_1 + \vec{\beta}_2 L(\lambda_1) + \cdots + \vec{\beta}_i L(\lambda_1) \cdots L(\lambda_{i-1}). \quad (3.11)$$

The root of an L-term $L(\lambda_i)$ is $-\lambda_i$. Hence, polynomial $R(s)$ is divisible by $L(\lambda_i)$ if one of the roots of $R(s)$ is also $-\lambda_i$, or simply if $R(-\lambda_i) = 0$.

Example 3.10 (Common L-terms). Consider the LST in Example 3.7, and recall that $L(\lambda) = \frac{s+\lambda}{\lambda}$. We can determine that

$$R(s) = \frac{1}{5} + \frac{1}{5}L(1) + \frac{2}{5}L(1)L(2) + \frac{1}{5}L(1)L(2)L(2),$$

is divisible by $L(5)$ by evaluating $R(-5)$ as follows

$$R(-5) = \frac{1}{5} + \frac{1}{5}(-4) + \frac{2}{5}(-4)\left(\frac{-3}{2}\right) + \frac{1}{5}(-4)\left(\frac{-3}{2}\right)\left(\frac{-3}{2}\right) = 0.$$

The sub-stochasticity of the resulting initial probability vector, on the other hand, can be checked while computing it. We can compute it by following the procedure described in the proof of Lemma 3.8, namely by building and then solving the system of equations. However, to take advantage of the special structure of ordered bidiagonal representations, we proceed differently.

We refer to Lemma 3.9. Let $\mathbf{Bi}_1 := \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_i)$, $\mathbf{Bi}_2 := \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_{i-1})$ and $\vec{e}|_x$ be a column vector of dimension x whose components are all equal to 1. The following lemma states that we can simply ignore the last $n - i$ states in both ordered bidiagonal chains.

Lemma 3.11. If $\vec{\delta}_j = \vec{\beta}_{j+1}$, for $i \leq j \leq n - 1$, then

$$\text{PH}(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)) = \text{ME}(\vec{\delta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots, \lambda_n), \vec{e}|_{n-1})$$

implies

$$\text{PH}([\vec{\beta}_1, \vec{\beta}_2, \dots, \vec{\beta}_i], \mathbf{Bi}_1) = \text{ME}([\vec{\delta}_1, \vec{\delta}_2, \dots, \vec{\delta}_{i-1}], \mathbf{Bi}_2, \vec{e}|_{i-1}). \quad (3.12)$$

Proof. Both $\text{PH}(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$ and $\text{ME}(\vec{\delta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots, \lambda_n), \vec{e}|_{n-1})$ have ordered bidiagonal representations. Furthermore, each of the last $n - i$ states in one representation is identical to the corresponding state in the other representation, namely both states have the same initial probability and the same total outgoing rate. The representations of $\text{PH}([\vec{\beta}_1, \vec{\beta}_2, \dots, \vec{\beta}_i], \mathbf{Bi}_1)$ and $\text{ME}([\vec{\delta}_1, \vec{\delta}_2, \dots, \vec{\delta}_{i-1}], \mathbf{Bi}_2, \vec{e}|_{i-1})$ can be obtained by collapsing the $n - i$ states in each of the ordered bidiagonal representations to its absorbing state, and assigning to the absorbing state an initial probability of $1 - \sum_{j=1}^i \beta_j$. \square

From Equation (3.12), we obtain

$$\begin{aligned} 1 - [\vec{\beta}_1, \vec{\beta}_2, \dots, \vec{\beta}_i] e^{\mathbf{Bi}_1 t} \vec{e}|_i &= 1 - [\vec{\delta}_1, \vec{\delta}_2, \dots, \vec{\delta}_{i-1}] e^{\mathbf{Bi}_2 t} \vec{e}|_{i-1}, \\ [\vec{\beta}_1, \vec{\beta}_2, \dots, \vec{\beta}_i] e^{\mathbf{Bi}_1 t} \vec{e}|_i &= [\vec{\delta}_1, \vec{\delta}_2, \dots, \vec{\delta}_{i-1}] e^{\mathbf{Bi}_2 t} \vec{e}|_{i-1}. \end{aligned} \quad (3.13)$$

Hence, to compute vector $[\vec{\delta}_1, \vec{\delta}_2, \dots, \vec{\delta}_{i-1}]$ from vector $[\vec{\beta}_1, \vec{\beta}_2, \dots, \vec{\beta}_i]$, $i - 1$ equations relating their components are required. Equation (3.13) can be evaluated at $i - 1$ different t values to obtain such a system of equations. However, such function evaluations in practice can be costly, because they involve matrix exponentiations. To avoid this, we proceed in a different fashion.

For a PH representation $(\vec{\alpha}, \mathbf{A})$, the j -th derivative, for $j \in \mathbb{Z}_{\geq 0}$, of its distribution function is given by

$$\frac{d^j}{dt^j} F(t) = -\vec{\alpha} \mathbf{A}^j e^{\mathbf{A}t} \vec{e}.$$

Evaluating these derivatives at $t = 0$ allows us to avoid computing the exponential of matrices. Hence, the components of vector $[\vec{\delta}_1, \vec{\delta}_2, \dots, \vec{\delta}_{i-1}]$ can be computed by solving the following system of equations

$$\begin{aligned} [\vec{\delta}_1, \vec{\delta}_2, \dots, \vec{\delta}_{i-1}] \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_{i-1})^j \vec{e}_{i-1} \\ = [\vec{\beta}_1, \vec{\beta}_2, \dots, \vec{\beta}_i] \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_i)^j \vec{e}_i, \quad 0 \leq j \leq i - 2. \end{aligned} \quad (3.14)$$

Once the system of equations (3.14) is solved, the sub-stochasticity of $[\vec{\delta}_1, \vec{\delta}_2, \dots, \vec{\delta}_{i-1}]$ can be determined simply by verifying that all of its components are nonnegative real numbers.

Example 3.12 (Redistributing the Initial Probabilities). *We continue using Example 3.10: a state that corresponds to $L(5)$ can be removed from the representation. Let $(\vec{\alpha}, \mathbf{A})$ and $(\vec{\gamma}, \mathbf{G})$ be the original and the resulting representations, respectively, where $\vec{\alpha} = [\frac{1}{5}, \frac{1}{5}, \frac{2}{5}, \frac{1}{5}, 0, 0, 0]$. Vector $\vec{\gamma}$ can be obtained by solving the following system of equations*

$$\begin{aligned} \vec{\gamma}_1 + \vec{\gamma}_2 + \vec{\gamma}_3 + \vec{\gamma}_4 + \vec{\gamma}_5 + \vec{\gamma}_6 &= 1, \\ -5\vec{\gamma}_6 &= 0, \\ -20\vec{\gamma}_5 + 25\vec{\gamma}_6 &= 0, \\ -60\vec{\gamma}_4 + 180\vec{\gamma}_5 - 125\vec{\gamma}_6 &= 0, \\ -120\vec{\gamma}_3 + 720\vec{\gamma}_4 - 1220\vec{\gamma}_5 + 625\vec{\gamma}_6 &= -60, \\ -240\vec{\gamma}_2 + 1680\vec{\gamma}_3 - 5820\vec{\gamma}_4 + 7380\vec{\gamma}_5 - 3125\vec{\gamma}_6 &= 780. \end{aligned}$$

The new initial probability vector $\vec{\gamma} = [\frac{1}{4}, \frac{1}{4}, \frac{1}{2}, 0, 0, 0]$.

3.4.3 Algorithm

A state that is associated with the L -term $L(\lambda_i)$ in Lemma 3.9 will be called a *removable* state. Formally, a state in an ordered bidiagonal representation is removable if and only if the numerator polynomial of the LST of the representation is divisible by the L -term associated with the state, and if removing the state results in a valid initial probability distribution.

Lemma 3.9 can be turned into an algorithm that reduces the size of a given APH representation.

Algorithm 3.13 (Reducing Acyclic Representation).

- 1: **function** REDACYCREP($\vec{\alpha}, \mathbf{A}$)
- 2: $(\vec{\beta}, \mathbf{Bi}) \leftarrow \text{SPA}(\vec{\alpha}, \mathbf{A})$

```

3:    $n \leftarrow \text{SIZEOF}(\vec{\beta}, \mathbf{Bi})$ 
4:    $i \leftarrow 2$ 
5:   while  $i \leq n$  do
6:     if  $\text{REMOVABLE}(i, (\vec{\beta}, \mathbf{Bi}))$  then
7:        $(\vec{\beta}', \mathbf{Bi}') \leftarrow \text{REMOVE}(i, (\vec{\beta}, \mathbf{Bi}))$ 
8:        $(\vec{\beta}, \mathbf{Bi}) \leftarrow (\vec{\beta}', \mathbf{Bi}')$ 
9:        $n \leftarrow n - 1$ 
10:    else
11:       $i \leftarrow i + 1$ 
12:    end if
13:  end while
14:  return  $(\vec{\beta}, \mathbf{Bi})$ 
15: end function

```

Algorithm 3.13 takes as input an APH representation, and outputs its reduced ordered bidiagonal representation. Function $\text{SPA}(\cdot)$, in the algorithm, transforms an APH representation to its ordered bidiagonal representation using the spectral polynomial algorithm [HZ06b]. Function $\text{SIZEOF}(\cdot)$ returns the size of the given representation. Function $\text{REMOVABLE}(\cdot)$ returns TRUE if state s_i is removable from the given representation, *i.e.*, $R(s) = 0$ in Equation (3.11), and the solution of the system of equations (3.14) has no negative components. Removing the state means redistributing the initial probability distribution, whose computation was shown in Equation (3.14). This is what function $\text{REMOVE}(\cdot)$ does, namely removing state s_i from the given representation.

The algorithm proceeds by checking for each state whether it is removable or not. If the state is removable then it is eliminated. Furthermore, the algorithm terminates once all states have been checked, and the removable ones have been eliminated. Hence, the algorithm does what it is supposed to do and terminates.

Let n be the size of the given APH representation. The spectral polynomial algorithm entails n matrix-vector multiplications of matrices and vectors, each having dimension n ; thus is of time complexity $\mathcal{O}(n^3)$. Checking whether state s_i is removable needs an evaluation of Equation (3.11). If Equation (3.11) is rewritten as

$$R(s) = \vec{\beta}_1 + L(\lambda_1) \left(\vec{\beta}_2 + L(\lambda_2) \left(\cdots \left(\vec{\beta}_{i-1} + L(\lambda_{i-1}) \vec{\beta}_i \right) \cdots \right) \right),$$

then the evaluation can be completed with $2i - 2$ multiplications and $2i - 2$ additions. In the worst case, this has time complexity $\mathcal{O}(n)$.

If state s_i is removable, eliminating it entails solving Equation (3.14). However, we observe that for any bidiagonal PH-generator \mathbf{Bi} of dimension n , $\mathbf{Bi}(j, j) = -\mathbf{Bi}(j, j+1)$ for $1 \leq j < n$. Now, since both PH-generators in the system of equations are bidiagonal, this observation allows us to prove the following lemma.

Lemma 3.14. *The system of equations (3.14) can be transformed into a system of equations given by*

$$\mathbf{A}[\vec{\delta}_1, \vec{\delta}_2, \dots, \vec{\delta}_{i-1}]^\top = \vec{b}, \quad (3.15)$$

where \mathbf{A} is an upper triangular matrix of dimension $i - 1$. Furthermore, this transformation requires $\mathcal{O}(i^2)$ multiplications and $\mathcal{O}(i^2)$ additions.

The proof of this lemma is available in Appendix B.1.

Since the transformation requires $\mathcal{O}(i^2)$ multiplications and $\mathcal{O}(i^2)$ additions, in the worst case, the transformation is of time complexity $\mathcal{O}(n^2)$. On the other hand, solving Equation (3.15) to obtain $\vec{\delta}_j$, for $1 \leq j \leq i - 1$, has time complexity $\mathcal{O}(n^2)$ in the worst case. This is because \mathbf{A} is an upper triangular matrix, which means we only need to apply backward substitutions.

Since the procedure for checking whether a state is removable and then removing it is carried out at most n times, the overall time complexity of the algorithm is $\mathcal{O}(n^3 + n(n + n^2 + n^2)) = \mathcal{O}(n^3)$.

3.5 Examples: Fault-Tolerant System

Fault trees [HK92] are used to model fault-tolerant systems and to analyze their reliability. The fault tree in Figure 3.9 models a fault-tolerant system, which consists of two identical components. The redundancy of the components is introduced to increase the system's reliability. The system fails if both components fail. Furthermore, each component is comprised of three sub-components, whose failure times are governed by exponential distributions with rate 1, 2 and 3, respectively. Each of the components fails if at least two of its sub-components fail.

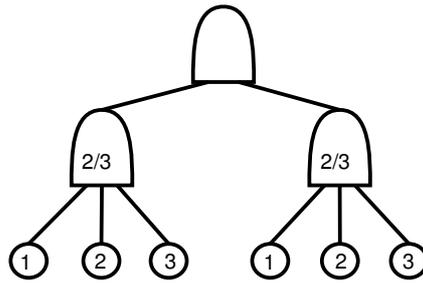


Figure 3.9: A Fault Tree Model of a Fault-Tolerant System

The fault tree gives rise to an APH distribution whose representation is depicted in Figure 3.10. The representation is obtained by applying a formalization of the fault tree semantics with exponential distributions as the basic events [DBB93]. We will discuss more about this type of fault trees in Chapter 7. This representation is the result of a previous lumping algorithm that is based on weak-bisimulation equivalence. This representation, however, is not minimal. In the following we will show that the APH distribution has a smaller ordered bidiagonal representation.

This subsection shows the usability of the proposed reduction algorithm on this APH representation. Inputting the representation into Algorithm 3.13, it is first transformed into an ordered bidiagonal representation. The ordered bidiagonal representation has the same number of states as the original representation. It is depicted in Figure 3.11.

In the following, we describe the elimination of state 4 from the obtained ordered bidiagonal representation. Let us consider the LST of this ordered bidiagonal representation expressed in L -terms. First, according to the observation right after Lemma 3.9, state 1 cannot be eliminated, because the numerator polynomial of the LST is not divisible by $L(3)$. Next, state 2 is not removable because $\frac{7}{5280} + \frac{599}{142560}L(3) = \frac{788}{142560}L(\frac{2364}{599})$ is not divisible by $L(4)$. State 3 also, since $\frac{7}{5280} + \frac{599}{142560}L(3) + \frac{1267}{142560}L(3)L(4)$ is not

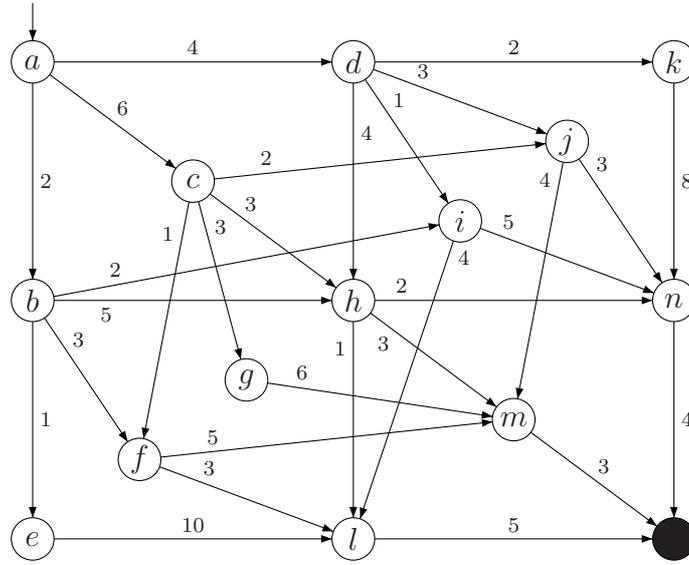


Figure 3.10: The PH Representation of the Fault Tree in Figure 3.9

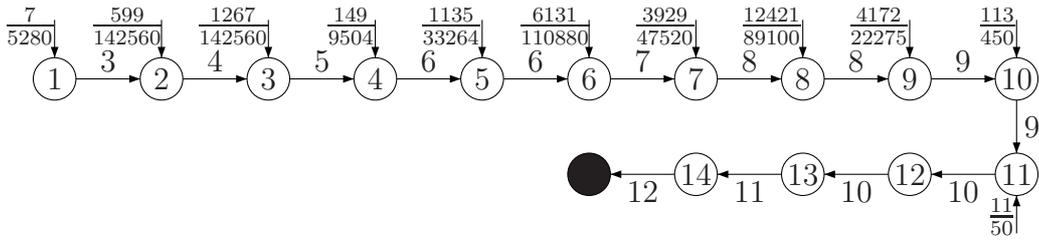


Figure 3.11: The Ordered Bidiagonal Representation of Figure 3.10

divisible by $L(5)$. For state 4, on the other hand, $\frac{7}{5280} + \frac{599}{142560}L(3) + \frac{1267}{142560}L(3)L(4) + \frac{149}{9504}L(3)L(4)L(5)$, is divisible by $L(6)$.

Now we have to compute the new initial probability distribution. Note, however, that the initial probabilities of states 1, 2, 3 and 4 are redistributed only to states 1, 2 and 3 after the removal of state 4. This means that instead of using the whole matrix, we need only to compute the new initial probabilities of states 1, 2 and 3 given the current values of the initial probabilities of states 1, 2, 3 and 4. The system of equations to be solved is

$$\left[\frac{7}{5280}, \frac{599}{142560}, \frac{1267}{142560}, \frac{149}{9504} \right] \mathbf{Bi}(3, 4, 5, 6)^j \vec{e}_4 = [\vec{\gamma}_1, \vec{\gamma}_2, \vec{\gamma}_3] \mathbf{Bi}(3, 4, 5)^j \vec{e}_3, \quad 0 \leq j \leq 2.$$

The solution is $\vec{\gamma} = \left[\frac{7}{2640}, \frac{41}{4752}, \frac{149}{7920} \right]$.

The ordered bidiagonal representation after all its removable states are eliminated is shown in Figure 3.12. It is a minimal representation, because the size of the representation is equal to its algebraic degree (cf. Lemma 2.14).

3.6 Relation to Lumping

One may wonder how likely it is to encounter a model that is reducible by our algorithm, or by lumping. For any pre-specified structured PH representation of size n ,

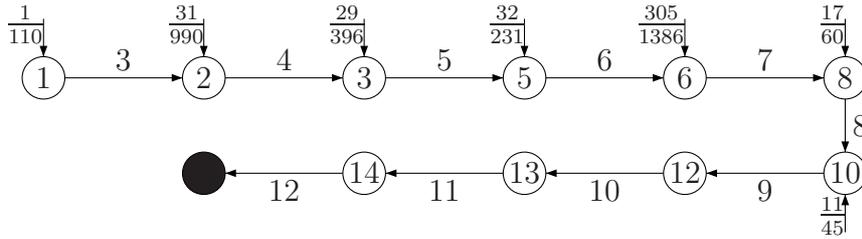


Figure 3.12: A Minimal Representation of Figure 3.11

the set of parameter values producing PH distributions of algebraic degree less than n has measure zero [CM02]. In other words, the chance of finding a reducible Markov chain appears very low. This is indeed true for models obtained from most parameter estimation mechanisms such as fitting methods.

However, this measure based interpretation is often misleading. Many cases have been reported that show the (sometimes tremendous) reduction obtained by lumping a model. The question thus is: why are aggregation techniques for Markov chains such as lumping successful after all? They are indeed successful for representations that are constructed out of a structured behavioral representation. The reason lies in the way models are constructed, which usually results in very specific and particular subsets of parameter values.

The constructive mechanisms used in building Markovian models from smaller components usually involve concatenation, choice, and composition operations, which correspond to stochastic operations: convolution, minimum (or convex combination), and maximum, respectively. These operations often produce models with repeatable, symmetric, and similar sub-structures, which enable reduction by the aggregation techniques. In the context of APH representations, for instance, the idea of the *core series*, which we will discuss in Chapter 4, identifies these sub-structures in the set of paths to the absorbing state.

To shed some light on the relation of our reduction algorithm to lumping, we here consider weak bisimilarity. The notion of weak bisimilarity is formalized in the following definition, which is a variation of the original definition [Bra02, BKHW05], accounting for the case of absorbing states (and otherwise treating the chain as unlabelled).

Let $\mathcal{M} = (\mathcal{S}, \mathbf{R}, \vec{\pi})$ be the underlying CTMC of a PH representation. For $C \subseteq \mathcal{S}$, let $\mathbf{R}(s, C) = \sum_{s' \in C} \mathbf{R}(s, s')$. If \mathcal{R} is an equivalence relation on \mathcal{S} , then \mathcal{S}/\mathcal{R} is the partitioning of \mathcal{S} induced by \mathcal{R} , and, for $s \in \mathcal{S}$, $[s]_{\mathcal{R}}$ is the partition (class) that contains s . Throughout this section we assume that a CTMC always has a single initial state. Therefore, an initial probability vector $\vec{\pi}$ is of the form of a unit vector, whose component that is equal to 1 corresponds to the starting state of the CTMC model.

Definition 3.15 (Weak Bisimulation). For a CTMC $\mathcal{M} = (\mathcal{S}, \mathbf{R}, \vec{\pi})$, let \mathcal{R} be an equivalence relation on \mathcal{S} . \mathcal{R} is a weak bisimulation on \mathcal{M} if for all $(s_1, s_2) \in \mathcal{R}$: $\mathbf{R}(s_1, C) = \mathbf{R}(s_2, C)$ for all $C \in \mathcal{S}/\mathcal{R}$ with $C \neq [s_1]_{\mathcal{R}}$, and if absorbing states are only related to absorbing states. States s_1 and s_2 are weakly bisimilar, denoted $s_1 \approx s_2$, if and only if there exists a weak bisimulation \mathcal{R} on \mathcal{M} such that $(s_1, s_2) \in \mathcal{R}$.

Weak bisimulation differs from the more prominent notion of strong bisimulation (or ordinary lumping [Buc94]) in that transitions that do not cross class boundaries are

not considered. This can be seen as the “class generalization” of the fact that in a CTMC the values of $\mathbf{R}(s, s)$ (i.e., loops at states) are irrelevant for the diagonal components of the corresponding generator matrix \mathbf{Q} .

States that are weakly bisimilar in a CTMC model can be lumped by moving to the quotient induced by this equivalence, thus producing an aggregated CTMC model. An algorithm for computing the weak-bisimulation quotient is at hand [DHS03]. It has cubic time complexity in the number of states of the original model.

Example 3.16 (Weakly Bisimilar Representations). *Figure 3.13 depicts two PH representations. The representations are weakly bisimilar, as their starting states are. For convenience, states belonging to the same weak-bisimulation equivalence class have the same labels. Thus, for instance, the three states labelled with 4 on the left belong to the same class, and can thus be lumped, resulting in the model on the right.*

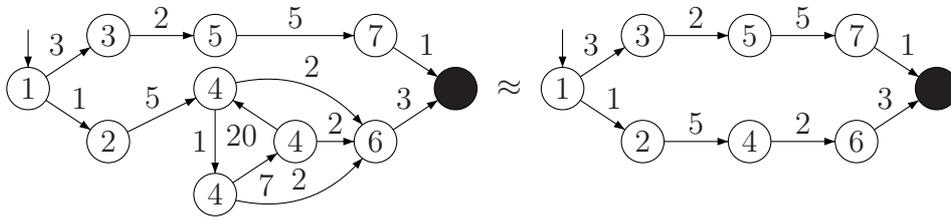


Figure 3.13: Two Weakly-Bisimilar Representations

Weak bisimulation and lumping play an important role in CTMC modelling and analysis. Weakly-bisimilar states and models possess the same probabilistic reachability properties [BKHW05], which means that their probability distributions of reaching certain subsets of the state space are equal. Hence weakly-bisimilar models are exchangeable so far as their reachability properties are concerned. Weak bisimulation can be used to identify bisimilar states in a model, and by lumping these bisimilar states, to reduce the size of the model. Weak-bisimulation-based lumping can also be applied to PH representations, without affecting their distributions.

Lemma 3.17. *Let $(\vec{\alpha}, \mathbf{A})$ be a phase-type representation. If $(\vec{\beta}, \mathbf{B})$ is obtained by lumping weakly-bisimilar states in $(\vec{\alpha}, \mathbf{A})$, then $\text{PH}(\vec{\alpha}, \mathbf{A}) = \text{PH}(\vec{\beta}, \mathbf{B})$ and the size of $(\vec{\beta}, \mathbf{B})$ is at most equal to the size $(\vec{\alpha}, \mathbf{A})$.*

The proof is a straightforward consequence of the fact that weak bisimulation does not alter transient probabilities of state classes [BKHW05], together with the particular role played by absorbing states in our variation of weak bisimulation. We now make precise the relation between our reduction algorithm and weak bisimulation.

Consider the representation depicted in Figure 3.7. The LST of the representation expressed in L -terms is given in Equation (3.9). Applying Lemma 3.9 with $i = 2$ to Equation (3.9), we obtain

$$\frac{1}{5} + \frac{1}{5}L(1) = \frac{2}{5}L(2), \quad (3.16)$$

which is divisible by $L(2)$. Removing the first $L(2)$ results in a new representation whose LST in L -terms is

$$\tilde{f}'(s) = \frac{\frac{2}{5} + \frac{2}{5}L(1) + \frac{1}{5}L(1)L(2)}{L(1)L(2)L(3)L(4)L(5)L(5)}. \quad (3.17)$$

Figure 3.14(a) depicts the Cox representation of the ordered bidiagonal representation in Figure 3.7. Theorem 3.5 provides the procedure to transform the ordered bidiagonal representation to its Cox representation. Observe that states 1 and 2 are weakly bisimilar. Figure 3.14(b) shows the Cox representation whose LST in L -terms is given in Equation (3.17). This representation is obtained when the two weakly-bisimilar states in the first Cox representation are lumped (denoted by state 21). This procedure is repeated in Figures 3.14(c)–(e), which will be described in more detail in Example 3.19.

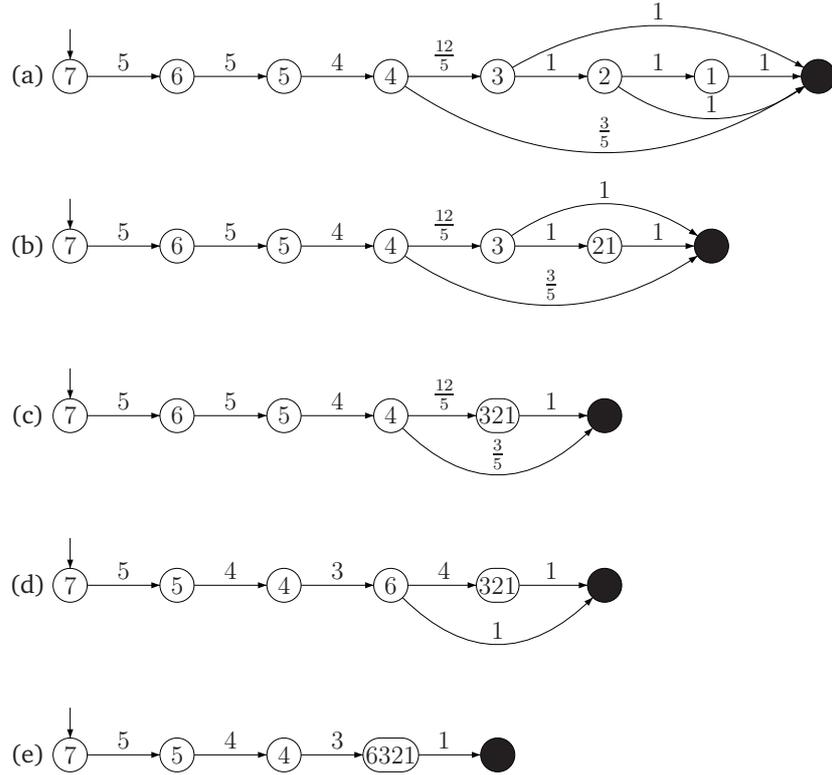


Figure 3.14: (a) The Cox Representation of Figure 3.7 and (b)–(e) its Weakly-Bisimilar Equivalents

Lemma 3.18. Let $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$ be an ordered bidiagonal representation of size n . If

1. $\vec{\beta}_2 \neq 0$, and
2. $\vec{\beta}_1 + \vec{\beta}_2 L(\lambda_1) = (\vec{\beta}_1 + \vec{\beta}_2) L(\lambda_i)$, for some $i \in \{2, \dots, n\}$ and $\vec{\beta}_j = 0$ for all $2 < j \leq i$,

then both $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$ and $(\vec{\gamma}, \mathbf{Bi}(\lambda_1, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots, \lambda_n))$ represent the same phase-type distribution for some sub-stochastic vector $\vec{\gamma}$ of dimension $n - 1$. Moreover, there is a weak bisimulation relating their respective Cox representations.

Proof. Assume now the conditions of the lemma to be true. The divisibility of the numerator polynomial by $L(\lambda_i)$ is straightforward. For the sub-stochasticity of vector $\vec{\gamma}$, observe that by the given conditions: $\vec{\gamma}_1 = \vec{\beta}_1 + \vec{\beta}_2$, and $\vec{\gamma}_j = \vec{\beta}_{j+1}$ for all $2 \leq j \leq n - 1$. Hence state s_i is removable and both representations are of the same PH distribution.

By Theorem 3.5, the ordered bidiagonal representation can also be represented by a Cox representation

$$(\vec{\kappa}, \mathbf{C}\mathbf{x}([\lambda_n, x_n], \dots, [\lambda_i, 1], \dots, [\lambda_2, x_2], \lambda_1)),$$

where $\vec{\kappa} = [\vec{\beta}e, 0, \dots, 0]$. By Equation (3.1) in the same theorem, $x_j = 1$ for all $2 < j \leq i$. Therefore, states s_k , for $2 \leq k \leq i$, can be reordered, and we have $(\vec{\kappa}, \mathbf{C}\mathbf{x}([\lambda_n, x_n], \dots, [\lambda_{i-1}, 1], \dots, [\lambda_2, 1], [\lambda_i, x_2], \lambda_1))$.

Rearranging $\vec{\beta}_1 + \vec{\beta}_2 L(\lambda_1) = (\vec{\beta}_1 + \vec{\beta}_2) L(\lambda_i)$ (recall that $L(\lambda) = \frac{s+\lambda}{\lambda}$), we obtain

$$\lambda_1 = \frac{\vec{\beta}_2}{\vec{\beta}_1 + \vec{\beta}_2} \lambda_i. \quad (3.18)$$

Inspecting the Cox representation or evaluating Equation (3.1), we can infer that the following relations hold

$$\begin{aligned} x_n x_{n-1} \cdots x_3 x_2 &= \vec{\beta}_1, \text{ and} \\ x_n x_{n-1} \cdots x_3 (1 - x_2) &= \vec{\beta}_2. \end{aligned} \quad (3.19)$$

From Equations (3.18) and (3.19): $\lambda_1 = (1 - x_2) \lambda_i$, which means that states s_1 and s_i are weakly bisimilar. Lumping the two states, we obtain

$$(\vec{\kappa}, \mathbf{C}\mathbf{x}([\lambda_n, x_n], \dots, [\lambda_{i-1}, 1], \dots, [\lambda_2, 1], \lambda_1))$$

with vector $\vec{\kappa}$ of appropriate dimension. This representation is the Cox representation of $(\vec{\gamma}, \mathbf{B}\mathbf{i}(\lambda_1, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots, \lambda_n))$. Hence both representations are related by a weak bisimulation. \square

The lemma indicates that once an APH representation is transformed into its ordered bidiagonal representation, it may be possible to reduce the state space by weak bisimulation. In the circumstances described in the lemma, our reduction algorithm is comprised in a weak-bisimulation aggregation on the Cox representation of the APH distribution.

Example 3.19 (Minimization with Weak Bisimulation). *Returning to the ordered bidiagonal representation depicted in Figure 3.7 and its Cox representation depicted in Figure 3.14(a), we immediately identify that states 1, 2, and 3 are weakly bisimilar (cf. Figures 3.14(a)–(c)). Since states 4, 5 and 6 can be reordered, state 6 can be placed after state 4 (cf. Figure 3.14(d)). Because the branching probabilities are retained when state 6 replaces state 4, the transition rate from state 6 to the absorbing state, then, becomes 1. Hence, state 6 is also weakly bisimilar to state 3, 2 and 1—or 321—(cf. Figure 3.14(e)).*

This fact can also be established by applying Lemma 3.18 to the numerator polynomial of Equation (3.9) as follows

$$\begin{aligned} & \frac{1}{5} + \frac{1}{5}L(1) + \frac{2}{5}L(1)L(2) + \frac{1}{5}L(1)L(2)L(2), \\ &= \frac{2}{5}L(2) + \frac{2}{5}L(2)L(1) + \frac{1}{5}L(1)L(2)L(2), \\ &= \frac{4}{5}L(2)L(2) + \frac{1}{5}L(2)L(2)L(1) = L(2)L(2)L(5). \end{aligned}$$

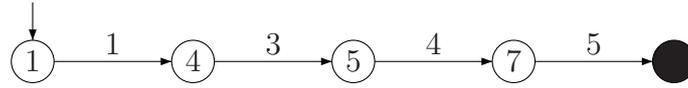


Figure 3.15: A Minimal Representation of Figures 3.13, 3.7, and 3.14

By lumping these four weakly-bisimilar states, we obtain their minimal representation depicted in Figure 3.15. We know this representation is minimal because it is a hypoexponential representation, and every hypoexponential representation is minimal (cf. Lemma 2.32).

In Equation (3.16), an L -term is added to a constant. The general rule of this addition operation is

$$L(x) + n = (n + 1)L((n + 1)x). \quad (3.20)$$

This general rule, in itself, actually establishes the weak-bisimilarity relations described in Lemma 3.18. This can be seen by rearranging Equation (3.20) as follows

$$\frac{1}{n+1} \frac{1}{L((n+1)x)} + \frac{n}{n+1} \frac{1}{L((n+1)x) L(x)} = \frac{1}{L(x)}.$$

If we express the last equation in a graphical way, we obtain the two representations in Figure 3.16. Weak bisimilarity relates the two representations in the figure.

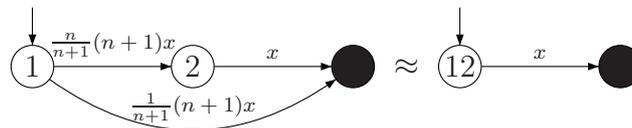


Figure 3.16: Weak Bisimilarity in the Conditions of Lemma 3.18

3.7 Conclusion

This chapter has introduced an algorithm to reduce the size of APH representations (viz. acyclic absorbing CTMCs) of APH distributions. In each iteration, the algorithm requires quadratic time (in the current number of states) to reduce the representation by at least one state, as long as no further reduction is possible. We have also clarified the relation between the reduction algorithm and the weak-bisimulation-based lumping.

In practice, one may wonder whether it is beneficial to run an overall cubic algorithm to reduce the size of the matrix representation of a PH distribution. This, of course, depends on the application context. If one intends to numerically compute the absorption probability at many time points (or at a single large time point), it might very well be worthwhile to run the suggested algorithm as a pre-processing step. This step reduces the dimensions of the involved matrices and vectors, and hence speeds up the subsequent (usually uniformization-based) iterations.

Furthermore, if the PH representation is used in a concurrency context, then a one-state saving in a single component—prior to exploring the cross product with other components in the system—saves states in the order of the entirety of all other components. This can, in general, lead to an exponential saving in the size of the overall system.

The approach is, in certain cases, even valuable for cyclic representations. The example in Figure 3.13 is a cyclic absorbing CTMC, yet it can be reduced to the representation in Figure 3.15. This requires first to apply the (cubic) weak-bisimulation-based lumping algorithm of [DHS03], and then our (cubic) reduction algorithm. The main point here is that the weak-bisimulation step returns an acyclic representation, which may look a bit artificial. That the algorithm works well on non-artificial examples has been shown by the fault tree example we considered.

Chapter 4

Operations on Erlang Distributions

Analysis of models comprising concurrent stochastic processes leads to state space explosions, just like in many other areas of model-based validation. The size of a model usually grows exponentially in the number of involved processes. Many such processes are governed by Erlang distributions.

An Erlang distribution is formed by the convolution of several exponential distributions of the same rate. The class of Erlang distributions is a subset of APH distributions. They are important in stochastic modelling for their suitability to approximate deterministic distributions or fixed delays [ACW92, MPvdL96]. For instance, according to [MPvdL96], an Erlang distribution with rate 5 and phase 5 is, for most general purposes, sufficient to approximate a fixed delay at time point 1. Better approximations can be achieved by increasing the phase while retaining the mean value of the Erlang distribution.

In this chapter, we study the effects of three operations—convolution, minimum and maximum—on Erlang distributions. A sequential execution of Erlang distributions corresponds to their convolution. The minimum operation, on the other hand, can be interpreted as a race among Erlang distributions: when several Erlang distributed events are started at the same time, the minimum arises if one has to wait for any of them to finish. In other words, it signals the occurrence of the earliest event in a concurrent execution. A concurrent execution of Erlang distributions corresponds to their maximum. If several Erlang distributed events are started at the same time, the maximum arises if one has to wait for all of them to finish. The maximum of several Erlang distributions signals the occurrence of the latest event in such a concurrent execution.

Of particular interest is the fact that Markovian representations of Erlang distributions always come as minimal representations (cf. Lemma 2.32). This is also the case for representations obtained by convolution operations on Erlang distributions, as we will show in Section 4.2. This means that when dealing with Erlang distributions or their convolutions, there seems to be not much potential when striving to reduce the size of their representations.

However, if one moves to the minimum or maximum of Erlang distributions, the picture becomes different. We will see that there is a substantial potential in reducing the size of resulting representations produced by these two operations: a significant number of states in the produced representations are redundant.

Related Work The notion of basic series, which we refine in this chapter, is introduced by CUMANI in [Cum82] as a tool to prove the transformation of an APH representation to its ordered bidiagonal representation. We are not aware of any previous work on the study of the above-mentioned operations on Erlang distributions, or on the more general PH distributions, aside from [BH02]. In that paper, the authors devised a simplified computation of the *mean value* of the maximum of PH distributions for an approximative compositional idea. A new method for computing this mean value and the expected total time spent in a set of transient states before absorptions in such circumstances was proposed in [Bre07].

Contribution The contributions of the chapter are twofold. First, we develop a constructive method to generate minimal representations of the convolution, minimum, and maximum of Erlang distributions. For convolution and minimum operations, the size of the state space of minimal representations is equal to the number of states in all involved distributions, and it grows linearly in the number of involved distributions. On the other hand, even in minimal representations, the state space produced by maximum operations grows exponentially in the number of involved distributions.

Secondly, as a by-product, we improve upon CUMANI's method for transforming an APH representation to its ordered bidiagonal representation by introducing the notion of the *core series*. This notion provides a useful tool to reason about the structure of the ordered bidiagonal representations of APH representations, and, in many cases, allows us to obtain smaller ordered bidiagonal representations than in the original transformation. Specifically, using the notion of the core series, we can remove redundant multiplicities of states with the same total outgoing rate from the representation.

Structure The chapter is organized as follows: In Section 4.1, we describe the notion of the core series. In Sections 4.2, 4.3 and 4.4, we investigate several properties of the convolution, minimum, and maximum, respectively, of an arbitrary number of Erlang distributions. We also study the form of the representations produced by each operation, and reason about their minimal representations. We summarize and conclude the chapter in Section 4.5.

4.1 Refining the Basic Series

In Section 3.2, we described a method to obtain the ordered bidiagonal representation of a given APH representation $(\vec{\alpha}, \mathbf{A})$. The method proceeds via the set $Paths(\mathcal{M})$, where $\mathcal{M} = (S, \mathbf{R}, \vec{\pi})$ is the underlying absorbing Markov chain of the APH representation. Since the CTMC \mathcal{M} is finite and acyclic, the set $Paths(\mathcal{M})$ is finite in size, and each path $\sigma \in Paths(\mathcal{M})$ is also of finite length. The elementary series we described in that section correspond to the elements of $Paths(\mathcal{M})$.

Assume that a PH representation $(\vec{\alpha}, \mathbf{A})$ is acyclic and of size n , and has total outgoing rates $\lambda_n \geq \lambda_{n-1} \geq \dots \geq \lambda_1 > 0$, so ordered by their magnitudes. Now consider the family $\langle \mathfrak{B}_1, \dots, \mathfrak{B}_n \rangle$ of n hypoexponential representations

$$\mathfrak{B}_i = (\vec{e}_1, \mathbf{Bi}(\lambda_i, \dots, \lambda_n)),$$

for $1 \leq i \leq n - 1$, and $\mathfrak{B}_n = (\vec{e}_1, \mathbf{Bi}(\lambda_n))$. These hypoexponential representations correspond to the basic series we described in Section 3.2.

Lemma 4.1. *For each path $\sigma \in Paths(\mathcal{M})$, there are $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{R}_{\geq 0}$, where $0 \leq \alpha_i \leq 1$, for $1 \leq i \leq n$, and $\sum_{i=1}^n \alpha_i = 1$, such that*

$$PH(\sigma) = \sum_{i=1}^n \alpha_i PH(\mathfrak{B}_i).$$

The lemma is a restatement of Lemma 1 in [Cum82], and thus of Theorem 3.2.

As described in Lemma 2.11, APH representation $(\vec{\alpha}, \mathbf{A})$ is a convex combination of its paths. Hence this lemma implies that the representation is, furthermore, a convex combination of its basic series.

Therefore, a particular convex combination of the basic series of \mathcal{M} constitutes a canonical representation of the APH representation. In the following, we show that the basic series as defined in [Cum82] are sometimes redundant. We again assume that the given PH representation is acyclic and of size n . Furthermore, the representation has $m \leq n$ distinct total outgoing rates $\lambda_{(m)} > \lambda_{(m-1)} > \dots > \lambda_{(1)} > 0$, so strictly ordered by their magnitudes.

Let $c(\sigma, \lambda)$ denote the number of occurrences of states with total outgoing rate λ in path σ . We extend this to $Paths(\mathcal{M})$ by defining

$$c(\mathcal{M}, \lambda) = \max_{\sigma \in Paths(\mathcal{M})} c(\sigma, \lambda).$$

Thus $c(\mathcal{M}, \lambda)$ denotes the maximum number of occurrences of states with total outgoing rate λ on any path in CTMC \mathcal{M} . Our main observation is that $c(\mathcal{M}, \lambda)$ can, for some particular rate λ , be considerably smaller than the number of states with total outgoing rate λ in CTMC \mathcal{M} .

We refine the definition of the basic series in the following, and then prove that each of the elementary series is still a convex combination of the thus refined series. For $1 \leq i \leq m$, let $k_i = \sum_{h=1}^i c(\mathcal{M}, \lambda_{(h)})$, $k_0 = 0$, and $l = k_m$. Thus, k_i denotes the maximum number of occurrences of states with total outgoing rates $\lambda_{(h)}$, for all $1 \leq h \leq i$, in any path of CTMC \mathcal{M} .

Obviously $m \leq l$. Furthermore, $l \leq n$, since for each λ occurring in \mathcal{M} , the chain contains at least $c(\mathcal{M}, \lambda)$ distinct states with total outgoing rate λ . The acyclicity of the chain is important in this observation, which gives us, ranging over all m distinct rates, the bound l for the number of states.

Definition 4.2 (Core Series). *Consider the family $\langle \mathfrak{C}_1, \dots, \mathfrak{C}_l \rangle$ of l hypoexponential representations: $\mathfrak{C}_l = (\vec{e}_1, \mathbf{Bi}(\lambda_{(m)}))$, and for $1 \leq i \leq l - 1$*

$$\mathfrak{C}_i = (\vec{e}_1, \mathbf{Bi}(\lambda_i, \dots, \lambda_l)),$$

where $\lambda_j = \lambda_{(h)}$ such that $k_{h-1} < j \leq k_h$, for $i \leq j \leq l$. This family of hypoexponential representations is called the core series of CTMC \mathcal{M} .

In the definition, we duplicate $\lambda_{(h)}$ in the series of the total outgoing rates $c(\mathcal{M}, \lambda_{(h)})$ -times, where $c(\mathcal{M}, \lambda_{(h)})$ is the maximum number of occurrences of states with total outgoing rate $\lambda_{(h)}$ in any path of \mathcal{M} . For instance, the core series $\mathfrak{C}_{c(\mathcal{M}, \lambda_{(m)})+1}$ looks like

$$s_{l-c(\mathcal{M}, \lambda_{(m)})} \xrightarrow{\lambda_{(m-1)}} \underbrace{s_{l-c(\mathcal{M}, \lambda_{(m)})+1} \xrightarrow{\lambda_{(m)}} s_{l-c(\mathcal{M}, \lambda_{(m)})+2} \xrightarrow{\lambda_{(m)}} \dots \xrightarrow{\lambda_{(m)}} s_{l+1}}_{c(\mathcal{M}, \lambda_{(m)})\text{-times}}$$

while the corresponding hypoexponential representation looks like

$$(\vec{e}_1, \mathbf{Bi}(\lambda_{(m-1)}, \underbrace{\lambda_{(m)}, \dots, \lambda_{(m)}}_{c(\mathcal{M}, \lambda_{(m)})\text{-times}})).$$

Example 4.3. Figure 4.1 depicts an APH representation and its elementary series. Each of the elementary series is weighted by its occurrence probability. The basic series of this representation are

$$\begin{aligned} \mathfrak{B}_1 &= (\vec{e}_1, \mathbf{Bi}(1, 2, 2, 3, 4, 5, 5)), & \mathfrak{B}_2 &= (\vec{e}_1, \mathbf{Bi}(2, 2, 3, 4, 5, 5)), \\ \mathfrak{B}_3 &= (\vec{e}_1, \mathbf{Bi}(2, 3, 4, 5, 5)), & \mathfrak{B}_4 &= (\vec{e}_1, \mathbf{Bi}(3, 4, 5, 5)), \\ \mathfrak{B}_5 &= (\vec{e}_1, \mathbf{Bi}(4, 5, 5)), & \mathfrak{B}_6 &= (\vec{e}_1, \mathbf{Bi}(5, 5)), \\ \mathfrak{B}_7 &= (\vec{e}_1, \mathbf{Bi}(5)). \end{aligned}$$

The elementary series, in terms of their distribution functions, can be expressed as the convex combinations of the basic series as follows

$$\sigma_1 = \frac{3}{5}\mathfrak{B}_3 + \frac{2}{5}\mathfrak{B}_4 \quad \text{and} \quad \sigma_2 = \frac{4}{15}\mathfrak{B}_1 + \frac{4}{15}\mathfrak{B}_2 + \frac{1}{3}\mathfrak{B}_3 + \frac{2}{15}\mathfrak{B}_4.$$

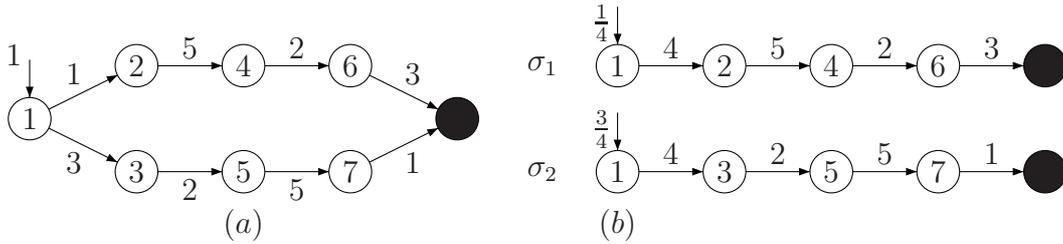


Figure 4.1: Acyclic Phase-Type Representation and its Elementary Series

Consider again the APH representation in Figure 4.1. Even though there are two states with total outgoing rate 2, and similarly with total outgoing rate 5, the maximum number of their occurrences in any path is $c(\mathcal{M}, 2) = 1$ and $c(\mathcal{M}, 5) = 1$. Therefore, the core series of this representation are

$$\begin{aligned} \mathfrak{C}_1 &= (\vec{e}_1, \mathbf{Bi}(1, 2, 3, 4, 5)), & \mathfrak{C}_2 &= (\vec{e}_1, \mathbf{Bi}(2, 3, 4, 5)), \\ \mathfrak{C}_3 &= (\vec{e}_1, \mathbf{Bi}(3, 4, 5)), & \mathfrak{C}_4 &= (\vec{e}_1, \mathbf{Bi}(4, 5)), \\ \mathfrak{C}_5 &= (\vec{e}_1, \mathbf{Bi}(5)). \end{aligned}$$

Furthermore, in terms of their distribution functions, the elementary series can be expressed as the convex combinations of the core series as follows

$$\sigma_1 = \mathfrak{C}_2 \quad \text{and} \quad \sigma_2 = \frac{2}{3}\mathfrak{C}_1 + \frac{1}{3}\mathfrak{C}_2.$$

In the following we show that each path is also a convex combination of several core series. To do that we require several concepts and a lemma.

Let $\mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)$ be a PH-generator, and $\lambda_n \geq \lambda_{n-1} \geq \dots \geq \lambda_1 > 0$. The polytope of this PH-generator is a simplex (see Section 3.1.1). Let $\psi \subseteq \{1, 2, \dots, n\}$ and $\psi \neq \emptyset$. With each ψ we associate a hypoexponential representation $(\vec{e}_1, \mathbf{Bi}_\psi)$, where the PH-generator \mathbf{Bi}_ψ is built by all λ_i 's such that $i \in \psi$. Let Ψ denote the collection of all such ψ 's.

Lemma 4.4 (Restatement of Lemma 2.45). *For each $\psi \in \Psi$, the PH distribution associated with ψ is on the boundary of polytope $\text{PH}(\mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$, except for that associated with $\psi = \{1\}$, which resides inside the polytope.*

We can now establish the desired canonicity result.

Lemma 4.5. *For each path $\sigma \in \text{Paths}(\mathcal{M})$, there are $\alpha_1, \alpha_2, \dots, \alpha_l \in \mathbb{R}_{\geq 0}$, where $0 \leq \alpha_i \leq 1$, for $1 \leq i \leq l$, and $\sum_{i=1}^l \alpha_i = 1$, such that*

$$\text{PH}(\sigma) = \sum_{i=1}^l \alpha_i \text{PH}(\mathfrak{C}_i).$$

Proof. Let $\mathbf{Bi} := \mathbf{Bi}(\mu_1, \mu_2, \dots, \mu_l)$ be the PH-generator of \mathfrak{C}_1 . Consider the polytope of PH-generator \mathbf{Bi} . Since \mathbf{Bi} is an ordered bidiagonal PH-generator, it is PH-simple (cf. Theorem 3.4), and the point mass at zero δ and (\vec{e}_i, \mathbf{Bi}) , for $1 \leq i \leq l$, are therefore vertices of the polytope. However, notice that $(\vec{e}_i, \mathbf{Bi}) = \mathfrak{C}_i$, for $1 \leq i \leq l$. Hence to prove the lemma we just have to show that each PH distribution associated with each path $\sigma \in \text{Paths}(\mathcal{M})$ ($\text{PH}(\sigma)$) resides in the polytope.

Construct the set Ψ as described in Lemma 4.4. We have to show that for each $\sigma \in \text{Paths}(\mathcal{M})$, we can find $\psi \in \Psi$ such that the associated representation of ψ is equivalent to the representation of σ .

Let $\sigma \in \text{Paths}(\mathcal{M})$ be an arbitrary path of length k (which is at most equal to l) given by $\sigma = (\vec{e}_1, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_k))$. First of all, by the definition of the core series, each λ_i , for $1 \leq i \leq k$, must occur in the core series. This means that we can define a mapping

$$f : \{1, \dots, k\} \rightarrow \{1, \dots, l\}$$

such that $\lambda_i = \mu_{f(i)}$, and thus $\sigma = (\vec{e}_1, \mathbf{Bi}(\mu_{f(1)}, \mu_{f(2)}, \dots, \mu_{f(k)}))$. If we, in addition, establish that f can be an injective mapping, then $\text{range}(f)$ is a subset ψ of the form required above, and $\sigma = (\vec{e}_1, \mathbf{Bi}_\psi)$, and we can conclude the proof.

The mapping f can be injective, because for each distinct λ occurring in σ , the number of its occurrences in σ is at most $c(\mathcal{M}, \lambda)$. Since

$$\{j \in \{1, \dots, l\} \mid \mu_j = \lambda\}$$

is of size $c(\mathcal{M}, \lambda)$, we can map the index subset

$$\{j \in \{1, \dots, k\} \mid \lambda_j = \lambda\}$$

injectively onto that set, and this holds for each distinct λ . □

4.2 Convolution Operation

In this section, we investigate the properties of the representation of the convolution of several Erlang distributions. We reason about the form of the representation produced by the convolution operation, and find out whether we can obtain its minimal representation. Consider Figure 4.2. It depicts a possible representation of $\text{con}(\text{Erl}(\lambda_1, k_1), \text{Erl}(\lambda_2, k_2))$, obtained by using Theorem 2.23(a). The resulting representation is a serial configuration of the two Erlang representations.

The main observation we obtain from the figure is the fact that the depicted representation consists of a single path from the starting to the absorbing state. This stays valid even when we extend the convolution to more than two Erlang representations. Furthermore, Erlang representations forming the convolution can be reordered (repositioned) in the serial configuration without altering the resulting distribution of the convolution. This can be easily shown in the LST domain, where convolution corresponds to multiplication, operation and multiplication is commutative.

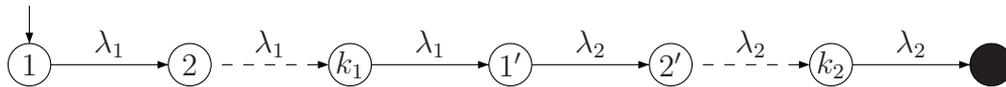


Figure 4.2: A Representation of $\text{con}(\text{Erl}(\lambda_1, k_1), \text{Erl}(\lambda_2, k_2))$

Further observation from Figure 4.2 is that the depicted representation is the same as a convolution of several exponential distributions of possibly different rates. Hence the representation is a hypoexponential representation. But then every hypoexponential representation is minimal (cf. Lemma 2.32).

To conclude: Given n Erlang distributions $\text{Erl}(\lambda_i, k_i)$, for $1 \leq i \leq n$, a minimal representation of the convolution of the n Erlang distributions can always be obtained, and it is of size $\sum_{i=1}^n k_i$. Therefore, the size of the minimal representation of the convolution of n Erlang distributions grows linearly in the number of involved Erlang distributions n .

4.3 Minimum Operation

In this section, we investigate the properties of the representation of the minimum of several Erlang distributions. As in the previous sections, we reason about the form of the representation produced by the minimum operation, and, by using the notion of the core series, obtain its minimal representation.

4.3.1 The Minimum of Two Erlang Distributions

Figure 4.3 depicts a representation of $\min(\text{Exp}(\lambda_1), \text{Exp}(\lambda_2))$. The representation is an exponential distribution with rate $\lambda_1 + \lambda_2$. The class of exponential distributions then is closed under minimum operations.

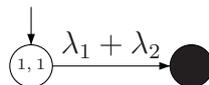


Figure 4.3: A Representation of $\min(\text{Exp}(\lambda_1), \text{Exp}(\lambda_2))$

Figure 4.4 depicts the representation of $\min(\text{Erl}(\lambda_1, k_1), \text{Erl}(\lambda_2, k_2))$, obtained by using Theorem 2.23(b). Observing the figure, we can list several interesting properties of this representation as follows.

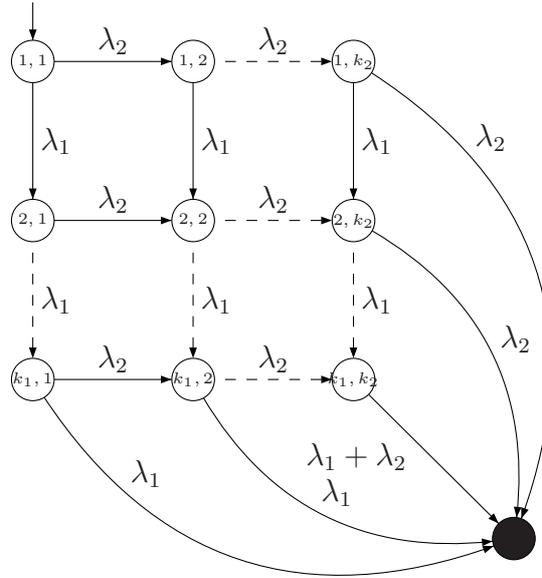


Figure 4.4: A Representation of $\min(\text{Erl}(\lambda_1, k_1), \text{Erl}(\lambda_2, k_2))$

Property 4.6. *There are $k_1 k_2$ states in the representation. Each state has total outgoing rate $\lambda_1 + \lambda_2$.*

Property 4.7. *Assume that $k_1 \leq k_2$. Each path in the representation is of length ranging from k_1 to $k_1 + k_2 - 1$. Since all states in all paths have total outgoing rate $\lambda_1 + \lambda_2$, there are at most $k_1 + k_2 - 1$ states with total outgoing rate $\lambda_1 + \lambda_2$, and at least k_1 states with total outgoing rate $\lambda_1 + \lambda_2$ traversed in any path.*

All assertions in the previous property are straightforward, except for the second. The shortest path can be achieved by moving downward to state $(k_1, 1)$ (cf. Figure 4.4), and then hitting the absorbing state. Any path that passes through state (k_1, k_2) is of the longest.

4.3.2 The Minimum of More Erlang Distributions

We generalize the concept to more than two Erlang distributions. Assume that we have n Erlang distributions $\text{Erl}(\lambda_i, k_i)$, for $1 \leq i \leq n$. To obtain the standard representation of the minimum of n Erlang distributions, Theorem 2.23(b) can be applied $n - 1$ times. Based on the obtained representation, we can generalize Properties 4.6 and 4.7, as follows.

Property 4.8. *There are $\prod_{i=1}^n k_i$ states in the representation. Each state has total outgoing rate $\sum_{i=1}^n \lambda_i$.*

This is straightforward from the properties of Kronecker plus operator (\oplus) and the PH-generators of each Erlang distributions involved. As an example, consider the graph in Figure 4.5, which represents the minimum of three Erlang representations. States labelled with ①, ②, and ③ correspond to the states that have outgoing transitions with rate λ_1 , λ_2 , and λ_3 , respectively, to the absorbing state. States labelled with ④, ⑤, and ⑥ correspond to the states that have outgoing transitions with rate $\lambda_1 + \lambda_2$,

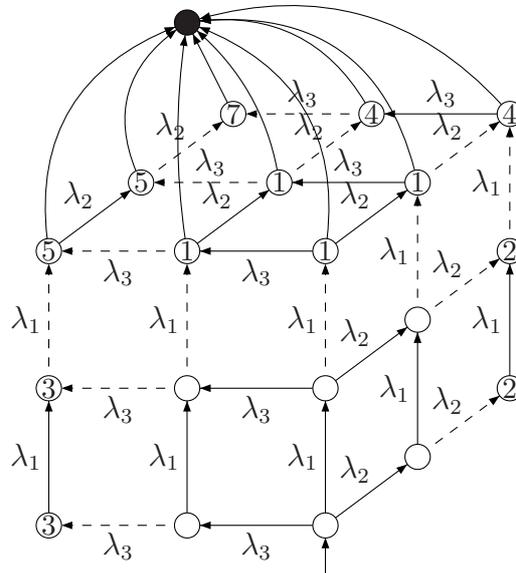
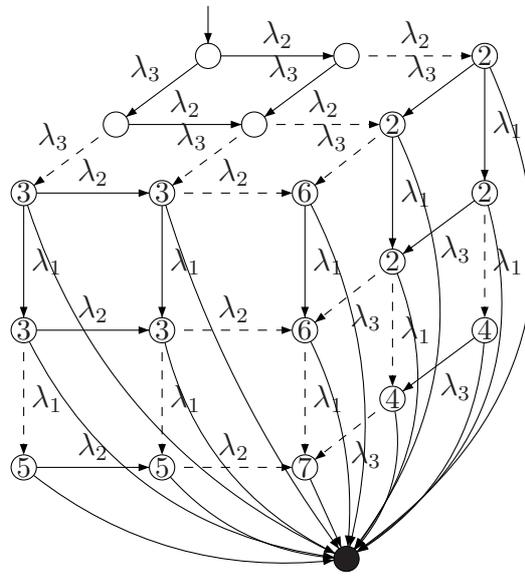


Figure 4.5: A representation of the minimum of $\text{Erl}(\lambda_1, k_1)$, $\text{Erl}(\lambda_2, k_2)$ and $\text{Erl}(\lambda_3, k_3)$, showing two opposite sides. For the sake of clarity, in the first figure we omit the transitions from states labelled with ①, while in the second figure we omit the transitions from states labelled with ②, ③, and ⑥.

$\lambda_1 + \lambda_3$, and $\lambda_2 + \lambda_3$, respectively, to the absorbing state. The single transition from the state labelled with $\textcircled{7}$ to the absorbing state has rate $\lambda_1 + \lambda_2 + \lambda_3$.

Property 4.9. *Assume that k_1 is the minimum of k_i , for $1 \leq i \leq n$. Each path in the representation is of length ranging from k_1 and to $\sum_{i=1}^n k_i - n + 1$. Since all states in all paths have total outgoing rate $\sum_{i=1}^n \lambda_i$, there are at most $\sum_{i=1}^n k_i - n + 1$ states with total outgoing rate $\sum_{i=1}^n \lambda_i$, and at least k_1 states with total outgoing rate $\sum_{i=1}^n \lambda_i$ traversed in any path.*

4.3.3 The Minimal of the Minimum

We now introduce a representation of the minimum of Erlang distributions that is minimal.

Let $\langle \mathfrak{C}_1, \dots, \mathfrak{C}_l \rangle$ be the core series of the standard representation of the minimum of n Erlang distributions $\text{Erl}(\lambda_i, k_i)$, for $1 \leq i \leq n$. Let \mathbf{Bi} be the ordered bidiagonal PH-generator associated with \mathfrak{C}_1 .

Lemma 4.10. *For some stochastic vector $\vec{\beta}$, $\text{MinErl} := (\vec{\beta}, \mathbf{Bi})$ is a representation of the minimum of n Erlang distributions $\text{Erl}(\lambda_i, k_i)$, for $1 \leq i \leq n$, and it has $\sum_{i=1}^n k_i - n + 1$ states, each with total outgoing rate $\sum_{i=1}^n \lambda_i$.*

Proof. That MinErl is a representation of the minimum of Erlang distributions $\text{Erl}(\lambda_i, k_i)$, for $1 \leq i \leq n$, is straightforward from the construction of the core series and Lemma 4.5. The stochastic vector $\vec{\beta}$ provides the parameters of the convex combination. The vector can be obtained by using the spectral polynomial algorithm of [HZ06b]. Property 4.9 and the way we construct the core series as described in Definition 4.2 assure us that the representation has $\sum_{i=1}^n k_i - n + 1$ states, each with total outgoing rate $\sum_{i=1}^n \lambda_i$. \square

The number of states in representation MinErl is $\sum_{i=1}^n k_i - n + 1$. This means that this representation of the minimum of Erlang distributions grows linearly in n . The main question right now is whether this representation is minimal.

Lemma 4.11. *The representation MinErl is minimal.*

Proof. MinErl is an ordered bidiagonal representation of size $l = \sum_{i=1}^n k_i - n + 1$. The LST of this representation—following Equation (3.8)—can be written as

$$\tilde{f}(s) = \frac{\vec{\beta}_1 + \vec{\beta}_2 L(\lambda) + \dots + \vec{\beta}_{l-1} L(\lambda)^{l-2} + \vec{\beta}_l L(\lambda)^{l-1}}{L(\lambda)^l},$$

where $\lambda = \sum_{i=1}^n \lambda_i$. Now, for MinErl to be of size l , $0 < \vec{\beta}_1 \leq 1$ must hold. We will show that none of the states of the representation is removable.

Assume that for $1 < i < l$, state i is removable, then

$$\vec{\beta}_1 + \vec{\beta}_2 L(\lambda) + \dots + \vec{\beta}_i L(\lambda)^{i-1}$$

must be divisible by $L(\lambda)$. However this cannot be true since $\vec{\beta}_1 \neq 0$.

Since none of the states in the representation is removable, and the distribution is of algebraic degree l , we conclude that the algebraic degree of the distribution is equal to the size of the representation. This proves that the representation is minimal. \square

By virtue of Lemma 4.10 and Lemma 4.11, we can conclude that the size of the minimal representation of the minimum of n Erlang distributions grows linearly in the number of involved Erlang distributions n .

Example 4.12. *In this example, we demonstrate the steps described in the proof of Lemma 4.10 to obtain the minimal representation of the minimum of Erlang distributions with the help of Figure 4.6. Figure 4.6(a) depicts three Erlang distributions: $\text{Erl}(1, 2)$, $\text{Erl}(2, 2)$, and $\text{Erl}(4, 2)$. The standard representation of their minimum—i.e., $\min\{\text{Erl}(1, 2), \text{Erl}(2, 2), \text{Erl}(4, 2)\}$ —is shown in Figure 4.6(b). The standard representation consists of eight transient states, each with total outgoing rate 7. Note that in the figure, a label on a state indicates the rate of the transition from the state to the absorbing state. Let $(\vec{\alpha}, \mathbf{A})$ denote this standard representation.*

We can apply the spectral polynomial algorithm (cf. Section 3.3.3) directly on $(\vec{\alpha}, \mathbf{A})$, and we obtain that $\mathbf{AP} = \mathbf{PBi}(7, 7, 7, 7, 7, 7, 7, 7)$, where

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{48}{343} & \frac{148}{343} & \frac{3}{7} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{16}{49} & \frac{26}{49} & \frac{1}{7} \\ 0 & 0 & 0 & 0 & 0 & \frac{8}{49} & \frac{27}{49} & \frac{2}{7} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{4}{7} & \frac{3}{7} \\ 0 & 0 & 0 & 0 & 0 & \frac{4}{49} & \frac{17}{49} & \frac{4}{7} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{2}{7} & \frac{5}{7} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{7} & \frac{6}{7} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Hence, the ordered bidiagonal representation of $(\vec{\alpha}, \mathbf{A})$ is given by $(\vec{\beta}, \mathbf{Bi}(7, 7, 7, 7, 7, 7, 7, 7))$, where $\vec{\beta} = \vec{\alpha}\mathbf{P} = [0, 0, 0, 0, \frac{48}{343}, \frac{148}{343}, \frac{3}{7}, 0]$. This ordered bidiagonal representation is depicted in Figure 4.6(c).

Instead of directly applying SPA on the standard representation $(\vec{\alpha}, \mathbf{A})$, Lemma 4.10 maintains that $\min\{\text{Erl}(1, 2), \text{Erl}(2, 2), \text{Erl}(4, 2)\}$ can be represented by $(\vec{\beta}', \mathbf{Bi}(7, 7, 7, 7))$ as well, for some stochastic vector $\vec{\beta}'$. This is because $\mathbf{Bi}(7, 7, 7, 7)$ is the ordered bidiagonal PH-generator associated with the longest core series of the $(\vec{\alpha}, \mathbf{A})$. According to Property 4.9, there are at most $2 + 2 + 2 - 3 + 1 = 4$ states with total outgoing rate 7, and at least two states with total outgoing rate 7 traversed in any path in the underlying CTMC of $(\vec{\alpha}, \mathbf{A})$. Hence, the core series of the underlying CTMC are given by $(\vec{e}_1, \mathbf{Bi}(7)), (\vec{e}_1, \mathbf{Bi}(7, 7)), (\vec{e}_1, \mathbf{Bi}(7, 7, 7))$ and $(\vec{e}_1, \mathbf{Bi}(7, 7, 7, 7))$.

As described in the lemma, SPA can be used to obtain vector $\vec{\beta}'$. This is carried out by solving $\mathbf{AQ} = \mathbf{QBi}(7, 7, 7, 7)$, from which we obtain

$$\mathbf{Q} = \begin{bmatrix} \frac{48}{343} & \frac{148}{343} & \frac{3}{7} & 0 \\ 0 & \frac{16}{49} & \frac{26}{49} & \frac{1}{7} \\ 0 & \frac{8}{49} & \frac{27}{49} & \frac{2}{7} \\ 0 & 0 & \frac{4}{7} & \frac{3}{7} \\ 0 & \frac{4}{49} & \frac{17}{49} & \frac{4}{7} \\ 0 & 0 & \frac{2}{7} & \frac{5}{7} \\ 0 & 0 & \frac{1}{7} & \frac{6}{7} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Hence, by Lemma 4.10 and Lemma 4.11, the minimal ordered bidiagonal representation of $(\vec{\alpha}, \mathbf{A})$ is given by $(\vec{\beta}', \mathbf{Bi}(7, 7, 7, 7))$, where $\vec{\beta}' = \vec{\alpha}\mathbf{Q} = [\frac{48}{343}, \frac{148}{343}, \frac{3}{7}, 0]$. This ordered bidiagonal representation is depicted in Figure 4.6(d).

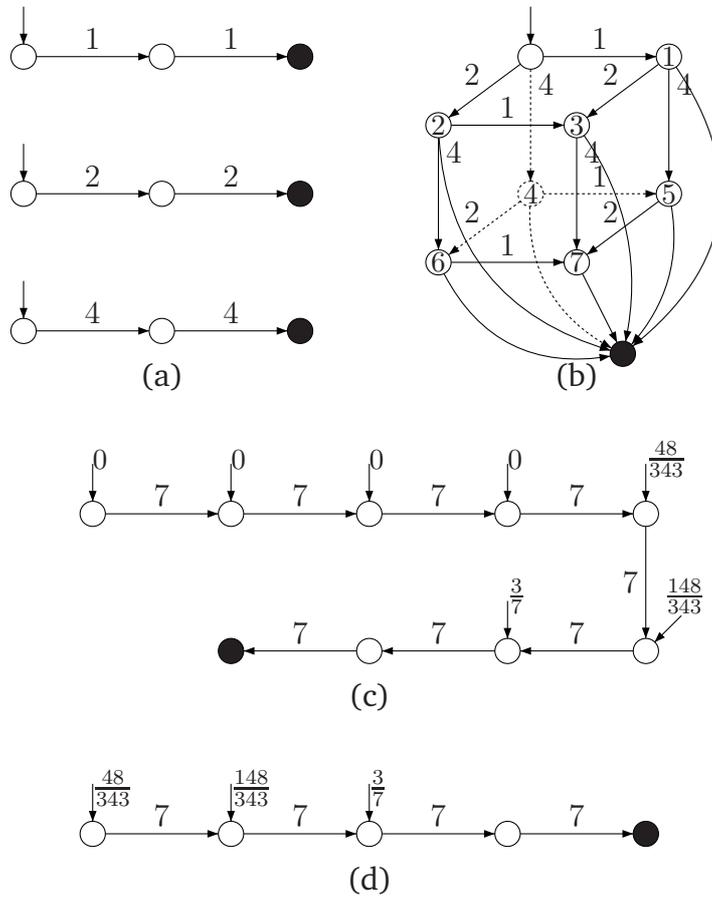


Figure 4.6: (a) Three Erlang representations, (b) The standard representation of the minimum of the three Erlang distributions: a label on a state indicates its rate to the absorbing state, (c) The ordered bidiagonal representation of the standard representation of the minimum produced by the spectral polynomial algorithm (SPA), and (d) The minimal representation of the minimum.

Note that for the case of the minimum of Erlang distributions, applying SPA on the standard representation by using the basic series (cf. Figure 4.6(c)) or by using the core series (cf. Figure 4.6(d)) yields the same result. This is because all states in the standard representation have the same total outgoing rate.

4.4 Maximum Operation

We continue our investigation with the maximum operation. We reason about the form of the representation produced by the maximum operation of several Erlang distributions. It turns out that the produced representation is much more complex. We first clarify several properties of the representation, before studying its minimal representation. We rely on the notion of the core series to obtain a smaller representation, and then we prove that this representation is minimal.

4.4.1 The Maximum of Two Erlang Distributions

Figure 4.7 depicts a representation of $\max(\text{Exp}(\lambda_1), \text{Exp}(\lambda_2))$. Since the representation is not an exponential distribution, we can conclude that the class of exponential distributions is not closed under maximum operations.

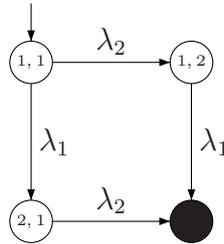


Figure 4.7: A Representation of $\max(\text{Exp}(\lambda_1), \text{Exp}(\lambda_2))$

We observe that the representation of $\max(\text{Exp}(\lambda_1), \text{Exp}(\lambda_2))$ has three states: one state with total outgoing rate λ_1 , one state with total outgoing rate λ_2 , and one state with total outgoing rate $\lambda_1 + \lambda_2$. Furthermore, there are exactly two paths from the starting to the absorbing state, each of length two.

Figure 4.8 depicts a representation of $\max(\text{Erl}(\lambda_1, k_1), \text{Erl}(\lambda_2, k_2))$, obtained by using Theorem 2.23(c). Observing the figure, we can list several interesting properties of this representation as follows.

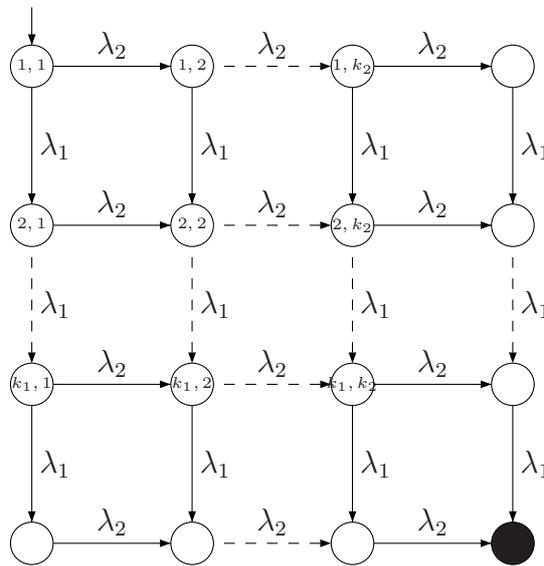


Figure 4.8: A Representation of $\max(\text{Erl}(\lambda_1, k_1), \text{Erl}(\lambda_2, k_2))$

Property 4.13. There are k_1 states with total outgoing rate λ_1 , k_2 states with total outgoing rate λ_2 , and $k_1 k_2$ states with total outgoing rate $\lambda_1 + \lambda_2$.

Property 4.14. There are $\frac{(k_1+k_2)!}{k_1!k_2!}$ distinct paths from the starting state to the absorbing state.

Since there are only two incoming transitions to the absorbing state, the number of distinct paths from the starting state to the absorbing state is the sum of the number of distinct paths from the starting state to state $(k_1, k_2 + 1)$ (i.e., the state to the right of (k_1, k_2)) and to state $(k_1 + 1, k_2)$ (i.e., the state below (k_1, k_2)). Therefore, we have a recursive function

$$N(k_1 + 1, k_2 + 1) = N(k_1, k_2 + 1) + N(k_1 + 1, k_2),$$

where $N(x, 1) = 1$ and $N(1, x) = 1$, for $x \in \mathbb{Z}_{\geq 0}$. Evaluating the recursive function we obtain

$$N(k_1 + 1, k_2 + 1) = \binom{k_1 + k_2}{k_1} = \binom{k_1 + k_2}{k_2} = \frac{(k_1 + k_2)!}{k_1!k_2!}. \quad (4.1)$$

The binomial functions in Equation (4.1) can also be obtained by observing that to generate a path in the representation we need to place k_2 λ_2 -states (or symmetrically k_1 λ_1 -states) in a path of length $k_1 + k_2$.

Property 4.15. *Each of these paths is of length $k_1 + k_2$. Let $k_1 \leq k_2$. There are at most $k_1 + k_2 - 1$ states with total outgoing rate $\lambda_1 + \lambda_2$, and at least k_1 states with total outgoing rate $\lambda_1 + \lambda_2$ traversed in any path. Furthermore, there are at most k_1 and k_2 states with total outgoing rate λ_1 and λ_2 , respectively, traversed in any path. However, at the least, we need only to traverse a single state with total outgoing rate λ_1 or λ_2 in any path.*

All assertions in the previous property are straightforward, except for the second. The least number of $(\lambda_1 + \lambda_2)$ -states can be achieved by moving downward k_1 times and then continuing the k_2 transitions for the rest of the traversal to the absorbing state. The most number of $(\lambda_1 + \lambda_2)$ -states can be achieved by just passing through λ_1 -states or λ_2 -states only once. Any path passing through state (k_1, k_2) satisfies this.

4.4.2 The Maximum of More Erlang Distributions

In the previous subsection we discussed the maximum of two Erlang distributions. In this section, we generalize the concept to more than two Erlang distributions. Assume for simplicity that we have n Erlang distributions $\text{Erl}(\lambda_i, k_i)$, for $1 \leq i \leq n$, where λ_i 's and all their possible sums are pair-wise distinct. The maximum of the n distributions can be obtained by multiplying their distribution functions. In terms of their representations, Theorem 2.23(c) can be applied $n - 1$ times to obtain the standard representation of the maximum. Based on the obtained representation, we can generalize Property 4.13–4.14, as follows.

Property 4.16. *For all $\psi \subseteq \{1, 2, \dots, n\}$, there are $\prod_{i \in \psi} k_i$ states with total outgoing rate $\sum_{i \in \psi} \lambda_i$.*

Since the maximum operation described in Theorem 2.23(c) is a cross-product operation, this property is straightforward. As an example, observe the graph in Figure 4.9, which represents the maximum of three Erlang distributions. The states labelled with ①, ②, and ③ correspond to the case where ψ is $\{1\}$, $\{2\}$, and $\{3\}$, respectively. The states labelled with ④, ⑤, and ⑥ correspond to the cases where ψ is $\{1, 2\}$, $\{1, 3\}$, and $\{2, 3\}$, respectively, while the states labelled with ⑦ correspond to $\psi = \{1, 2, 3\}$.

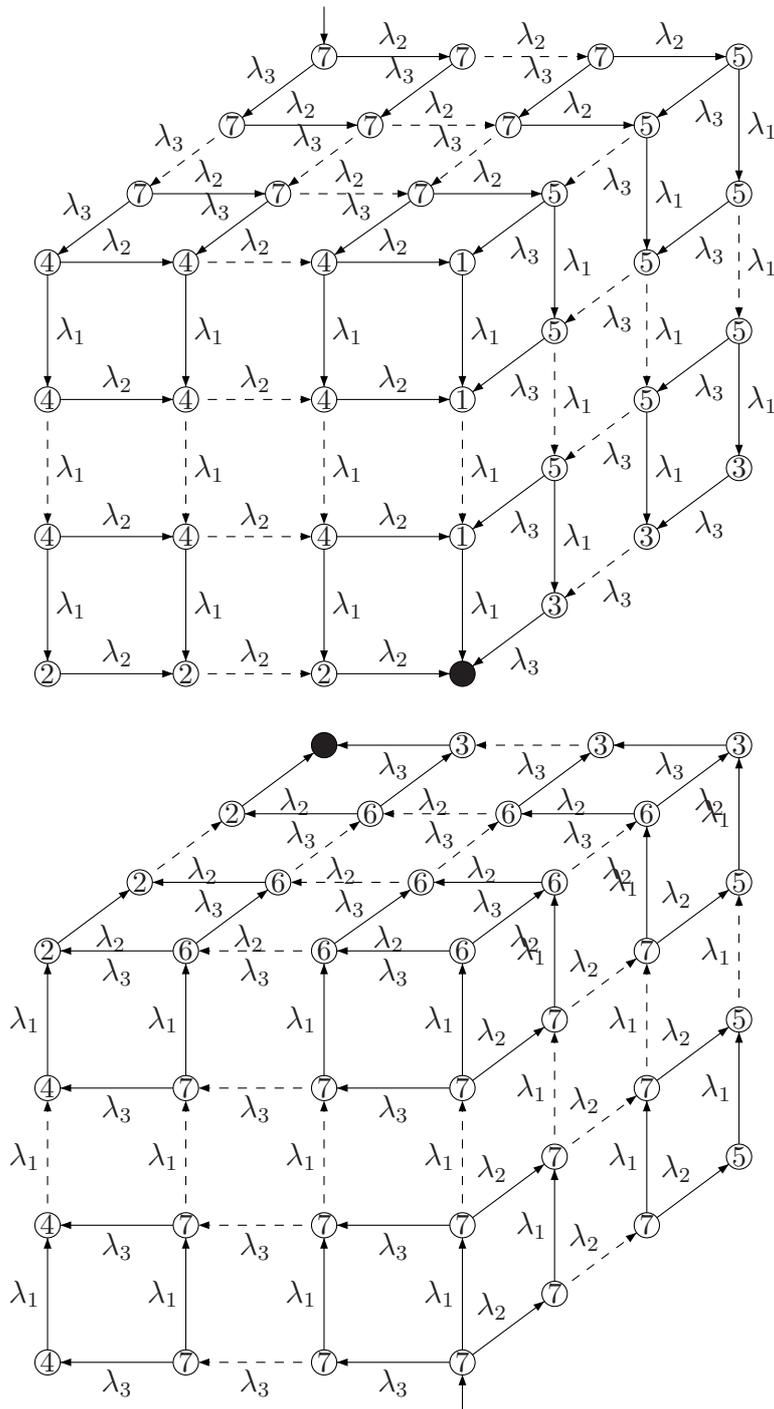


Figure 4.9: A representation of the maximum of $\text{Erl}(\lambda_1, k_1)$, $\text{Erl}(\lambda_2, k_2)$ and $\text{Erl}(\lambda_3, k_3)$, showing two opposite sides. The second cube is obtained by rotating the first cube 180° about the straight line from the leftmost state labelled with ② to the rightmost state labelled with ③.

Accumulating those states for all different sets ψ , we can conclude that the size of the representation is $\sum_{\forall\psi} \prod_{i \in \psi} k_i$.

We can think of ψ , in this case, as the set of Erlang distributions that have not yet finished as one traverses the representation from the starting state (i.e., when $\psi = \{1, 2, \dots, n\}$) to the absorbing state (i.e., when $\psi = \emptyset$).

Property 4.17. *There are $\frac{(\sum_{j=1}^n k_j)!}{\prod_{j=1}^n k_j!}$ distinct paths from the starting state to the absorbing state.*

Reasoning inductively, assume that we have the representation of the maximum of $n - 1$ Erlang distributions. Building the cross product of this representation with the n -th Erlang representation of length k_n means that each path in the $n - 1$ Erlang representation is extended by k_n . Furthermore, for each one of the paths we have to place k_n λ_n -states within a path of length $\sum_{i=1}^n k_i$. Hence the number of paths is

$$\begin{aligned} \prod_{i=2}^n \binom{\sum_{j=1}^i k_j}{k_i} &= \binom{k_1 + k_2}{k_2} \binom{k_1 + k_2 + k_3}{k_3} \dots \binom{\sum_{j=1}^n k_j}{k_n}, \\ &= \frac{(k_1 + k_2)!}{k_1!} \frac{(k_1 + k_2 + k_3)!}{(k_1 + k_2)!k_3!} \dots \frac{(\sum_{j=1}^n k_j)!}{(\sum_{j=1}^{n-1} k_j)!k_n!} = \frac{(\sum_{j=1}^n k_j)!}{\prod_{j=1}^n k_j!}. \end{aligned}$$

Property 4.18. *Each of these paths is of length $\sum_{i=1}^n k_i$. For $\psi \subseteq \{1, 2, \dots, n\}$, at most $\sum_{i \in \psi} k_i - |\psi| + 1$ states with total outgoing rate $\sum_{i \in \psi} \lambda_i$ are traversed in any path, and the paths containing $\sum_{i \in \psi} k_i - |\psi| + 1$ states with total outgoing rate $\sum_{i \in \psi} \lambda_i$ exist.*

The length of the paths is straightforward. According to Property 4.16, there are $\prod_{i \in \psi} k_i$ states with total outgoing rate $\sum_{i \in \psi} \lambda_i$. These states form a region in the representation. The regions are distinguished by their state labels in Figure 4.9. The region has a similar structure to the representation of the maximum of $\text{Erl}(\lambda_i, k_i - 1)$ for all $i \in \psi$. The special paths described in Property 4.18 are the paths that contain the longest sequence of $(\sum_{i \in \psi} \lambda_i)$ -states while traversing within the $(\sum_{i \in \psi} \lambda_i)$ -region. These are the paths in the representation of the maximum of $\text{Erl}(\lambda_i, k_i - 1)$ for all $i \in \psi$ whose length is $\sum_{i \in \psi} (k_i - 1)$. However, the last state in the smaller representation is also a $(\sum_{i \in \psi} \lambda_i)$ -state, hence they are of length

$$\sum_{i \in \psi} (k_i - 1) + 1 = \sum_{i \in \psi} k_i - |\psi| + 1.$$

Our region construction proves the existence of such special paths.

4.4.3 The Minimal of the Maximum

We are now in the position to introduce a representation of the maximum of Erlang distributions that is minimal.

Let $\langle \mathfrak{C}_1, \dots, \mathfrak{C}_l \rangle$ be the core series of the standard representation of the maximum of n Erlang distributions $\text{Erl}(\lambda_i, k_i)$, for $1 \leq i \leq n$. Let \mathbf{Bi} be the ordered bidiagonal PH-generator associated with \mathfrak{C}_1 .

Lemma 4.19. *For some stochastic vector $\vec{\beta}$, $\text{MaxErl} := (\vec{\beta}, \mathbf{Bi})$ is a representation of the maximum of n Erlang distributions $\text{Erl}(\lambda_i, k_i)$, for $1 \leq i \leq n$, and it has $\sum_{i \in \psi} k_i - |\psi| + 1$ states with total outgoing rate $\sum_{i \in \psi} \lambda_i$, for all $\psi \subseteq \{1, 2, \dots, n\}$.*

Proof. That MaxErl is a representation of the maximum of Erlang distributions $\text{Erl}(\lambda_i, k_i)$, for $1 \leq i \leq n$, is straightforward from the construction of the core series and Lemma 4.5. The stochastic vector $\vec{\beta}$ provides the parameters of the convex combination. The vector can be obtained by using the spectral polynomial algorithm [HZ06b]. Property 4.18 and the way we construct the core series as described in Definition 4.2 assure us that the representation has $\sum_{i \in \psi} k_i - |\psi| + 1$ states with total outgoing rate $\sum_{i \in \psi} \lambda_i$, for all $\psi \subseteq \{1, 2, \dots, n\}$. \square

The number of states in representation MaxErl is

$$\sum_{\forall \psi} \left(\sum_{i \in \psi} (k_i - 1) + 1 \right) = \sum_{\forall \psi} \sum_{i \in \psi} (k_i - 1) + 2^n - 1.$$

However, every member of $\{1, 2, \dots, n\}$ is involved in 2^{n-1} subsets, hence

$$\sum_{\forall \psi} \left(\sum_{i \in \psi} (k_i - 1) + 1 \right) = 2^{n-1} \sum_{i=1}^n (k_i - 1) + 2^n - 1. \quad (4.2)$$

Thus the size of the representation grows exponentially in n .

Recall that in Section 4.4.2, we assumed that we are given n Erlang distributions $\text{Erl}(\lambda_i, k_i)$, for $1 \leq i \leq n$, where λ_i 's and all their possible sums are pair-wise distinct. The assumption is necessary to allow us to derive the number of states (cf. Equation (4.2)) in representation MaxErl. Without the assumption, the equation only provides an upper bound of the number of states. Furthermore, Lemma 4.19 will not be valid without the assumption, as the structure described in the lemma will not be obtained.

Nevertheless, even without the assumption, we can still build the core series of the standard representation of the maximum. Using the core series, an ordered bidiagonal representation of the standard representation can then be formed. From now on, let MaxErl denote the ordered bidiagonal representation associated with the built core series regardless of the distinctness of the total outgoing rates.

Lemma 4.20. *The representation MaxErl is minimal.*

Proof. To show that representation MaxErl is minimal, we show that its size is equal to the algebraic degree of its distribution. Let l be the size (length) of the longest core series in the standard representation of the maximum. Then MaxErl must be of the form $(\vec{\beta}, \mathbf{Bi}(\mu_1, \mu_2, \dots, \mu_l))$. Let $\mathbf{Bi} := \mathbf{Bi}(\mu_1, \mu_2, \dots, \mu_l)$. We show that the algebraic degree of $\text{PH}(\vec{\beta}, \mathbf{Bi})$ is exactly l .

Consider the polytope of PH-generator \mathbf{Bi} , $\text{PH}(\mathbf{Bi})$. Since \mathbf{Bi} is an ordered bidiagonal PH-generator, it is PH-simple (cf. Theorem 3.4). Therefore, the polytope is l -dimensional, i.e., it resides in an l -dimensional affine subspace [Roc70]. Let $\psi \subseteq \{1, 2, \dots, l\}$ and $\psi \neq \emptyset$. With each ψ we associate a hypoexponential representation $q_\psi := (\vec{e}_1, \mathbf{Bi}_\psi)$, where the PH-generator \mathbf{Bi}_ψ is built by all μ_j 's such that $j \in \psi$. Let Ψ denote the collection of all such ψ . By Lemma 2.45, the associated PH distribution of each $\psi \in \Psi$ is on the boundary of the polytope.

Consider l polytopes

$$\begin{aligned} & \text{conv}(\{\delta, q_{\{l\}}, q_{\{l-1,l\}}, \dots, q_{\{3,\dots,l-1,l\}}, q_{\{2,3,\dots,l-1,l\}}\}), \\ & \text{conv}(\{\delta, q_{\{l\}}, q_{\{l-1,l\}}, \dots, q_{\{3,\dots,l-1,l\}}, q_{\{1,3,\dots,l-1,l\}}\}), \\ & \quad \vdots \\ & \text{conv}(\{\delta, q_{\{l\}}, q_{\{l-2,l\}}, \dots, q_{\{2,\dots,l-2,l\}}, q_{\{1,2,\dots,l-2,l\}}\}), \\ & \text{conv}(\{\delta, q_{\{l-1\}}, q_{\{l-2,l-1\}}, \dots, q_{\{2,\dots,l-2,l-1\}}, q_{\{1,2,\dots,l-2,l-1\}}\}), \end{aligned}$$

where $\text{conv}(\{x_1, \dots, x_n\})$ is the convex hull of the set $\{x_1, \dots, x_n\}$ (see Appendix A.3.2). Every point generating each of the polytopes lies on the boundary of polytope $\text{PH}(\mathbf{Bi})$, and therefore each of the polytopes is a subset of polytope $\text{PH}(\mathbf{Bi})$.

Furthermore, each of the smaller polytopes is an $(l-1)$ -dimensional polytope, *i.e.*, it resides in an $(l-1)$ -dimensional affine subspace. To show this, we take an arbitrary smaller polytope

$$\text{conv}(\{\delta, q_{\{l\}}, \dots, q_{\{i+1,\dots,l\}}, q_{\{i-1,i+1,\dots,l\}}, \dots, q_{\{1,\dots,i-1,i+1,\dots,l\}}\}),$$

for $1 \leq i \leq l$, and show that points

$$q_{\{l\}}, \dots, q_{\{i+1,\dots,l\}}, q_{\{i-1,i+1,\dots,l\}}, \dots, q_{\{1,\dots,i-1,i+1,\dots,l\}}$$

are linearly independent. Points $q_{\{l\}}, \dots, q_{\{i+1,\dots,l\}}$ correspond to $\vec{e}_l, \dots, \vec{e}_{i+1}$; therefore they are linearly independent. For the rest, according to Lemma 2.44 (or Lemma 6 in [DL82])

$$\begin{aligned} q_{\{i-1,i+1,\dots,l\}} & \in \text{conv}(\{q_{\{i,\dots,l\}}, q_{\{i-1,\dots,l\}}\}) = \text{conv}(\{\vec{e}_i, \vec{e}_{i-1}\}), \\ q_{\{i-2,i-1,i+1,\dots,l\}} & \in \text{conv}(\{q_{\{i-1,\dots,l\}}, q_{\{i-2,\dots,l\}}\}) = \text{conv}(\{\vec{e}_{i-1}, \vec{e}_{i-2}\}), \\ & \quad \vdots \\ q_{\{1,\dots,i-1,i+1,\dots,l\}} & \in \text{conv}(\{q_{\{2,\dots,l\}}, q_{\{1,\dots,l\}}\}) = \text{conv}(\{\vec{e}_2, \vec{e}_1\}), \end{aligned}$$

are also linearly independent of each other.

The intersection of all of the $(l-1)$ -dimensional affine subspaces in which the smaller polytopes reside and polytope $\text{PH}(\mathbf{Bi})$ is exactly the region containing APH distributions of algebraic degree $l-1$ or less with poles taken from $\{-\mu_1, -\mu_2, \dots, -\mu_l\}$.

Let the underlying CTMC of the standard representation of the maximum of n Erlang distributions $\text{Erl}(\lambda_i, k_i)$, for $1 \leq i \leq n$, be $\mathcal{M} = (\mathcal{S}, \mathbf{R}, \vec{\pi})$. From the structure of this representation, we know that each path $\sigma \in \text{Paths}(\mathcal{M})$ is of length $\sum_{i=1}^n k_i$, which is less than $l-1$. For each of the $(l-1)$ -dimensional affine subspaces, we show that there is a path $\sigma_{\text{out}} \in \text{Paths}(\mathcal{M})$ whose distribution resides outside it.

Take an arbitrary $(l-1)$ -dimensional affine subspace. There must be a total outgoing rate λ_h —and the corresponding distribution $q_{\{h\}}$ —whose multiplicity in the set generating the $(l-1)$ -dimensional affine subspace is $c(\mathcal{M}, \lambda_h) - 1$. Take a path $\sigma_{\text{out}} \in \text{Paths}(\mathcal{M})$ such that there are $c(\mathcal{M}, \lambda_h)$ states with total outgoing rate λ_h in the path. The existence of this path is guaranteed by Property 4.18. The distribution associated with this path resides outside the $(l-1)$ -dimensional affine subspace. \square

From Lemma 4.19 and Lemma 4.20, we can conclude that the size of the minimal representation of the maximum of n Erlang distributions grows exponentially in the number of involved Erlang distributions n .

The minimality proof in both Lemma 4.20 and Lemma 4.11 basically proceeds in a similar way, *i.e.*, that in each case, a smaller APH representation can be obtained, and then the size of this APH representation is shown to be equal to the algebraic degree of its APH distribution. Similar argument is also used in the case of the convolution of Erlang distributions (cf. Section 4.2). We can conclude then that any APH distribution resulted from the convolution, minimum, or maximum of Erlang distributions is triangular ideal. In Chapter 5 Lemma 5.9, we will show that the minimal representation of any triangular-ideal APH distribution can always be obtained by using Algorithm 3.13. Hence, the minimality is a property of the APH distributions resulted from the operations, instead of the specific representations or structures produced by the operations, *i.e.*, cross products and concatenations.

Furthermore, since each of the three operations on PH distributions is commutative and associative, the standard representations of the convolution, minimum, or maximum operations applied to more than two Erlang constituents does not have to be constructed in a single step prior to the minimization. Instead, repeated applications of the operation, each works on two constituents, can be interspersed with minimizations to obtain the final minimal representations. In this way, we can avoid having to build standard representations that grow exponentially fast. This is demonstrated in Example 4.21, and later in the first case study of Chapter 7.

Example 4.21. *Figure 4.10(a) depicts three Erlang distributions: $\text{Erl}(3, 2)$, $\text{Erl}(4, 2)$, and $\text{Erl}(5, 2)$. The standard representation of the maximum of the first two Erlang distributions—*i.e.*, $\max\{\text{Erl}(3, 2), \text{Erl}(4, 2)\}$ —is shown in Figure 4.10(b). The standard representation consists of eight transient states, two states with total outgoing rate 3, two states with total outgoing rate 4, and four states with total outgoing rate 7. The minimal ordered bidiagonal representation of this maximum is depicted in Figure 4.10(c), which is of size 7. Let this ordered bidiagonal representation be denoted by $(\vec{\beta}, \mathbf{Bi})$.*

Now, the standard representation of the $\max\{\text{PH}(\vec{\beta}, \mathbf{Bi}), \text{Erl}(5, 2)\}$ is depicted in Figure 4.10(d). The size of this representation is 23. From the discussion right before this example, we have $\max\{\text{Erl}(3, 2), \text{Erl}(4, 2), \text{Erl}(5, 2)\} = \max\{\text{PH}(\vec{\beta}, \mathbf{Bi}), \text{Erl}(5, 2)\}$, and, furthermore, the minimal ordered bidiagonal representations of the two APH distributions should be the same. This is indeed the case: the minimal ordered bidiagonal representation of the two APH distributions is depicted in Figure 4.10(e). This representation consists of only 19 transient states.

4.5 Conclusion

In this chapter, we introduced the notion of the core series, a refinement of the basic series. Using this notion, we can remove redundant multiplicities of states with the same total outgoing rate when building the ordered bidiagonal representation of a given APH representation. Like the idea of the basic series, the idea of the core series has no algorithmic value in general: this is because of the need to consider all possible paths in the representation to obtain either series, which is usually of exponential complexity. However, for some highly structured representations, like those produced by the minimum and maximum operations on ordered bidiagonal representations or Erlang distributions, the idea is valuable. We demonstrated this in this chapter, and we will show its further use again in Section 5.3.2.

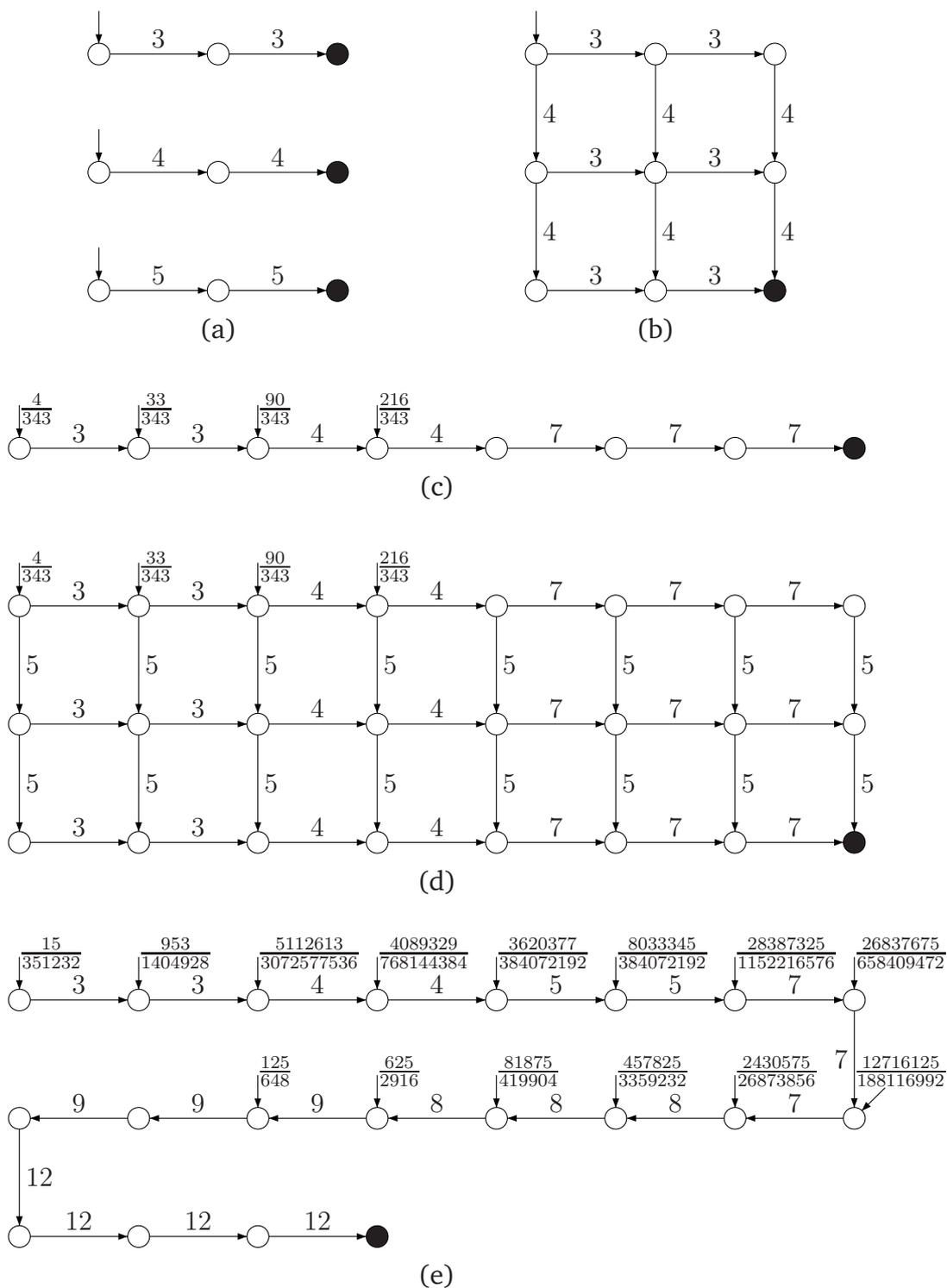


Figure 4.10: (a) Three Erlang representations, (b) The standard representation of the maximum of the first two Erlang distributions, (c) The minimal ordered bidiagonal representation of the maximum of the first two Erlang distributions, (d) The representation of the maximum of the third Erlang distribution with the maximum of the first two Erlang distributions, and (e) The minimal representation of the maximum.

We also investigated the structure and properties of the convolution, minimum, and maximum of Erlang distributions, and showed their minimal representations. In the cases of convolution and minimum operations, the precise number of states in their minimal representations can be known in advance for any arbitrary Erlang constituents. In the case of the maximum operation, on the other hand, the number of states in the minimal representation depends on the combinations of the rates of the involved Erlang constituents. We assumed these rates to be distinct from each other during our discussion to enable us to derive an upper bound for the number of states in the minimal representation. This distinctness assumption plays no role in the minimality of the resulting representation.

While the standard Markov chain representation for the minimum or maximum operation is basically the hypercube-structured cross product of the constituent Erlang distributions, we have shown how a smaller representation can be constructed, which in particular removes duplicate total outgoing rates wherever possible. Albeit smaller, we have shown that in the case of the maximum operation, in theory, it still grows exponentially (in the number of the constituent Erlang distributions), and that this is inevitable, since it is a minimal representation. The proofs use properties of the polytope induced by the minimal representation together with counting arguments.

This result sheds interesting light on the Kronecker representations for stochastic Petri nets (SPN) or other concurrent formalisms [PA91, BCDK00, HK01], where the maximum of such distributions is represented implicitly by the Kronecker sum of its constituent matrices. While computing the resulting matrix explicitly would lead to the very same exponential growth, keeping it implicit avoids this, since it allows storage linear in the constituent sizes. The price to pay however is that, in the analysis of such systems, the concrete matrix entries are computed as needed, which may be costly time-wise. With the results of this paper, it is definite that at least for Erlang distributions, there is no explicit way out of this, since the state space must grow exponentially.

In practice, however, the minimal representation can be significantly smaller than the standard representation produced by the maximum operation. For instance, the standard representation of the maximum of three Erlang distributions with distinct rates, each with phase 10 is of size 1330. Its minimal representation, on the other hand, is only of size 115. The case study in Section 7.1 will further demonstrate this significance.

The results obtained in this chapter can stand as a rough guideline for the results of operations on more complex representations than Erlang, such as hypoexponential or ordered bidiagonal representations. Based on the structure of the core series, we can infer that the size of the minimal representations of operations on these more complex representations cannot be smaller than the size of the minimal representations of operations on Erlang distributions when the constituents are of similar size.

We remark that both strong and weak-bisimulation-based lumpings have an effect on the standard representations of the minimum or maximum of Erlang distributions only when some of the constituent Erlang distributions have the same rates. Even in this case, however, bisimulation-based lumpings are not better than our reduction algorithm, for they cannot produce the minimal representations.

Chapter 5

The Use of APH Reduction

In Chapter 3, we have presented an efficient algorithm to reduce the size of APH representations. We have also analyzed the time complexity of the reduction algorithm, and provided an early perspective on its usage by a simple example. In this chapter, we put the algorithm in the context of contemporary known results. We delve more deeply in analyzing the areas in which the algorithm has the most potential. Seen from a broader perspective, our effort strives to move forward the current standing of the problem of finding the order of PH distributions.

Related Work The problem of finding the order of PH distributions has remained an open problem since the concept was first conceived. The order of a PH distribution describes the minimal number of states needed to represent it as an absorbing Markov chain. Some lower bounds of the order are known. The earliest known lower bound is the algebraic degree [Neu81, O’C90]: the order of a PH distribution cannot be less than its algebraic degree. Furthermore, the order of a PH distribution is shown to be no less than the inverse of its coefficient of variation in [AS87]. Another lower bound of the order, which depends on the largest real pole of the LST of the PH distribution and its complex poles, was provided in [O’C91].

Some researchers broadened our knowledge on the order of PH distributions by studying simpler subsets of PH distributions. An overview of APH distributions and the theory of their triangular order is presented in [O’C93]. The concept of the dual of PH representations and its use in determining whether a given PH representation is minimal is introduced in [CC93]. The authors of [CC96] and [CM03] established several conditions under which PH distributions are ideal. Recall that a PH distribution is ideal if and only if its order is equal to its algebraic degree. Most of these conditions have to be verified on the LST of the PH distributions. In [HZ06b], [HZ06a], and [HZ07a], the authors made significant progress in finding the order of APH distributions. They provided an algorithm for obtaining the ordered bidiagonal representation of any APH distribution. They also clarified some properties of ordered bidiagonal representations, and proposed a novel algorithm to reduce ordered bidiagonal representations to their minimal representations, and thereby solved the problem of finding the order of APH distributions.

Other researchers, such as the authors of [BHST03, BHT04], looked for PH distributions that have particular structures and forms that consequently result in some ease in analyzing the representations. In [MC99], the authors provided a method for finding canonical and sparse—*i.e.*, having small number of transitions—representations of PH

distributions. The sparse representations are in the form of mixtures of monocyclic Erlangs. A monocyclic Erlang is a modified Erlang representation with a single loop from the last to the first state. Although these representations may blow up the state space, they are of interest because of their sparseness. On the other hand, in [CM02], the authors showed that, in the set of all size n PH representations of some pre-specified structure, the set of all parameter values giving rise to PH distributions of algebraic degree less than n has measure zero. Hence, almost all PH representations are of size that is equal to their algebraic degree.

Contribution The contributions of the chapter are threefold. First, we clarify the use of the reduction algorithm: although it does not always produce minimal representations, when the given APH representation represents a triangular-ideal APH distribution, the algorithm can always reduce it to its minimal representation. Recall that an APH distribution is triangular ideal if and only if its triangular order is equal to its algebraic degree. Second, we investigate the effect of the convolution, minimum, and maximum operations on triangular-ideal APH distributions. We study the structure of the representations produced by the operations. For the minimum and maximum operations, we show that the algorithm benefits from the structure. Lastly, we show that given two triangular-ideal APH distributions, the three operations almost always produce a triangular-ideal APH distribution.

Structure The chapter is organized as follows: Section 5.1 lays out the reasons why our reduction algorithm does not always produce minimal APH representations. In Section 5.2, we study the case of triangular-ideal APH distributions. We review existing results, and explain the role of our reduction algorithm in this case. Section 5.3 describes an investigation into the effects of the convolution, minimum, and maximum operations on triangular-ideal APH distributions. In Section 5.4, we show that given two triangular-ideal APH distributions, the three operations almost always produce a triangular-ideal APH distribution. We summarize and conclude the chapter in Section 5.5.

5.1 Minimal and Non-Minimal Representations

In the example in Section 3.5, the reduction algorithm produces a minimal representation for the given APH representation. This can be verified by the fact that the size of the produced representation—depicted in Figure 3.12—is equal to its algebraic degree and hence the representation is minimal (cf. Theorem 2.26). The algebraic degree, in this case, is precisely the number of L -terms that are not the divisor of the numerator polynomial, which is equal to the number of states in the representation.

To arrive at a minimal representation is not always possible in general. To shed some light on this, consider the representations depicted in Figure 5.1.

Figure 5.1 shows two distinct APH representations of size 5 and 4, respectively. Both representations in the figure, however, have the same PH distribution; their distribution function is given by

$$F(t) = 1 - \frac{648}{1045}e^{-2t} + \frac{3}{5}e^{-16t} - \frac{1024}{1045}e^{-21t}, \quad t \in \mathbb{R}_{\geq 0}.$$

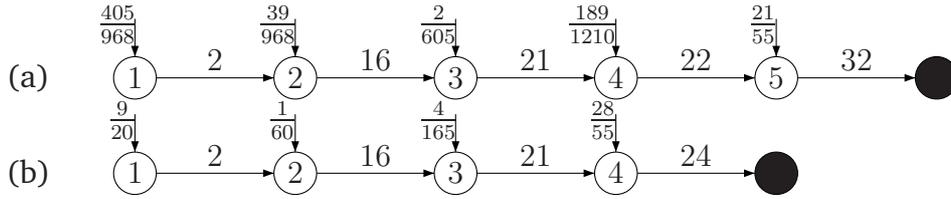


Figure 5.1: Non-Minimal and Minimal Representations

The LST of the PH distribution is

$$\tilde{f}(s) = \frac{672(s^2 + 16s + 55)}{55(s+2)(s+16)(s+21)}, \quad s \in \mathbb{R}_{\geq 0}. \quad (5.1)$$

The algebraic degree of the PH distribution is 3, and the poles of its LST are $s = -2$, $s = -16$, and $s = -21$.

If we insist on obtaining an ordered bidiagonal “representation” of the distribution by only using states with total outgoing rates 2, 16, and 21 (therefore, of size 3), we obtain a matrix-exponential distribution whose representation is depicted in Figure 5.2. Obviously, this is no longer a PH representation, but an ME representation.

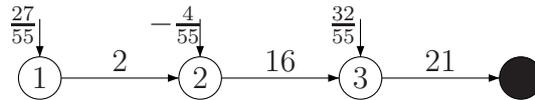


Figure 5.2: A Matrix-Exponential Representation of those in Figure 5.1

Now a question arises: what is the relationship between the poles of the LST of a particular APH distributions and each of its APH representations? The following lemma provides the answer.

Lemma 5.1. *If $-\lambda \in \mathbb{R}$ is a pole of the Laplace-Stieltjes transform of an acyclic phase-type distribution, then every acyclic phase-type representation of the distribution has at least one state with total outgoing rate λ .*

Proof. Every APH representation must have a unique ordered bidiagonal representation of the form $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$. The LST of the ordered bidiagonal representation expressed in L -terms is (from Equation (3.8))

$$\tilde{f}(s) = \frac{\vec{\beta}_1 + \vec{\beta}_2 L(\lambda_1) + \dots + \vec{\beta}_n L(\lambda_1)L(\lambda_2)\dots L(\lambda_{n-1})}{L(\lambda_1)L(\lambda_2)\dots L(\lambda_n)}. \quad (5.2)$$

Recall that an L -term $L(\lambda) = \frac{s+\lambda}{\lambda}$.

The LST of the APH distribution (in irreducible ratio form) can be obtained by removing all common factors from the numerator and denominator polynomials in Equation (5.2). Therefore, the poles of the LST of the distribution are among the poles of Equation (5.2), namely they are among the zeros of the L -terms in the denominator polynomial. But then each L -term $L(\lambda_i)$ in the denominator polynomial represents a state with total outgoing rate λ_i in $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$, and thus in the original APH representation. \square

Every APH representation contains states that represent poles of the LST of the associated APH distribution. Hence, by Lemma 5.1 and Equation (5.1), any APH representation of the PH distribution depicted in Figure 5.1 must contain states with total rates 2, 16, and 21. It is clear then that this PH distribution has no APH representation of size 3, because, as shown in Figure 5.2, the “representation” that consists only of the three states is no longer APH but ME representation. It follows that the representation depicted in Figure 5.1(b) is the smallest APH representation of the PH distribution. The PH distribution is then of triangular order 4. It is not clear, however, whether the order of the PH distribution is also 4: it may have a cyclic PH representation of size 3.

Algorithm 3.13 cannot reduce the size of the representation depicted in Figure 5.1(a), because none of its states is removable in the sense of Section 3.4.3 (or Algorithm 3.13). However, the representation depicted in Figure 5.1(b) is of smaller size. Therefore, the algorithm is not guaranteed to produce the smallest or minimal representations. The reason behind this deficiency lies in the fact that the algorithm is bound to the set of present total outgoing rates (as L -terms), while in reality the representation depends on the interplay of total outgoing rates and the initial probability distribution. These dependencies are in full generality difficult to detect, because we are then left with the problem of finding matches over a continuous domain of candidates, akin to the non-linearity of the problem encountered in [HZ07a].

Recall that HE and ZHANG in [HZ07a] proposed an algorithm for computing the minimal ordered bidiagonal representations of APH distributions. Initially, their algorithm transforms the given APH representation to the ordered bidiagonal representation that contains only states that represent the poles of the LST of the distribution. As shown in Figure 5.2, the resulting ordered bidiagonal representation is not necessarily of a PH distribution, but is certainly of an ME distribution. If the representation is not of a PH distribution, a new state is appended to the representation, and its total outgoing rate is determined by solving a system of non-linear equations. This is performed one by one until the obtained representation represents a PH distribution. The first such representation found is a minimal APH representation.

Given the representation depicted in Figure 5.1(a), for instance, the algorithm of HE and ZHANG can produce the minimal representation depicted in Figure 5.1(b). However, even this minimal representation is not unique. As reported by the authors in [HZ07a], valid total outgoing rates—namely those that correspond to valid initial probability vectors—of an appended state form an interval of real values.

5.2 When Order = Algebraic Degree

In this section, we review existing results that specify conditions under which a PH distribution is ideal. Most of the results apply to APH distributions. We show that our reduction algorithm always produces minimal representations when it is applied to triangular-ideal APH distributions.

5.2.1 Known Results

The first result is for APH distributions whose LSTs have numerator polynomials of degree zero or one.

Theorem 5.2 ([CC96]). *Let the Laplace-Stieltjes transform of a phase-type distribution be $\tilde{f}(s) = \frac{p(s)}{q(s)}$. If all of the following conditions hold*

1. $p(s)$ and $q(s)$ are co-prime polynomials,
2. $q(s)$ is of degree n and has n real roots, and
3. $p(s)$ is of degree less than or equal to one,

then the phase-type distribution has APH representations of size n .

It follows that all PH distributions satisfying the conditions are acyclic (cf. Theorem 2.28). Furthermore, the representations described are minimal and triangular minimal, since its algebraic degree is equal to its size (cf. Lemma 2.26). When such representations are transformed into their ordered bidiagonal representations, they start from the first or the first two states. Those that start from the first state cover all hypoexponential representations and hence also all Erlang representations.

Figure 5.3 depicts the associated Cox representation of the ordered bidiagonal representation that start from the first two states. Since $p(s)$ is of degree 1, it must be of the form $p(s) = \frac{s+\lambda}{\lambda}$, where $\lambda \in \mathbb{R}_+$ and $\lambda \neq \mu_i$, for $1 \leq i \leq n$, since otherwise $q(s)$ will be of degree $n - 1$. From the LST, we identify that $x = \frac{\mu_n}{\lambda}$, and therefore $\lambda > \mu_n$. In fact, this condition is necessary so that the LST is of a distribution function, namely that the inverse of the LST is positive for all $t \in \mathbb{R}_+$ [BHM87, CC96].

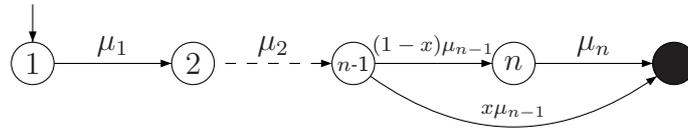


Figure 5.3: Cox Representation with Numerator Polynomial of Degree 1

The second result is for APH distributions whose LSTs have numerator polynomials of degree two.

Theorem 5.3 ([CC96]). *Let the Laplace-Stieltjes transform of a phase-type distribution be $\tilde{f}(s) = \frac{p(s)}{q(s)}$, where $p(s)$ and $q(s)$ are co-prime polynomials with real roots, and*

$$\begin{aligned}
 p(s) &= L(\lambda_1)L(\lambda_2), \quad \lambda_1 \geq \lambda_2 > 0, \text{ and} \\
 q(s) &= \prod_{i=1}^n L(\mu_i), \quad \mu_1 \geq \mu_2 \geq \dots \geq \mu_n > 0.
 \end{aligned}
 \tag{5.3}$$

The phase-type distribution has APH representations of size n if and only if $\lambda_2 > \mu_n$ and $(\lambda_1 + \lambda_2) \geq (\mu_{n-1} + \mu_n)$.

It follows that all PH distributions satisfying the conditions are acyclic, and the representations described must be minimal and triangular minimal. When such representations are transformed into their ordered bidiagonal representations, the first three states have nonzero initial probability. Figure 5.4 depicts the associated Cox representation of the ordered bidiagonal representation.

Equation (5.3) restricts all zeros of $p(s)$ to be real. Similar to the previous case, λ_1 and λ_2 must be positive and different from any of μ_i , for $1 \leq i \leq n$, since otherwise

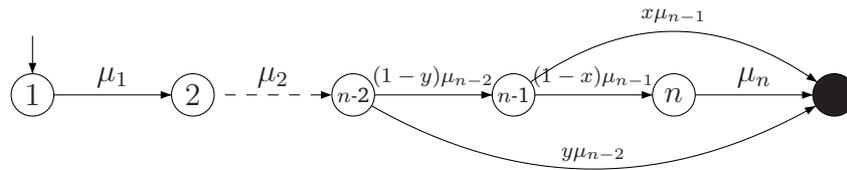


Figure 5.4: Cox Representation with Numerator Polynomial of Degree 2

$q(s)$ will be of smaller degree. The condition $\lambda_2 > \mu_n$ arises from the same reasoning as in the previous case. The condition $(\lambda_1 + \lambda_2) \geq (\mu_{n-1} + \mu_n)$ is necessary for the PH distribution to have a Cox representation of size n . From the LST, we can compute

$$x = \frac{\mu_n(\lambda_1 + \lambda_2 - \mu_{n-1}\mu_n)}{\lambda_1\lambda_2 - \mu_{n-1}\mu_n}, \quad \text{and} \quad y = \frac{\mu_{n-1}\mu_n}{\lambda_1\lambda_2}.$$

For the case where both zeros of $p(s)$ are complex (and hence one is the conjugate of the other), the corresponding APH distribution may or may not have an APH representation of size n . We refer back to Example 2.29. The numerator polynomial of the LST shown in Equation (2.22) has zeros $s = -\frac{11}{4} \pm i\frac{1}{4}\sqrt{7}$. The example showed that the APH distribution associated with this LST has a representation of size 3. On the other hand, consider the following LST

$$\tilde{f}(s) = \frac{\frac{1}{3}(s^2 + 2s + 3)}{L(1)L(2)L(3)L(5)}.$$

The numerator polynomial of the LST is of degree 2, and has zeros $s = -1 \pm i\sqrt{2}$. We will show in Section 5.3.1 that the PH distribution associated with this LST has no APH representation of size 4.

The third result is a generalization of the previous two, namely for APH distributions whose LSTs have numerator polynomials that are of any degree less than the degree of the denominator polynomials.

Theorem 5.4 ([CM03]). *Let the Laplace-Stieltjes transform of a phase-type distribution be $\tilde{f}(s) = \frac{p(s)}{q(s)}$, where $p(s)$ and $q(s)$ are co-prime polynomials with real roots, and*

$$p(s) = \prod_{i=1}^m L(\lambda_i), \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m > 0, \quad \text{and}$$

$$q(s) = \prod_{i=1}^n L(\mu_i), \quad \mu_1 \geq \mu_2 \geq \cdots \geq \mu_n > 0, \quad n > m.$$

If $\lambda_m \geq \mu_n, \lambda_{m-1} \geq \mu_{n-1}, \dots, \lambda_1 \geq \mu_{n-m+1}$ then the phase-type distribution has APH representations of size n .

Proof. Remark: We present a proof here because there are some ambiguities in the original proof provided in [CM03].

The LST of the PH distribution can be rewritten as

$$\tilde{f}(s) = \underbrace{\prod_{i=1}^{n-m} \frac{1}{L(\mu_i)}}_{(a)} \underbrace{\prod_{i=n-m+1}^n \frac{L(\lambda_{i-n+m})}{L(\mu_i)}}_{(b)}. \quad (5.4)$$

Each term of part (a) in Equation (5.4) corresponds to an exponential distribution with rate μ_i . Therefore, part (a) can be represented by a convolution of $n - m$ exponential distributions, producing a hypoexponential representation of size $n - m$. Each term of part (b), on the other hand, corresponds to the PH distribution of the representation depicted in Figure 5.5.

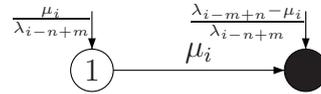


Figure 5.5: Representation of Each Term of Part (b) in Equation (5.4)

Part (b), then, can be represented by a convolution of m such representations, which produces a bidiagonal representation of size m . The final representation is formed by a convolution of the hypoexponential and the bidiagonal representations, and it is of size n . □

Example 5.5. Let the LST of a PH distribution be given by

$$\tilde{f}(s) = \frac{1}{L(3)} \frac{L(5)}{L(2)} \frac{L(4)}{L(1)}.$$

The representations of $\frac{1}{L(3)}$, $\frac{L(5)}{L(2)}$, and $\frac{L(4)}{L(1)}$ are depicted in Figure 5.6(a), (b), and (c), respectively. The convolution of the three representations is shown in Figure 5.6(d). The resulting representation is of size 3.

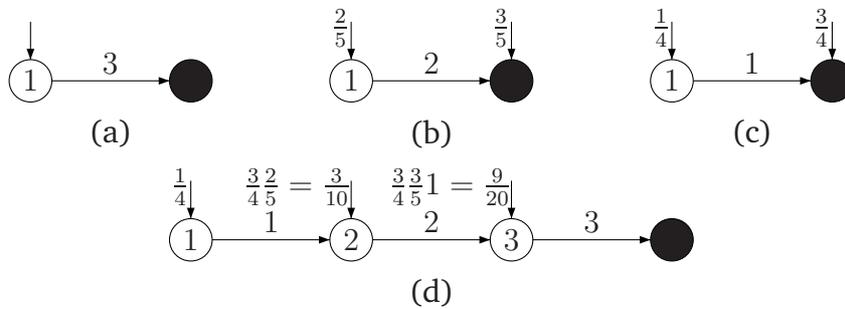


Figure 5.6: Representations Discussed in Example 5.5

A PH distribution satisfying the conditions in Theorem 5.4 must be acyclic, and the representation described is minimal and triangular minimal. In its ordered bidiagonal form, the first $m + 1$ states have nonzero initial probability. Note that the theorem only provides the necessary conditions for the PH distribution to be of (triangular) order n .

The fourth known result is described in the following theorem. It establishes a connection between the algebraic degree, the size, and the PH-simplicity of the PH-generators of a PH representation and of its dual.

Theorem 5.6 ([CC93]). Let $(\vec{\alpha}, \mathbf{A})$ be a phase-type representation of size n . The associated phase-type distribution has algebraic degree n if and only if the PH-generator of the representation and the PH-generator of its dual representation are both PH-simple.

The theorem applies not only to APH distributions. It follows from the theorem that if a PH-generator of a PH representation of a particular size n or the PH-generator of its dual is not PH-simple, then the algebraic degree of the associated PH distribution must be less than n . It does not follow, however, that the PH distribution has a representation of smaller size, for the representation of size n might already be minimal.

This theorem is especially advantageous for APH distributions. The ordered bidiagonal PH-generator of an APH distribution is always PH-simple (cf. Theorem 3.4). It remains to check whether the PH-generator of its Cox representation, *i.e.*, the dual of the ordered bidiagonal one, is also PH-simple to establish that the size of the ordered bidiagonal representation is equal to the algebraic degree of the APH distribution, and that therefore it is a minimal representation.

We have discussed four existing results in this section. The first three concern several conditions under which an APH distribution is triangular ideal and (therefore) ideal. These conditions are not easy and practical to check, for they work mostly on the LST domain and involve factorizations of polynomials. Nevertheless, the results are interesting in themselves, because they give us insight into the properties of APH distributions and representations. The fourth result provides the most general method to verify that the size of a PH representation is equal to the algebraic degree of its distribution.

5.2.2 Our Reduction Algorithm

The results described in the previous subsection establish several conditions under which an APH distribution is triangular ideal. We may encounter such an APH distribution, however, in a representation having strictly larger size than the order of its distribution. In this case, verifying the conditions will be difficult. Even more so if we wish to build a representation of the same size as the order of the distribution. Obtaining such minimal representation is of high interest in the field of stochastic modelling and analysis. In this section, we prove that, given an APH representation whose APH distribution is triangular ideal—no matter how large the size of the representation is—our reduction algorithm is certain to produce a minimal representation.

Definition 5.7. Let $\text{Red}(\vec{\alpha}, \mathbf{A})$ denote the reduced representation of acyclic phase-type representation $(\vec{\alpha}, \mathbf{A})$ obtained by applying Algorithm 3.13.

From the algorithm, $\text{Red}(\vec{\alpha}, \mathbf{A})$ is an ordered bidiagonal representation. We need the following lemma in our main proof.

Lemma 5.8. Let $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$ be an ordered bidiagonal representation. For an arbitrary $\lambda_x \geq \lambda_i$, $1 \leq i \leq n$, there is a unique sub-stochastic vector $\vec{\beta}'$ such that

$$\text{PH}(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)) = \text{PH}(\vec{\beta}', \mathbf{Bi}(\lambda_1, \dots, \lambda_i, \lambda_x, \lambda_{i+1}, \dots, \lambda_n)).$$

Proof. The LST of $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$ expressed in L -terms (from Equation (3.8)) is

$$\frac{\vec{\beta}_1 + \vec{\beta}_2 L(\lambda_1) + \dots + \vec{\beta}_n L(\lambda_1) L(\lambda_2) \cdots L(\lambda_{n-1})}{L(\lambda_1) L(\lambda_2) \cdots L(\lambda_n)},$$

which can be rewritten as

$$\frac{\vec{\beta}_1}{L(\lambda_1)} \frac{1}{L(\lambda_2) \cdots L(\lambda_n)} + \frac{\vec{\beta}_2 L(\lambda_1) + \dots + \vec{\beta}_n L(\lambda_1) L(\lambda_2) \cdots L(\lambda_{n-1})}{L(\lambda_1) L(\lambda_2) \cdots L(\lambda_n)}. \quad (5.5)$$

Recall the identity in Equation (3.2), which, when expressed in L -terms, can be written as

$$\frac{1}{L(\lambda)} = \frac{p}{L(\mu)} + \frac{(1-p)}{L(\lambda)L(\mu)},$$

where $\lambda \leq \mu$ and $p = \frac{\lambda}{\mu}$. Let $w = \frac{\lambda_1}{\lambda_x}$. Then, by using the identity, the first term of Equation (5.5) can be expressed in a different way, and Equation (5.5) then becomes

$$\left(\frac{\vec{\beta}_1(1-w)}{L(\lambda_1)L(\lambda_x)} + \frac{\vec{\beta}_1 w}{L(\lambda_x)} \right) \frac{1}{L(\lambda_2) \cdots L(\lambda_n)} + \frac{\vec{\beta}_2 + \cdots + \vec{\beta}_n L(\lambda_2) \cdots L(\lambda_{n-1})}{L(\lambda_2) \cdots L(\lambda_n)}. \quad (5.6)$$

We can construct an APH representation of Equation (5.6). A possible representation is depicted in Figure 5.7.

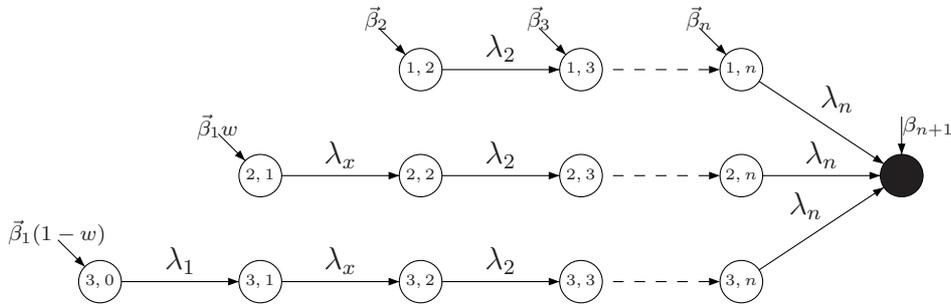


Figure 5.7: A Possible Representation of the LST in Equation (5.6)

We can now build the core series of the obtained representation by following the procedure described in Definition 4.2. Consider the multisets of the total outgoing rates in the first branch (which starts in state $(1, 2)$) and the second branch (which starts in state $(2, 1)$) in Figure 5.7. Each of these multisets is a subset of the the multiset of the total outgoing rates in the third branch (which starts in state $(3, 0)$). Therefore, the longest core series is formed by ordering the total outgoing rates in this third branch ascendingly, and it is of size (length) $n + 1$.

By using Lemma 4.5 on the resulting core series, we obtain an ordered bidiagonal representation $(\vec{\beta}', \mathbf{Bi}(\lambda_1, \dots, \lambda_i, \lambda_x, \lambda_{i+1}, \dots, \lambda_n))$ of size $n + 1$ for some sub-stochastic vector $\vec{\beta}'$. Since throughout the procedure the LST remains the same, the two ordered bidiagonal representations represent the same PH distribution. Now, PH-generator $\mathbf{Bi}(\lambda_1, \dots, \lambda_i, \lambda_x, \lambda_{i+1}, \dots, \lambda_n)$ is PH-simple (cf. Theorem 3.4). By Definition 2.36, the obtained vector $\vec{\beta}'$ is unique for this ordered bidiagonal representation, in the sense that this is the only sub-stochastic vector such that

$$\text{PH}(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n)) = \text{PH}(\vec{\beta}', \mathbf{Bi}(\lambda_1, \dots, \lambda_i, \lambda_x, \lambda_{i+1}, \dots, \lambda_n))$$

holds. □

We can then conclude that there is a unique way of incorporating a proper new state with a particular total outgoing rate to an existing ordered bidiagonal representation. Furthermore, since the ordered bidiagonal representation is a canonical form,

for the particular sets of total outgoing rates the obtained ordered bidiagonal representation is unique. Note that λ_x must not be less than λ_1 , since, in that case, the resulting representation may be of a different distribution, because $-\lambda_1$ is the largest pole of the LST of the distribution. According to Theorem 2.20, a PH distribution is characterized by the unique largest real pole of its LST.

Lemma 5.9. *Let $(\vec{\alpha}, \mathbf{A})$ be an acyclic phase-type representation. If $\text{PH}(\vec{\alpha}, \mathbf{A})$ is triangular ideal, then the size of representation $\text{Red}(\vec{\alpha}, \mathbf{A})$ is equal to the triangular order of $\text{PH}(\vec{\alpha}, \mathbf{A})$.*

Proof. Let m be the dimension of PH-generator \mathbf{A} , and let the ordered bidiagonal representation of $(\vec{\alpha}, \mathbf{A})$ be $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_m))$. Let n be the algebraic degree of $\text{PH}(\vec{\alpha}, \mathbf{A})$. Since n is also the triangular order of the distribution, $\text{PH}(\vec{\alpha}, \mathbf{A})$ must have at least one acyclic representation of size n . Let $(\vec{\gamma}, \mathbf{Bi}(\mu_1, \mu_2, \dots, \mu_n))$ be the ordered bidiagonal representation of this acyclic representation.

Let $\mathcal{P} = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$ and $\mathcal{Q} = \{\mu_1, \mu_2, \dots, \mu_n\}$ be the multisets of the total outgoing rates of states of ordered bidiagonal representations $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_m))$ and $(\vec{\gamma}, \mathbf{Bi}(\mu_1, \mu_2, \dots, \mu_n))$, respectively. Based on Lemma 5.1, we have $\mathcal{Q} \subseteq \mathcal{P}$.

For each $\lambda \in \mathcal{P} - \mathcal{Q}$, according to Lemma 5.8, a state with total outgoing rate λ can be incorporated into $(\vec{\gamma}, \mathbf{Bi}(\mu_1, \mu_2, \dots, \mu_n))$, resulting in a new ordered bidiagonal representation, say $(\vec{\gamma}', \mathbf{Bi})$, of size $n + 1$ having the same APH distribution, where $\vec{\gamma}'$ is unique for the PH-generator \mathbf{Bi} . It follows that all $\lambda \in \mathcal{P} - \mathcal{Q}$ can be incorporated into $(\vec{\gamma}, \mathbf{Bi}(\mu_1, \mu_2, \dots, \mu_n))$ in any arbitrary order, and by the uniqueness of the resulting initial probability vectors, we must obtain $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_m))$.

Therefore, each state associated with an L -term $L(\lambda)$, where $\lambda \in \mathcal{P} - \mathcal{Q}$, is removable from $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_m))$, namely (1) both the numerator and denominator polynomials of the LST of $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_m))$ expressed in L -term contain $L(\lambda)$; and (2) removing $L(\lambda)$ from both polynomials results in a sub-stochastic initial probability vector. Furthermore, the removal of all such states can be performed in any arbitrary order.

Algorithm 3.13 eliminates all removable states in a particular order: from the state with the smallest total outgoing rate to the state with the largest total outgoing rate. (For a particular L -term $L(\lambda)$, the procedure described in Lemma 5.8 reverses the procedure in line 7 of the algorithm). From the uniqueness of the initial probability vector for a particular ordered bidiagonal representation, eliminating all removable states must result in $(\vec{\gamma}, \mathbf{Bi}(\mu_1, \mu_2, \dots, \mu_n))$. Therefore, $\text{Red}(\vec{\alpha}, \mathbf{A}) = (\vec{\gamma}, \mathbf{Bi}(\mu_1, \mu_2, \dots, \mu_n))$, and the size of $\text{Red}(\vec{\alpha}, \mathbf{A})$ is equal to the triangular order of $\text{PH}(\vec{\alpha}, \mathbf{A})$. \square

In conclusion, Lemma 5.9 establishes that applying Algorithm 3.13 to any APH representation whose APH distribution is triangular ideal is certain to result in a minimal representation. The algorithm can also be applied to APH representations whose APH distribution is not triangular ideal, although in this case it is not guaranteed to result in a minimal representation.

In the following, we summarize the inner mechanism and the effects of the reduction algorithm. First, we clarify some notions. For an ordered bidiagonal representation $(\vec{\beta}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_n))$, a *common factor* is an L -term that exists both in the numerator and denominator polynomials of the LST of the ordered bidiagonal representation when it is expressed in L -terms. A common factor is removable if removing the fac-

tor results in a valid initial probability distribution. Conversely, a common factor is *irremovable* if removing the factor results in a non-valid initial probability distribution.

The reduction algorithm removes all unnecessary common factors from the numerator and denominator polynomials of (the LST of the form of) Equation (3.8). These removable common factors must have real zeros, since all factors of the denominator polynomial have real zeros. At every step of the removals, the resulting form of the LST has a straightforward associated APH representation. Once the reduction algorithm finishes, we obtain a certain expression of the LST and its associated representation. We distinguish the following possible results:

- (1) If the numerator and denominator polynomials have no common factors, then the associated representation is minimal, because the triangular order (and therefore the order) of its distribution is equal to its algebraic degree, *i.e.*, its distribution is (triangular) ideal.
- (2) From result (1), it follows that if the numerator polynomial only has complex zeros, then the associated representation is minimal, because the (triangular) order of its distribution is equal to its algebraic degree, *i.e.*, its distribution is (triangular) ideal.
- (3) If the numerator and denominator polynomials have a single irremovable common factor, then the associated representation is triangular minimal, and the triangular order of its distribution is not equal to its algebraic degree, *i.e.*, its distribution is not triangular ideal.
- (4) If the numerator and denominator polynomials have more than one irremovable common factor, then the associated representation may or may not be triangular minimal, and the triangular order of its distribution is certainly not equal to its algebraic degree, *i.e.*, its distribution is not triangular ideal.

Results (1) and (2) are straightforward.

For result (3), assume that the algebraic degree of the distribution of the ordered bidiagonal representation is n . The LST of the ordered bidiagonal representation once all of its removable common factors are removed is then of the form

$$\frac{p(s)}{q(s)} = \frac{L(\lambda)}{L(\lambda)} \frac{p'(s)}{\prod_{i=1}^n L(\lambda_i)}.$$

Since the common factor $\frac{L(\lambda)}{L(\lambda)}$ is irremovable, the LST fragment $\frac{p'(s)}{\prod_{i=1}^n L(\lambda_i)}$ has no ordered bidiagonal representation of size n , namely there is no sub-stochastic vector $\vec{\beta}$ of dimension n such that

$$p'(s) = \vec{\beta}_1 + \vec{\beta}_2 L(\lambda_1) + \cdots + \vec{\beta}_n L(\lambda_1) L(\lambda_2) \cdots L(\lambda_{n-1}).$$

Hence the PH distribution of the ordered bidiagonal representation must be of triangular order more than n . But the ordered bidiagonal representation is itself of size $n + 1$. Therefore, the triangular order of the PH distribution is $n + 1$, and the ordered bidiagonal representation is indeed triangular minimal. As an example, assume that given an APH representation, Algorithm 3.13 reduces it to the representation depicted in Figure 5.1(b). This representation has one irremovable common factor, namely $L(24)$, and therefore it is triangular minimal.

As for result (4), we may be given another APH representation, which Algorithm 3.13 then reduces to the representation depicted in Figure 5.1(a). This representation has more than one irremovable common factor, namely $L(22)$ and $L(32)$. Since, as we have shown, the associated distribution has a representation of size 4, this representation is not triangular minimal.

5.3 Operations on APH Representations

Let $(\vec{\alpha}', \mathbf{A})$ and $(\vec{\beta}', \mathbf{B})$ be two given APH representations whose APH distributions are triangular ideal. Further, let $(\vec{\alpha}, \mathbf{Bi}(\lambda_1, \dots, \lambda_m))$ and $(\vec{\beta}, \mathbf{Bi}(\mu_1, \dots, \mu_n))$ be the minimal ordered bidiagonal representations of the given APH representations, respectively. The ordered bidiagonal representations are obtained by applying Algorithm 3.13 to the original APH representations.

In this section, we study the effects of the convolution, minimum, and maximum operations on APH representations whose APH distributions are triangular ideal. Specifically, we (1) investigate the structure—namely the state space and the paths—of the APH representation produced by each operation; and (2) find out whether the produced APH distribution is also triangular ideal.

Definition 5.10. *An operation on acyclic phase-type representations is triangular-ideal-preserving if and only if, given acyclic phase-type representations whose phase-type distributions are triangular ideal, the operation produces a representation whose distribution is also triangular ideal.*

Hence, goal (2) amounts to showing that convolution, minimum, and maximum operations are triangular-ideal-preserving. A triangular-ideal-preserving operation is certainly desirable: if we confine ourselves to working only with APH representations whose distributions are triangular ideal (and thus whose minimal representations we can always obtain, cf. Lemma 5.9), then the operation is guaranteed to produce APH representations whose minimal representations are known and can also be obtained through the application of Algorithm 3.13.

5.3.1 Convolution Operation

Let $(\vec{\delta}, \mathbf{D})$ be the convolution of $(\vec{\alpha}, \mathbf{Bi}(\lambda_1, \dots, \lambda_m))$ and $(\vec{\beta}, \mathbf{Bi}(\mu_1, \dots, \mu_n))$, namely $(\vec{\delta}, \mathbf{D}) = \text{con}((\vec{\alpha}, \mathbf{Bi}(\lambda_1, \dots, \lambda_m)), (\vec{\beta}, \mathbf{Bi}(\mu_1, \dots, \mu_n)))$. Figure 5.8 depicts two possible representations of $(\vec{\delta}, \mathbf{D})$.

Using Definition 4.2 to construct the core series of $(\vec{\delta}, \mathbf{D})$, we can conclude that at least $m+n$ states is required in any possible representation of $\text{PH}(\vec{\delta}, \mathbf{D})$, no matter how many states with common total outgoing rates both representations $(\vec{\alpha}, \mathbf{Bi}(\lambda_1, \dots, \lambda_m))$ and $(\vec{\beta}, \mathbf{Bi}(\mu_1, \dots, \mu_n))$ share.

To determine whether the convolution operation is triangular-ideal-preserving, let the irreducible LSTs of $(\vec{\alpha}, \mathbf{Bi}(\lambda_1, \dots, \lambda_m))$ and $(\vec{\beta}, \mathbf{Bi}(\mu_1, \dots, \mu_n))$ be

$$\frac{p_1(s)}{q_1(s)} = \frac{\vec{\alpha}_1 + \vec{\alpha}_2 L(\lambda_1) + \dots + \vec{\alpha}_m L(\lambda_1) L(\lambda_2) \cdots L(\lambda_{m-1})}{L(\lambda_1) L(\lambda_2) \cdots L(\lambda_m)}$$

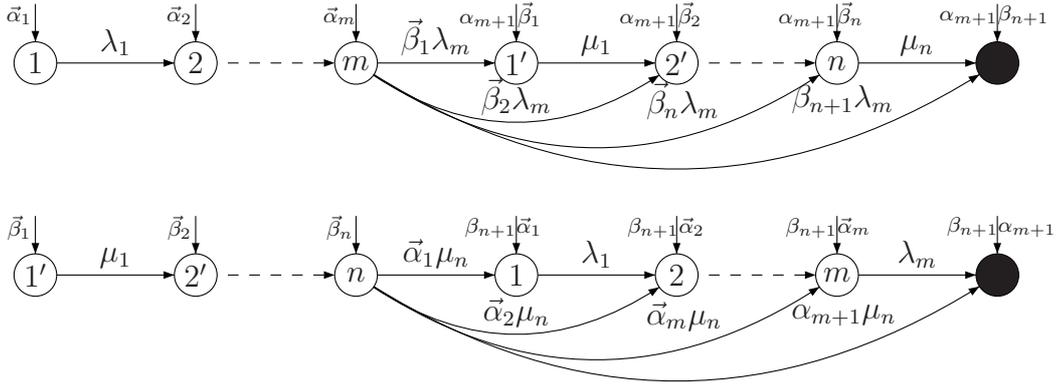


Figure 5.8: Two Representations of the Convolution of Two Ordered Bidiagonal Representations $(\vec{\alpha}, \text{Bi}(\lambda_1, \lambda_2, \dots, \lambda_m))$ and $(\vec{\beta}, \text{Bi}(\mu_1, \mu_2, \dots, \mu_n))$

and respectively

$$\frac{p_2(s)}{q_2(s)} = \frac{\vec{\beta}_1 + \vec{\beta}_2 L(\mu_1) + \dots + \vec{\beta}_n L(\mu_1) L(\mu_2) \cdots L(\mu_{n-1})}{L(\mu_1) L(\mu_2) \cdots L(\mu_n)}.$$

Now the LST of the convolution of the two representations is

$$\frac{p(s)}{q(s)} = \frac{p_1(s) p_2(s)}{q_1(s) q_2(s)}. \quad (5.7)$$

The LST in Equation (5.7) is reducible only if at least one of $\frac{p_1(s)}{q_2(s)}$ and $\frac{p_2(s)}{q_1(s)}$ is reducible. Without loss of generality, assume that $\frac{p_1(s)}{q_2(s)}$ is reducible. In this case $p_1(s)$ is divisible by some $L(\mu_i)$, for $1 \leq i \leq n$. Letting $p_1(s) = p'_1(s) L(\mu_i)$, Equation (5.7) can be written

$$\begin{aligned} \frac{p'_1(s)}{q_1(s)} \frac{\vec{\beta}_1 + \vec{\beta}_2 L(\mu_1) + \dots + \vec{\beta}_i L(\mu_1) \cdots L(\mu_{i-1})}{L(\mu_1) \cdots L(\mu_{i-1}) L(\mu_{i+1}) \cdots L(\mu_n)} \\ + \frac{p_1(s)}{q_1(s)} \frac{\vec{\beta}_{i+1} + \vec{\beta}_{i+2} L(\mu_{i+1}) + \dots + \vec{\beta}_n L(\mu_{i+1}) \cdots L(\mu_{n-1})}{L(\mu_{i+1}) \cdots L(\mu_n)}. \end{aligned}$$

The second term in the previous equation can be represented by a representation of size less than $m + n$, namely exactly $m + n - i$. If the first term can be represented by a representation of size $m + n - 1$, then so can the whole convolution, because both terms contain similar structure of state space. The first term can be represented by a representation of size $m + n - 1$ only if the APH distribution associated with the LST $\frac{p'_1(s)}{q_1(s)}$ is of triangular order m , and has an APH representation of size m . This is because the second factor (multiplicand) in the first term of the equation can easily be shown to have an APH representation of size $n - 1$.

The distribution associated with the LST $\frac{p'_1(s)}{q_1(s)}$ is of algebraic degree m . It remains to show that the distribution always possesses an APH representation of size m . This problem can be expressed in a slightly different way. Let \mathcal{P} be a set of polynomials

$$\mathcal{P} = \left\{ 1, \left(\frac{s + \lambda_1}{\lambda_1} \right), \left(\frac{s + \lambda_1}{\lambda_1} \right) \left(\frac{s + \lambda_2}{\lambda_2} \right), \dots, \left(\frac{s + \lambda_1}{\lambda_1} \right) \cdots \left(\frac{s + \lambda_n}{\lambda_n} \right) \right\},$$

where each of $\lambda_i > 0$, for $1 \leq i \leq n$, and $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. Let \mathcal{C} be the convex hull of \mathcal{P} . Take an arbitrary $p(s) \in \mathcal{C}$ such that

$$p(s) = p'(s) \left(\frac{s + \mu}{\mu} \right),$$

where $\left(\frac{s + \mu}{\mu} \right)$ is a real factor of $p(s)$. We have to determine whether $p'(s) \in \mathcal{C}$ for all $p(s) \in \mathcal{C}$.

For $n = 1, 2$, this holds. But it does not hold for $n \geq 3$, which will be demonstrated by using the APH representation depicted in Figure 5.9. The LST of the distribution associated with the representation is

$$\tilde{f}(s) = \frac{\frac{1}{12}(s^3 + 6s^2 + 11s + 12)}{L(1)L(2)L(3)L(5)} = \frac{\frac{1}{3}(s^2 + 2s + 3)L(4)}{L(1)L(2)L(3)L(5)}. \tag{5.8}$$

The numerator polynomial of Equation (5.8) is a convex combination of polynomials $\{1, L(1), L(1)L(2), L(1)L(2)L(3)\}$, and it is divisible by $L(4)$. However, the polynomial $\frac{1}{3}(s^2 + 2s + 3)$ is not a convex combination of polynomials $\{1, L(1), L(1)L(2)\}$. Therefore, the distribution associated with LST

$$\tilde{f}'(s) = \frac{\frac{1}{3}(s^2 + 2s + 3)}{L(1)L(2)L(3)L(5)},$$

has no APH representation of size 4.

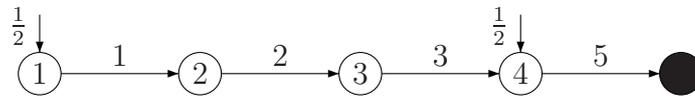


Figure 5.9: A Representation Associated with the LST in Equation (5.8)

Example 5.11. Let $(\vec{\alpha}, \mathbf{Bi}(1, 2, 3, 5))$ be the ordered bidiagonal representation depicted in Figure 5.9. Furthermore, let $(\vec{\gamma} = [\frac{1}{2}, \frac{1}{2}], \mathbf{Bi}(1, 4))$ be another ordered bidiagonal representation. The size of both representations is equal to their respective algebraic degree, hence their APH distributions are triangular ideal.

The ordered bidiagonal representation of the convolution of the two representations is depicted in Figure 5.10. The resulting representation is of size 6. However, the algebraic degree of the distribution associated with this representation is 5. Furthermore, this distribution has no representation of size 5. Therefore, the APH distribution is of triangular order 6 and is not triangular ideal.

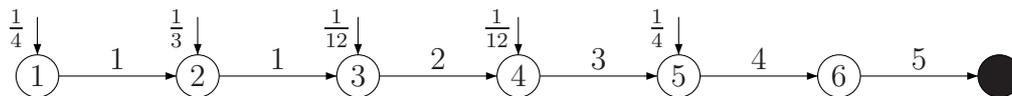


Figure 5.10: A Representation Produced by a Convolution Operation

The representation depicted in Figure 5.10 is triangular minimal, even though its APH distribution is not triangular ideal.

We can conclude that a convolution operation on two APH representations whose APH distributions are triangular ideal does not always produce an APH representation whose distribution is triangular ideal. Therefore, convolution operations are not triangular-ideal-preserving.

5.3.2 Minimum and Maximum Operations

Minimum Let $(\vec{\delta}, \mathbf{D}) = \min((\vec{\alpha}, \mathbf{Bi}(\lambda_1, \dots, \lambda_m)), (\vec{\beta}, \mathbf{Bi}(\mu_1, \dots, \mu_n)))$. Figure 5.11 depicts a possible representation of the minimum of the two ordered bidiagonal representations.

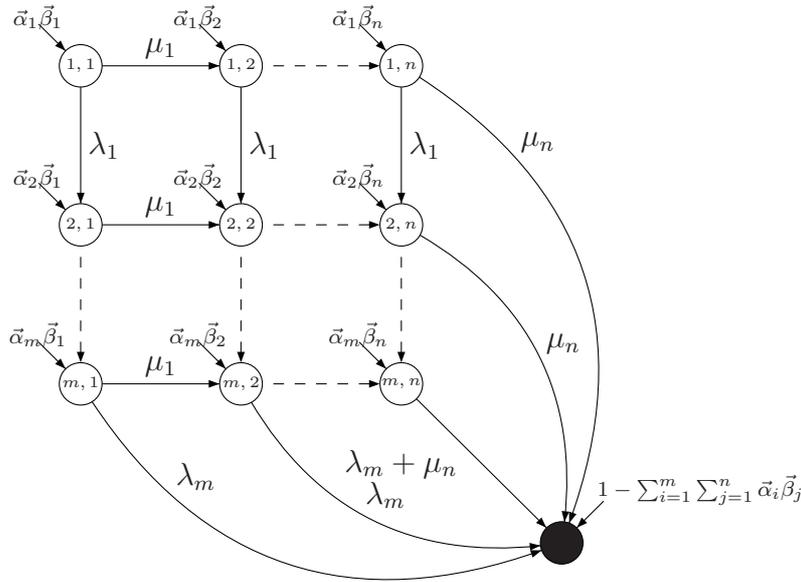


Figure 5.11: A Representation of the Minimum of Two Ordered Bidiagonal Representations $(\vec{\alpha}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_m))$ and $(\vec{\beta}, \mathbf{Bi}(\mu_1, \mu_2, \dots, \mu_n))$

Constructing the core series of $(\vec{\delta}, \mathbf{D})$, the number of states required to represent $\text{PH}(\vec{\delta}, \mathbf{D})$ is ranging from $m + n - 1$, namely for the case when $(\vec{\delta}, \mathbf{D})$ is the minimum of two Erlang distributions, to mn when all states in both representations $(\vec{\alpha}, \mathbf{Bi}(\lambda_1, \dots, \lambda_m))$ and $(\vec{\beta}, \mathbf{Bi}(\mu_1, \dots, \mu_n))$ have distinct total outgoing rates, and all their possible sums are also distinct.

From the structure of $(\vec{\delta}, \mathbf{D})$, we can infer that the number of distinct states with similar total outgoing rates in each ordered bidiagonal representation involved in the operation influences the number of states required in the minimal representation of $\text{PH}(\vec{\delta}, \mathbf{D})$. The more we have such states, the fewer states are needed in the minimal representation.

Maximum Let $(\vec{\delta}, \mathbf{D}) = \max((\vec{\alpha}, \mathbf{Bi}(\lambda_1, \dots, \lambda_m)), (\vec{\beta}, \mathbf{Bi}(\mu_1, \dots, \mu_n)))$. Figure 5.12 depicts a possible representation of the maximum of the two ordered bidiagonal representations.

We can construct the core series of $(\vec{\delta}, \mathbf{D})$ and observe that the number of states required to represent $\text{PH}(\vec{\delta}, \mathbf{D})$ is ranging from $m + n + (m + n - 1)$, namely for the

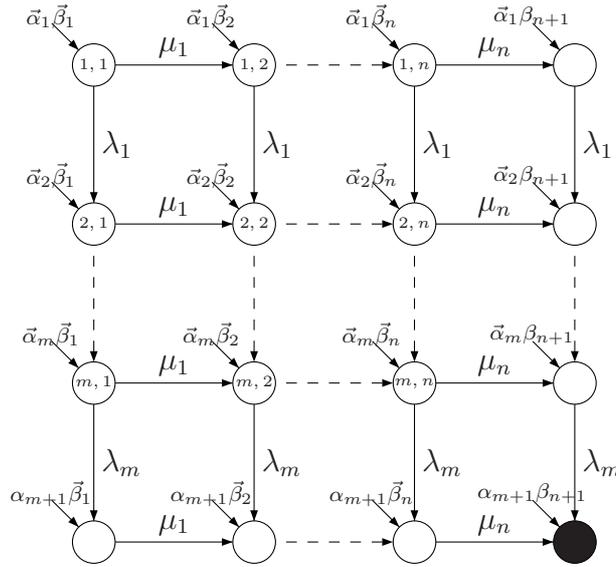


Figure 5.12: A Representation of the Maximum of Two Ordered Bidiagonal Representations $(\vec{\alpha}, \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_m))$ and $(\vec{\beta}, \mathbf{Bi}(\mu_1, \mu_2, \dots, \mu_n))$

case when $(\vec{\delta}, \mathbf{D})$ is the maximum of two Erlang distributions, to $m + n + (mn)$, namely when all states in both $(\vec{\alpha}, \mathbf{Bi}(\lambda_1, \dots, \lambda_m))$ and $(\vec{\beta}, \mathbf{Bi}(\mu_1, \dots, \mu_n))$ have distinct total outgoing rates, and all their possible sums are also distinct.

Similar to the case of the minimum operation, we can infer that the number of distinct states with similar total outgoing rates in each ordered bidiagonal representation involved in the maximum operation influences the number of states required in the minimal representation of $\text{PH}(\vec{\delta}, \mathbf{D})$.

Triangular-Ideal Preservation In the previous subsection, we showed that the convolution operation is not triangular-ideal-preserving. For the minimum and maximum operations, the situation is not as clear. We close this section by proposing the following conjecture. Although we have no strong indication as to the validity of the conjecture, we find that counterexamples are hard to find.

Conjecture 5.12. *The minimum and maximum operations are triangular-ideal-preserving.*

Algorithmic Improvement We mentioned in our characterizations of the structure of $(\vec{\delta}, \mathbf{D})$ produced by the minimum and maximum operations that the number of states with similar total outgoing rates in each ordered bidiagonal representation involved in the operation influences the number of states required in the minimal representation of $\text{PH}(\vec{\delta}, \mathbf{D})$. We will clarify and make precise this assertion in the rest of the section.

For $k \leq m$ and $l \leq n$, assume that k states of $(\vec{\alpha}, \mathbf{Bi}(\lambda_1, \dots, \lambda_m))$ have a common total outgoing rate λ , and, similarly, l states of $(\vec{\beta}, \mathbf{Bi}(\mu_1, \dots, \mu_n))$ have a common total outgoing rate μ . Since the minimum and maximum operations are basically cross-product operations, somewhere in the underlying Markov chain of $(\vec{\delta}, \mathbf{D})$, we will find a partial chain like that depicted in Figure 5.13. The partial chain forms the cross product of the k λ -states and the l μ -states.

Now, applying the idea of the core series only to the partial chain, we can conclude that we need only $k + l - 1$ $(\lambda + \mu)$ -states—instead of as many as kl —to represent the partial chain. Moreover, the same is true for $(\vec{\delta}, \mathbf{D})$: assuming these are the only $(\lambda + \mu)$ -states in the representation, any path in it traverses at most $k + l - 1$ $(\lambda + \mu)$ -states; the paths that traverse $k + l - 1$ $(\lambda + \mu)$ -states are those that pass through states $(1, 1)$ and (k, l) in the figure. Therefore, only $k + l - 1$ $(\lambda + \mu)$ -states are needed in the representation, regardless of the initial probabilities of the partial chain.

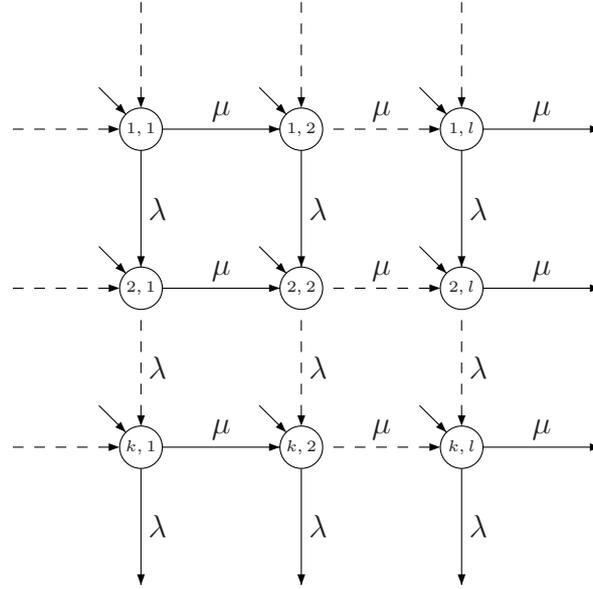


Figure 5.13: The Cross-Product between k λ -States and l μ -States

The same reasoning can be applied to all other pairs of total outgoing rates. Assume that there are m' and n' distinct total outgoing rates in $(\vec{\alpha}, \mathbf{Bi}(\lambda_1, \dots, \lambda_m))$ and $(\vec{\beta}, \mathbf{Bi}(\mu_1, \dots, \mu_n))$, respectively. Then there are $m'n'$ such pairs. The number of states that can be removed from the cross product of such a pair increases as the number of states in each member of the pair increases.

Further, assume that the number of states required in the representation $(\vec{\delta}, \mathbf{D})$, once all multiplicities of total outgoing rates in the cross products of all pairs are removed, is p . Then p is no less than the number of states in the core series of $(\vec{\delta}, \mathbf{D})$, since we assumed that $(\lambda + \mu)$ -states are confined to the cross product of λ -states and μ -states. We can build an ordered bidiagonal representation $\mathbf{Bi}(\nu_1, \dots, \nu_p)$ from these p states (total outgoing rates). Now, we can use the spectral polynomial algorithm (SPA) to transform $(\vec{\delta}, \mathbf{D})$ into $(\vec{\kappa}, \mathbf{Bi}(\nu_1, \dots, \nu_p))$ by solving the system of equations (from Equation (3.5))

$$\mathbf{P}(*, i) = \left(\prod_{j=i}^{p-1} \frac{1}{\nu_j} (\mathbf{D} + \nu_{j+1} \mathbf{I}) \right) \mathbf{P}(*, p), \quad 1 \leq i \leq p - 1, \quad (5.9)$$

where \mathbf{P} is a matrix with unit row-sums (*i.e.*, $\mathbf{P}\vec{e} = \vec{e}$), and $\vec{\kappa} = \vec{\delta}\mathbf{P}$. By Lemma 4.5 and Lemma 5.8, $\mathbf{Bi}(\nu_1, \dots, \nu_p)$ PH-majorizes \mathbf{D} , and therefore the sub-stochasticity of vector $\vec{\kappa}$ is guaranteed. Once ordered bidiagonal representation $(\vec{\kappa}, \mathbf{Bi}(\nu_1, \dots, \nu_p))$ is obtained, Algorithm 3.13 can be used to further reduce the size of the representation.

The pre-processing—namely the removal of the multiplicities in the cross products of all total outgoing rate pairs—we just introduced is advantageous for both SPA and the reduction algorithm. As we will show in several case studies in Chapter 7, SPA consumes most of the computation time of the reduction procedure. From Equation (5.9), it is evident that the number of iterations in SPA can be significantly reduced if many states with similar total outgoing rates exist in each ordered bidiagonal representation used in the minimum or maximum operations.

Furthermore, since the number of states in $(\vec{\kappa}, \mathbf{Bi}(\nu_1, \dots, \nu_p))$ is smaller than in $(\vec{\delta}, \mathbf{D})$, the number of states that have to be checked—whether removable or not—in Algorithm 3.13 also decreases. There can still be, however, many states to remove, namely those with total outgoing rates whose multiplicities in the core series of $(\vec{\delta}, \mathbf{D})$ cannot be detected by the pre-processing. In the next section, we will show that given two ordered bidiagonal representations whose APH distributions are triangular ideal, the number of states in the minimal representation of the minimum or the maximum of the two ordered bidiagonal representations will almost always be equal to the number of states in the core series of $(\vec{\delta}, \mathbf{D})$.

5.4 Almost Surely Minimal

It is shown in [CM02] that in the set of all size n PH representations of some pre-specified structure, the set of all parameter values giving rise to PH distributions of algebraic degree less than n has measure zero. Stated differently, PH distributions of algebraic degree n are *almost everywhere* in the set of all size n PH representations of some pre-specified structure. Recall that a property holds almost everywhere if the set of points the property fails has measure zero. Therefore a PH representation whose size is greater than its algebraic degree arise not from the structure of the representation, but rather from the particular parameter values of the representations [Fac03].

In the following we will prove a somewhat stronger but more restricted result.

Lemma 5.13. *Let $\mathbf{Bi}(\lambda_1, \dots, \lambda_n)$ be an ordered bidiagonal PH-generator. In the polytope $\text{PH}(\mathbf{Bi}(\lambda_1, \dots, \lambda_n))$, the set of all $\text{PH}(\vec{\alpha}, \mathbf{Bi}(\lambda_1, \dots, \lambda_n))$, where $\vec{\alpha} \in \mathbb{R}_{\geq 0}^n$ and $\vec{\alpha}\vec{e} \leq 1$, whose algebraic degree is less than n has measure zero.*

Proof. Consider the polytope of PH-generator $\mathbf{Bi}(\lambda_1, \dots, \lambda_n)$. Since $\mathbf{Bi}(\lambda_1, \dots, \lambda_n)$ is an ordered bidiagonal PH-generator, it is PH-simple (cf. Theorem 3.4). Therefore, the polytope is n -dimensional, *i.e.*, it resides in an n -dimensional affine subspace. Let $\psi \subseteq \{1, 2, \dots, n\}$ and $\psi \neq \emptyset$. With each ψ we associate a bidiagonal representation $q_\psi := (\vec{e}_1, \mathbf{Bi}_\psi)$, where the PH-generator \mathbf{Bi}_ψ is built by all λ_j 's such that $j \in \psi$. Let Ψ denote the collection of all such ψ . By Lemma 2.45, the associated PH distribution of each $\psi \in \Psi$ is on the boundary of the polytope.

We have shown in the proof of Lemma 4.20 that each of the n polytopes

$$\begin{aligned} & \text{conv}(\{\delta, q_{\{n\}}, q_{\{n-1, n\}}, \dots, q_{\{3, \dots, n-1, n\}}, q_{\{2, 3, \dots, n-1, n\}}\}), \\ & \text{conv}(\{\delta, q_{\{n\}}, q_{\{n-1, n\}}, \dots, q_{\{3, \dots, n-1, n\}}, q_{\{1, 3, \dots, n-1, n\}}\}), \\ & \quad \vdots \\ & \text{conv}(\{\delta, q_{\{n\}}, q_{\{n-2, n\}}, \dots, q_{\{2, \dots, n-2, n\}}, q_{\{1, 2, \dots, n-2, n\}}\}), \\ & \text{conv}(\{\delta, q_{\{n-1\}}, q_{\{n-2, n-1\}}, \dots, q_{\{2, \dots, n-2, n-1\}}, q_{\{1, 2, \dots, n-2, n-1\}}\}). \end{aligned}$$

is an $(n - 1)$ -dimensional polytope, *i.e.*, it resides in an $(n - 1)$ -dimensional affine subspace.

The intersection of all of these $(n - 1)$ -dimensional affine subspaces and the polytope $\text{PH}(\mathbf{Bi}(\lambda_1, \dots, \lambda_n))$ is exactly the region containing APH distributions of algebraic degree $n - 1$ or less with poles taken from $\{-\lambda_1, -\lambda_2, \dots, -\lambda_n\}$. We refer to this region as \mathcal{Q} . Thus, \mathcal{Q} is the union of countably many (in this case n) subsets of $(n - 1)$ -dimensional affine subspaces. But then in the n -dimensional affine subspace on which \mathcal{Q} and the polytope $\text{PH}(\mathbf{Bi}(\lambda_1, \dots, \lambda_n))$ reside, the region \mathcal{Q} has measure zero. \square

Remark: The lemma can also be proved using Theorem 5.6 and the fact that all ordered bidiagonal PH-generators are PH-simple. The PH-generator of the dual of an ordered bidiagonal representation, however, is not always PH-simple, but can be shown to be generically PH-simple in a similar manner as the proof of Theorem 5.1 in [CM02].

The lemma shows that even when we fix a particular size n representation structure (namely ordered bidiagonal) and particular parameter values for the total outgoing rates of the states of the representation, PH distributions of algebraic degree n are still almost everywhere. In this sense, the result described in Lemma 5.13 is stronger than that of [CM02] described above. On the other hand, the result is also more restricted, because it applies solely to ordered bidiagonal representations, and hence to only APH representations.

We showed in the previous lemma that almost all of the set of initial probability distributions of any ordered bidiagonal representation of size n gives rise to APH distributions of algebraic degree n . Since the results of the convolution, minimum, and maximum operations can always be transformed into ordered bidiagonal representations, we expect the same result can be established for them. In the rest of the section, we will prove that this is indeed the case. First, we require the following lemma.

Lemma 5.14. *Let $(\vec{e}_1, \mathbf{Bi}(\lambda_1, \dots, \lambda_m))$ and $(\vec{e}_1, \mathbf{Bi}(\mu_1, \dots, \mu_n))$ be two arbitrary hypoexponential representations. Then the number of states in the longest core series of*

1. $\text{con}((\vec{e}_1, \mathbf{Bi}(\lambda_1, \dots, \lambda_m)), (\vec{e}_1, \mathbf{Bi}(\mu_1, \dots, \mu_n)))$,
2. $\text{min}((\vec{e}_1, \mathbf{Bi}(\lambda_1, \dots, \lambda_m)), (\vec{e}_1, \mathbf{Bi}(\mu_1, \dots, \mu_n)))$, or
3. $\text{max}((\vec{e}_1, \mathbf{Bi}(\lambda_1, \dots, \lambda_m)), (\vec{e}_1, \mathbf{Bi}(\mu_1, \dots, \mu_n)))$,

is equal to the algebraic degree of its corresponding APH distribution.

A proof sketch of this lemma is available in Appendix B.2.

Now, we can prove the main theorem.

Theorem 5.15. *Convolution, minimum and maximum operations are triangular-ideal-preserving almost everywhere.*

Proof. To prove the lemma, we show that given two arbitrary ordered bidiagonal PH-generators \mathbf{Bi}_1 and \mathbf{Bi}_2 , the set of all possible initial probability vectors $\vec{\alpha}$ and $\vec{\beta}$ such that either $\text{con}((\vec{\alpha}, \mathbf{Bi}_1), (\vec{\beta}, \mathbf{Bi}_2))$, $\text{min}((\vec{\alpha}, \mathbf{Bi}_1), (\vec{\beta}, \mathbf{Bi}_2))$ or $\text{max}((\vec{\alpha}, \mathbf{Bi}_1), (\vec{\beta}, \mathbf{Bi}_2))$ represents a triangular-ideal APH distribution has measure zero in the set of all possible initial probability vectors $\vec{\alpha}$ and $\vec{\beta}$.

We accomplish this task as follows: A core series is constructed for the representation produced by the operation. Our knowledge of the structure of this representation

and the total outgoing rates of all states in it allow us to do the construction. Let p be the size (length) of the longest core series in this representation. Now, p is a new tighter lower bound for the number of states required to represent the APH distribution produced by the operation.

By using the obtained core series, we build the ordered bidiagonal PH-generator $\mathbf{Bi}(\nu_1, \dots, \nu_p)$ for the produced representation. PH-generator $\mathbf{Bi}(\nu_1, \dots, \nu_p)$ is PH-simple (cf. Theorem 3.4). For the given initial probability vectors $\vec{\alpha}$ and $\vec{\beta}$, let the ordered bidiagonal representation of the representation produced by the operation be $(\vec{\gamma}, \mathbf{Bi}(\nu_1, \dots, \nu_p))$.

In the rest of the proof, we will show that the PH-generator of the dual representation of $(\vec{\gamma}, \mathbf{Bi}(\nu_1, \dots, \nu_p))$ is PH-simple almost everywhere in the set of all possible $\vec{\alpha}$ and $\vec{\beta}$. By Theorem 5.6, if the PH-generator of the dual representation is PH-simple almost everywhere, then the algebraic degree of $\text{PH}(\vec{\gamma}, \mathbf{Bi}(\nu_1, \dots, \nu_p))$ is p for almost all $\vec{\alpha}$ and $\vec{\beta}$. Therefore, the set of $\vec{\alpha}$ and $\vec{\beta}$ such that the representation produced by the operation represents an APH distribution whose order is not equal to its algebraic degree (*i.e.*, not a triangular-ideal APH distribution) has measure zero.

To show that the PH-generator of the dual of $(\vec{\gamma}, \mathbf{Bi}(\nu_1, \dots, \nu_p))$ is PH-simple almost everywhere in the set of all possible $\vec{\alpha}$ and $\vec{\beta}$, we use a method similar to that of [CM02] we mentioned above. Let \mathbf{B} be the PH-generator of the dual representation of $(\vec{\gamma}, \mathbf{Bi}(\nu_1, \dots, \nu_p))$. From Equations (2.16) and (2.17), we obtain

$$\mathbf{B} = \mathbf{M}^{-1} \mathbf{Bi}(\nu_1, \dots, \nu_p)^\top \mathbf{M} \quad \text{and} \quad \vec{B} = \mathbf{M}^{-1} \vec{\gamma}^\top,$$

where matrix $\mathbf{M} = \text{diag}(\vec{m})$, and vector $\vec{m} = -\vec{\gamma} \mathbf{Bi}(\nu_1, \dots, \nu_p)^{-1}$. From Equation (2.24), PH-generator \mathbf{B} is PH-simple if and only if matrix

$$\begin{aligned} \mathbf{R} &= [\vec{B} \quad \mathbf{B}\vec{B} \quad \dots \quad \mathbf{B}^{n-1}\vec{B}], \\ &= \mathbf{M}^{-1} [\vec{\gamma}^\top \quad \mathbf{Bi}(\nu_1, \dots, \nu_p)^\top \vec{\gamma}^\top \quad \dots \quad (\mathbf{Bi}(\nu_1, \dots, \nu_p)^\top)^{n-1} \vec{\gamma}^\top] \end{aligned} \quad (5.10)$$

has rank p .

For matrix \mathbf{R} to have full rank p (and hence for \mathbf{B} to be PH-simple), $\det(\mathbf{R})$ must be nonzero. Matrix \mathbf{M} is nonsingular, and the determinant of \mathbf{M}^{-1} is a nontrivial polynomial (namely $\det(\mathbf{M}^{-1}) \neq 0$) in the parameters $\vec{\gamma}_1, \dots, \vec{\gamma}_p$. It follows that the determinant is a nontrivial polynomial in the $m+n$ parameters $\vec{\alpha}_1, \dots, \vec{\alpha}_m, \vec{\beta}_1, \dots, \vec{\beta}_n$. On the other hand, the determinant of matrix

$$[\vec{\gamma}^\top \quad \mathbf{Bi}(\nu_1, \dots, \nu_p)^\top \vec{\gamma}^\top \quad \dots \quad (\mathbf{Bi}(\nu_1, \dots, \nu_p)^\top)^{n-1} \vec{\gamma}^\top]$$

is also a polynomial in the mentioned $m+n$ parameters. In conclusion, $\det(\mathbf{R})$ is a polynomial in the $m+n$ parameters $\vec{\alpha}_1, \dots, \vec{\alpha}_m, \vec{\beta}_1, \dots, \vec{\beta}_n$.

Now, $\det(\mathbf{R}) = 0$ defines an algebraic variety in \mathbb{R}^{m+n} . An algebraic variety in \mathbb{R}^q is defined as the set of common zeros of a finite number of polynomials in q variables [CM02, Lin74]. We call an algebraic variety proper if at least one of the polynomials is nontrivial. A proper algebraic variety, furthermore, has measure zero in the parameter set \mathbb{R}^q . Therefore, if we prove that this algebraic variety $\det(\mathbf{R}) = 0$ is proper, namely that a particular realization $\vec{\alpha}'_1, \dots, \vec{\alpha}'_m, \vec{\beta}'_1, \dots, \vec{\beta}'_n$ results in a PH-generator \mathbf{B} that is PH-simple, then $\det(\mathbf{R}) \neq 0$ almost everywhere.

The particular realizations for the operations are given in Lemma 5.14. They are given by $\vec{\alpha}'_1 = 1, \vec{\alpha}'_i = 0$ for $2 \leq i \leq m, \vec{\beta}'_1 = 1$ and $\vec{\beta}'_i = 0$ for $2 \leq i \leq n$. \square

We established that the three operations are triangular-ideal-preserving almost everywhere. Hence, given two arbitrary triangular-ideal APH distributions, regardless of the size of their representations, we are almost certain that the APH distribution of their convolution, minimum, or maximum is also triangular ideal, and hence its APH representation is reducible to minimal size by applying our reduction algorithm. In other words, if we restrict ourselves to use only triangular-ideal APH distributions in a stochastic modelling formalism that is equipped with the three operations, we will only deal with models that are *almost surely minimal*.

5.5 Conclusion

In this chapter, we analyzed the reduction algorithm more extensively. We showed that it does not always produce the minimal representation of the input representation. The algorithm has also been compared with current existing results, which mostly specify the conditions under which an APH distribution is triangular ideal. Most of these existing results are not algorithmic, and, since most of them work in the LST domain, it is not obvious how to take advantage of them algorithmically. Here, the reduction algorithm can play a role, since, as we have shown, the algorithm always produces minimal representations when the input APH distributions are triangular ideal.

A question, however, arises as to how useful it is to have such an algorithm. After all, as we have repeatedly mentioned, for any pre-specified structure of a PH representation of size n , almost all parameter values produce PH distributions of algebraic degree n [CM02].

We tried to answer this question throughout the chapter by showing that PH distributions we usually use are not picked from the set of these parameter values uniformly, and therefore arise randomly. Instead, they are constructed using operations that induce some structure on the produced representations. As a result, more often than not, we end up with ideal (or triangular-ideal) PH distributions, but specified in PH representations of strictly larger size. This is certainly the case for the representations produced by the convolution, minimum, and maximum of APH representations. We studied the effect of these operations, and showed that given two arbitrary ordered bidiagonal representations whose APH distributions are triangular ideal, the three operations almost always produce a triangular-ideal APH distribution, whose minimal representation can always be obtained.

This result is encouraging because these operations occur in many varieties in many stochastic modelling formalisms. Furthermore, since each of the operations is commutative and associative, we can build a stochastic modelling formalism that is compositional based on the operations. In the next chapter, we will develop a simple calculus that captures the intention behind and the purpose of these operations.

Chapter 6

A Simple Stochastic Calculus

In this chapter, we develop a simple stochastic process calculus that we use to generate and manipulate APH representations. We call the calculus the *Cox & Convenience Calculus* or, for short, CCC.

A process calculus is a formalism for specifying a system as a composition of smaller subsystems by means of a formal language with well-defined semantics. The formalism provides a mechanism to capture the important aspects and the behaviors of the system. It also enables us to reason about the system at a syntactical level. A process calculus is usually defined by its set of actions and several operations. An action signifies the occurrence of an activity, and it also forms the simplest process. An action constitutes the basic behavior of a process. A more complex process, exhibiting a more complex behavior, can be obtained by composing several processes by using the operations. Several types of composition of processes involve synchronizations among actions that are common to the processes. This entire process specification is carried out in a purely syntactical way by means of the language. There currently exist many process calculi, the most prominent among them are CSP [Hoa78], CCS [Mil95], and ACP [BW90]. A specification language called LOTOS [BB87], which is based on the CSP, has been standardized by ISO.

Ordinary process calculi can capture the functional behavior of a system through its actions and the structure of their occurrences. One of the ways to incorporate non-functional behaviors, such as time, duration, and probability to a process calculus is by adding a stochastic notion to govern the occurrences—when or for how long—of the actions. The resulting calculus is called a *stochastic process calculus*. Most stochastic process calculi are Markovian, *i.e.*, the stochastic behaviors of the calculi are governed at the basic level by exponential distributions.

Related Work Like ordinary process calculi, many variants of stochastic process calculi have been proposed. Among Markovian calculi include PEPA [Hil96], EMPA [BG96], MTIPP [HR94], IMC [Her02], IGSMC [BG02], and Spades [DK05]. These stochastic process calculi use exponential distributions to govern the time durations between the occurrences of actions. The primary difference among them stems from the way they treat actions and durations, namely whether several time durations—each governed by an exponential distribution—can occur one after the other, or whether only a single time duration is allowed to expire between the occurrence of two actions. More recently, a stochastic process calculus in which the basic durations are governed by PH distributions is proposed in [Wol08].

The stochastic process calculus we develop in this chapter is loosely based on the previous calculi. A process in our calculus reflects the completion time of some activity. As basic processes we use exponential distributions. More complex processes are built by composing basic processes using several operations. The calculus has no action synchronization mechanism and no recursion operation. All processes, from the basic to the most complex, are therefore acyclic. The semantics of the calculus will be absorbing CTMCs. These absorbing CTMCs correspond to APH representations of APH distributions that govern the completion times of processes.

Contribution The contributions of this chapter are twofold. First, we develop a simple stochastic process calculus that can be used to specify the completion time of a system. We show that every process generated by the language of the calculus coincides with a representation of an APH distribution. Second, we introduce three notions of equivalence among the processes. Two of these notions are variants of the strong and weak-bisimulation equivalences on absorbing CTMCs. The third works on the level of the distributions, and equates two processes whenever their completion times are distributed in the same way. This latter notion enables us to use our reduction algorithm to reduce the size of processes.

Structure The chapter is organized as follows: In Section 6.1, we define the language of CCC by specifying its syntax and semantics. The semantics of the language is shown in Section 6.2 to correspond to APH representations, whose distributions describe the completion time of the processes. In Section 6.3, we define three notions of equivalence on CCC processes. We show that each notion is a congruence with respect to all operators of the language. Section 6.4 lays out the procedures to check each notion of equivalence. We summarize and conclude the chapter in Section 6.5.

6.1 CCC Processes

In this section, we first provide the general intuitions behind CCC processes. Afterwards, we define a language to specify CCC processes as well as the semantics of each operator in the language.

6.1.1 Intuitions

The calculus we are developing in this chapter can be regarded as being composed of two parts: Cox and convenience. Both are intended to be a means for capturing the mechanism behind the generation (Cox) and manipulation (convenience) of APH representations.

Cox To design a process calculus that captures APH representations, we start from the basic building blocks, the simplest processes that can be expressed. We choose exponential distributions to be the basic building blocks of our process calculus. An exponential distribution with rate $\lambda \in \mathbb{R}_+$ is expressed in the process calculus simply by (λ) . The intended semantics of this basic process (λ) is an (unspecified) activity that runs for a duration that is distributed according to an exponential distribution with

rate λ and then terminates. In general, the random completion time of the activities a process entails is regarded as the behavior of the process.

As mentioned in Section 3.1.2, one of the advantages of the Cox representations is the fact that each of them starts in a single state, which is important in developing the process calculus. Our process calculus will generate APH representations in Cox forms. From the basic building blocks, then, we aim at generating the whole set of Cox representations by using one additional operator. For this purpose we introduce the disabling operator \triangleleft . A disabling operation takes the form of $(\mu) \triangleleft (\lambda)P$, where $\mu \in \mathbb{R}_{\geq 0}$, $\lambda \in \mathbb{R}_+$, and P is an arbitrary process generated by the calculus.

Intuitively, we interpret the disabling operation as follows. Let $(\lambda)P$ be a process P preceded by a time duration distributed according to an exponential distribution with rate λ . Process (μ) , on the one hand, and process $(\lambda)P$, on the other hand, disable each other, in the sense that the occurrence of (μ) cancels process $(\lambda)P$, and the occurrence of (λ) cancels process (μ) .

The intended precise semantics of the operation $(\mu) \triangleleft (\lambda)P$ is a race to terminate between processes (μ) and (λ) . The race between two exponential distributions with rates μ and λ is the minimum of the two distributions, and this corresponds to an exponential distribution with rate $\mu + \lambda$. Thus, the delay until the race finishes is governed by an exponential distribution with rate $\mu + \lambda$. Furthermore, the winner of the race is decided probabilistically: process (μ) wins with probability $\frac{\mu}{\mu + \lambda}$, and process (λ) wins with probability $\frac{\lambda}{\mu + \lambda}$. At the end of the race, if process (μ) wins, the whole process terminates, while if process (λ) wins, process P is started.

By observing the semantics of Cox representations and the semantics of the disabling operator above, we expect that, by repeated applications of the disabling operations on existing processes built from the basic building blocks, we can always form a process that describes an arbitrary Cox representation. Furthermore, since any APH representation can be transformed into a Cox representation, a process that describes any APH representation can always be obtained. This will be made formal in Lemma 6.11.

Example 6.1. We refer back to the Cox representation discussed in Example 3.6, which we reproduce in Figure 6.1. In CCC, the Cox representation can be expressed by the following process

$$P = \left(\frac{5}{4}\right) \triangleleft \left(\frac{11}{4}\right) \left(\left(\frac{14}{11}\right) \triangleleft \left(\frac{19}{11}\right)(1) \right).$$

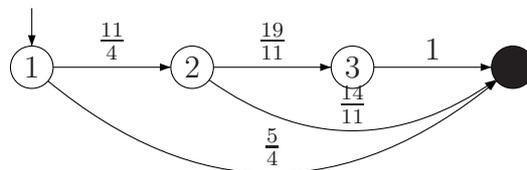


Figure 6.1: A Cox Representation

Convenience The second part of the process calculus is the “convenience” part, namely the part that helps us manipulate APH representations. In Section 5.3, we have described three operations on APH representations, namely convolution, minimum, and

maximum. These three operations constitute the mechanism for the manipulation of APH representations that we would like to be captured by the process calculus. Recall that these three operations on APH representations are actually inherited from the corresponding operations on PH distributions. On the distribution level, the operations correspond to the sum, minimum, and maximum operations, respectively, of two distributions. In our process calculus, we define three composition operators to carry out these operations. They are sequential (\cdot), choice ($+$), and parallel (\parallel) operators, respectively.

Process $P.Q$ describes the *sequential composition* of processes P and Q in that particular order. This process behaves as P until it terminates. Upon the termination of P , it then behaves as Q . The overall behavior of $P.Q$ is therefore the sum of the completion times of P and Q .

Process $P + Q$, on the other hand, describes the *choice composition* of processes P and Q . This process behaves either as P or as Q , whichever terminates first. Since a process describes its random completion time, we can think of process $P + Q$ as the race between the two processes: whichever has the least completion time wins. Therefore, when either of them terminates, the process itself terminates. The overall behavior of $P + Q$ is then the minimum of the completion times of P and Q .

Lastly, process $P\parallel Q$ describes the *parallel composition* of processes P and Q . This process behaves as P and as Q , at the same time. For the process to terminate, both processes P and Q must terminate. Hence, the completion time of process $P\parallel Q$ corresponds to the largest completion time of its components. In other words, the overall behavior of $P\parallel Q$ is the maximum of the completion times of P and Q .

6.1.2 Syntax

The language for the specification of the CCC processes is specified by a grammar, which consists of non-terminal and terminal symbols. The terminal symbols play the roles of disabling, sequential, choice, and parallel operators. We define the set \mathcal{L} of all CCC expressions as follows.

Definition 6.2 (CCC Syntax). *Let \mathcal{L} be the language defined by the following grammar*

$$P ::= (\lambda) \mid (\mu) \triangleleft (\lambda)P \mid P.P \mid P + P \mid P\parallel P$$

where each of $\lambda \in \mathbb{R}_+$ and $\mu \in \mathbb{R}_{\geq 0}$ is called a rate. Each element of \mathcal{L} is called a CCC process.

In the rest of the thesis, we use $P, P_1, P_2, \dots, P', Q, R, \dots$ to range over processes, and $\lambda, \lambda_1, \lambda_2, \dots, \mu, \nu, \dots$ to range over rates.

We fix the precedence of the operators in the language as follows: disabling (\triangleleft), sequential (\cdot), choice ($+$), and parallel (\parallel) operators. Thus, the disabling operator takes precedence over sequential, choice, and parallel operators, sequential operator takes precedence over choice and parallel operators, and so on. To circumvent these default precedence rules, we may use parentheses. Furthermore, the operators described above are binary operators. Unparenthesized repeated applications of the same operator are assumed to have a left-associative evaluation order: thus $P\parallel Q\parallel R$ is evaluated as $(P\parallel Q)\parallel R$.

6.1.3 Semantics

We have established the syntax of the language of the stochastic calculus we are developing. We also have described the intuitive interpretation of each operator in the language. In the following we formally define the semantics of the language. The semantics comes in two stages. In the first stage, the entire language is mapped onto a labelled transition system, while in the second, the obtained labelled transition system is interpreted as an absorbing CTMC.

MLTS Semantics In the first stage, we define a Markovian labelled transition system, onto which language \mathcal{L} is mapped. For that purpose, first, let Lab be the set of strings defined by the grammar

$$w ::= \varepsilon \mid \triangleleft_l .w \mid \triangleleft_r .w \mid +_l .w \mid +_r .w \mid \parallel_l .w \mid \parallel_r .w$$

where ε is the empty string, and ‘.’ is the concatenation operator. The subscripts l and r stand for left and right, respectively. The strings will be used to differentiate different transitions having the same rate between two processes.

Definition 6.3 (MLTS Semantics). A Markovian Labelled Transition System (MLTS) is a tuple $\mathfrak{M} = (\mathcal{L}, P_0, \longrightarrow)$, where

- \mathcal{L} is the set of all CCC processes,
- P_0 is the initial process, and
- $\longrightarrow \subseteq (\mathcal{L} \times (\mathbb{R}_+ \times Lab) \times \mathcal{L})$ is the least relation satisfying the derivation rules listed in Table 6.1

The expression $stop$, in the Table 6.1, is a terminal symbol describing a terminated behavior. As a process, $stop$ does nothing.

A member of the set \longrightarrow is called a transition. The transitions among processes are obtained by applying the given set of the derivation rules. For our reading convenience, we write $P \xrightarrow{\lambda, w} Q$ to denote the existence of a transition from state P to Q with rate λ and label w , instead of $(P, (\lambda, w), Q) \in \longrightarrow$.

The formal (first stage) semantics of language \mathcal{L} is given in the style of *Structural Operational Semantics* (SOS) [Pl04, AFV01]. The intuitive interpretation described above is formalized by means of SOS derivation rules depicted in the table. In general, a SOS derivation rule is of the form

$$\frac{\text{Premises}}{\text{Conclusions}} \text{ (Conditions),}$$

which stipulates that given that the Conditions are valid, the validity of the Premises, under a certain substitution, implies the validity of the Conclusions under the same substitutions [AFV01]. Conditions or Premises are empty in some rules. Those rules with empty Premises serve as the axioms of the SOS.

Among the SOS rules of language \mathcal{L} listed in Table 6.1 are three axioms, namely rules (1), (2.a), and (2.b). Axiom (2.a) specifies that for all processes of the form $(\mu) \triangleleft(\lambda)P$, transition $(\mu) \triangleleft(\lambda)P \xrightarrow{\mu, \triangleleft_l} stop$ is valid. The remaining rules in the table are derivation rules; take for instance rule (4.a). This rule states that given that process P'

Table 6.1: The SOS Derivation Rules for Language \mathcal{L}

$$(1) \quad \frac{}{(\lambda) \xrightarrow{\lambda, \varepsilon} \text{stop}}$$

$$(2.a) \quad \frac{}{(\mu) \triangleleft (\lambda) P \xrightarrow{\mu, \triangleleft_l} \text{stop}} \quad \frac{}{(\mu) \triangleleft (\lambda) P \xrightarrow{\lambda, \triangleleft_r} P} \quad (2.b)$$

$$(3.a) \quad \frac{P \xrightarrow{\lambda, w} P'}{P.Q \xrightarrow{\lambda, w} P'.Q} \quad (P' \neq \text{stop}) \quad \frac{P \xrightarrow{\lambda, w} \text{stop}}{P.Q \xrightarrow{\lambda, w} Q} \quad (3.b)$$

$$(4.a) \quad \frac{P \xrightarrow{\lambda, w} P'}{P + Q \xrightarrow{\lambda, +l.w} P' + Q} \quad (P' \neq \text{stop})$$

$$(4.b) \quad \frac{Q \xrightarrow{\lambda, w} Q'}{P + Q \xrightarrow{\lambda, +r.w} P + Q'} \quad (Q' \neq \text{stop})$$

$$(4.c) \quad \frac{P \xrightarrow{\lambda, w} \text{stop}}{P + Q \xrightarrow{\lambda, +l.w} \text{stop}}$$

$$(4.d) \quad \frac{Q \xrightarrow{\lambda, w} \text{stop}}{P + Q \xrightarrow{\lambda, +r.w} \text{stop}}$$

$$(5.a) \quad \frac{P \xrightarrow{\lambda, w} P'}{P \parallel Q \xrightarrow{\lambda, \parallel_l.w} P' \parallel Q} \quad \frac{Q \xrightarrow{\lambda, w} Q'}{P \parallel Q \xrightarrow{\lambda, \parallel_r.w} P \parallel Q'} \quad (5.b)$$

is not equal to stop , whenever $P \xrightarrow{\lambda, w} P'$ is a valid transition, then so is the transition $P + Q \xrightarrow{\lambda, +l, w} P' + Q$. The other rules are interpreted in a similar fashion.

Each transition in an MLTS has two labels: the rate of the corresponding transition and an additional label. The additional label is necessary to allow us to differentiate transitions that otherwise would be indistinguishable, namely to explicitly produce different transitions if there exist different derivations with the same rate between two processes. For instance, the process $(\lambda) + (\mu)$ has two transitions to process stop , one with rate λ and the other with rate μ . In this case, the additional label is not required. However, in the case of a process of the form $(\lambda) + (\lambda)$, the additional label is necessary to uphold that there are indeed two different derivations for rate λ , namely the transitions

$$(\lambda) + (\lambda) \xrightarrow{\lambda, +l} \text{stop} \quad \text{and} \quad (\lambda) + (\lambda) \xrightarrow{\lambda, +r} \text{stop}.$$

This technical means to overcome the restriction of multiple transitions with the same rate (or in other contexts, action or label) between any two processes was first described in [BC88], and has been widely used, for instance in [HR94].

In addition to the SOS derivation rules above, a structural congruence rule, which is an identity related to rule (5), is required

$$\text{stop} \parallel \text{stop} \stackrel{\text{def}}{=} \text{stop}. \quad (6.1)$$

This rule defines the identity of a parallel composition of two stop processes. The rule cannot be deduced from the SOS derivation rules, because it is a state-related—instead of a transition-related—rule, and it is essential for the second-stage semantics. This notion of equality between process definitions is called structural congruence, for instance, in π -calculus [Mil99]. Two processes are structurally congruent, if they are identical up to a structure.

Given a process $P \in \mathcal{L}$, the SOS derivation rules enable us to derive all transitions that originate from P . This is achieved by repeated applications of the rules. Performing the derivation in depth-first manner, we obtain a path. Formally, a *finite path* $\sigma^{\mathfrak{M}}$ of length $n \in \mathbb{Z}_{\geq 0}$ in an MLTS \mathfrak{M} is an alternating sequence of processes and labels

$$P_0; (\lambda_1, w_1); P_1; (\lambda_2, w_2); P_2; \cdots; P_{n-1}; (\lambda_n, w_n); P_n,$$

such that, for all $1 \leq i \leq n$, it holds that $P_{i-1} \xrightarrow{\lambda_i, w_i} P_i$. Note that $\sigma^{\mathfrak{M}} = P_0$ is a path of length 0. A process $P' \in \mathcal{L}$ is *reachable* from $P \in \mathcal{L}$, if there is $n \in \mathbb{Z}_{\geq 0}$ and a path $\sigma^{\mathfrak{M}}$ of length n such that $P = P_0$ and $P' = P_n$. If process P' is reachable from process P we write $P \Longrightarrow P'$.

Now, the *derivatives* of P are all those processes in \mathcal{L} that are reachable from P . We refer to the set of these processes as $\text{Reach}(P)$, i.e.,

$$\text{Reach}(P) = \{P' \in \mathcal{L} \mid P \Longrightarrow P'\}.$$

Example 6.4. Consider a process $P \in \mathcal{L}$ defined as follows

$$\begin{aligned} P &= P_1.P_2 \parallel P_3, \text{ where} \\ P_1 &= (\lambda_1) + (\lambda_1).(\lambda_2), \\ P_2 &= (\lambda_3), \text{ and} \\ P_3 &= (\lambda_4).(\lambda_5). \end{aligned}$$

Process P is a composition of processes P_1 , P_2 and P_3 . Figure 6.2 depicts the semantics of these processes in MLTSS. Using the SOS derivation rules, process P can be obtained by composing the three previously depicted processes. According the precedence rules, process $P_1.P_2$ is evaluated first, followed by the evaluation of $P_1.P_2\|P_3$. The resulting MLTS semantics of process P is depicted in Figure 6.3.

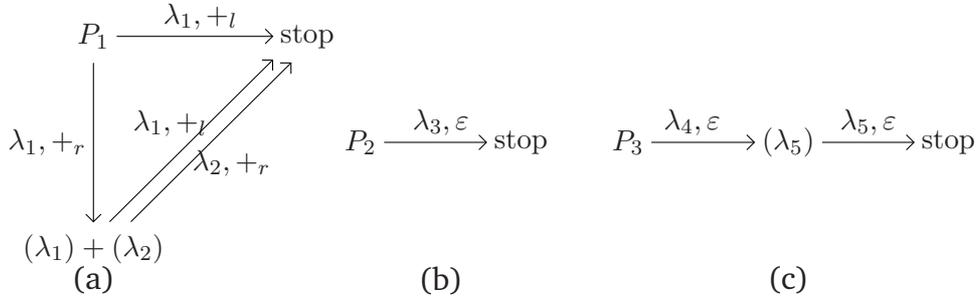


Figure 6.2: The MLTSS of (a) P_1 , (b) P_2 and (c) P_3

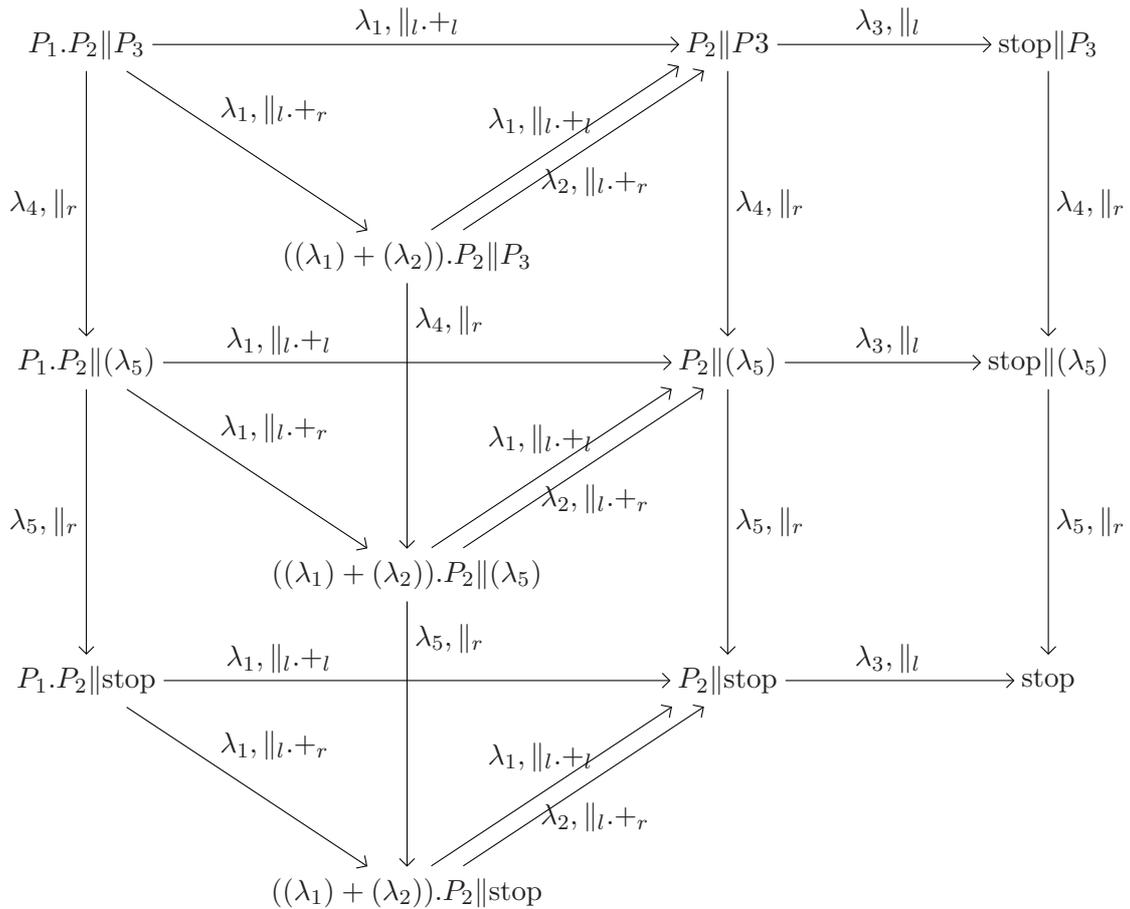


Figure 6.3: The MLTS of $P_1.P_2\|P_3$

Both Figure 6.2 and Figure 6.3 depict only the set of the reachable processes of each process. For instance, process P_1 can reach three processes, namely $\text{Reach}(P_1) =$

$\{P_1, (\lambda_1) + (\lambda_2), \text{stop}\}$, while process $P_1.P_2\|\text{stop}$ can reach four processes, namely

$$\text{Reach}(P_1.P_2\|\text{stop}) = \{P_1.P_2\|\text{stop}, ((\lambda_1) + (\lambda_2)).P_2\|\text{stop}, P_2\|\text{stop}, \text{stop}\}.$$

From this point onward, we will omit the additional labels from all transitions. If more than one transition exists between two particular processes, we will ensure that all transitions are listed or depicted without explicitly distinguishing them through the additional labels.

Absorbing CTMC Semantics In the second stage of the semantics, we define an interpretation of the MLTS we obtained in the first stage in terms of an absorbing CTMC. For that purpose, we need to define a function that accumulates the rate of all transitions between two processes in the MLTS.

Definition 6.5. Let the function $\gamma : (\mathcal{L} \times \mathcal{L}) \rightarrow \mathbb{R}_{\geq 0}$ be defined as follows

$$\gamma(P, Q) = \sum_{(\lambda, w) \in \{(\lambda, w) \mid P \xrightarrow{\lambda, w} Q\}} \lambda. \quad (6.2)$$

Function γ is necessary because the SOS derivation rules may derive more than one transition between two distinct processes. In CTMC, in contrast, there can be no more than one transition between any two distinct states. The function γ is used to unify these transitions into a single transition by summing their rates.

Definition 6.6 (Absorbing CTMC Semantics). The absorbing CTMC semantics of a process $P \in \mathcal{L}$, whose MLTS semantics is $\mathfrak{M} = (\mathcal{L}, P, \longrightarrow)$, is $\mathcal{M}_P = (\mathcal{S}, \mathbf{R}, \vec{\pi})$, where

- the state space $\mathcal{S} = \text{Reach}(P)$,
- the rate matrix $\mathbf{R}(P, Q) = \gamma(P, Q)$, and
- the initial probability distribution $\vec{\pi} = \vec{e}_P$, where \vec{e}_P is the unit vector at position P of $\mathbb{R}_{\geq 0}^{|\mathcal{S}|}$.

Once we have the semantics of a CCC process in terms of an MLTS, we can map the obtained MLTS onto an absorbing CTMC. We note that the size of the state space of \mathcal{M}_P is $|\mathcal{S}| = |\text{Reach}(P)|$.

Example 6.7. Continuing from Example 6.4, the absorbing CTMC semantics of the MLTS depicted in Figure 6.3 is presented in Figure 6.4. For reading convenience, we have renamed all reachable MLTS processes in the CTMC figure: $P_1.P_2\|P_3$ to 1, $((\lambda_1) + (\lambda_2)).P_2\|P_3$ to 2, and so on. The process stop in the MLTS is renamed to an absorbing state, the black-shaded state in the CTMC. The positional correspondence between the processes and the states is evident from the two figures.

6.2 CCC Processes and PH Distributions

The absorbing CTMC depicted in Figure 6.4 can represent an APH distribution. Hence, the distribution of the time to hit the absorbing state in the underlying CTMC governs the completion time of the process $P_1.P_2\|P_3$. However, are we sure that the absorbing CTMC semantics of any process $P \in \mathcal{L}$ always represents an APH distribution?

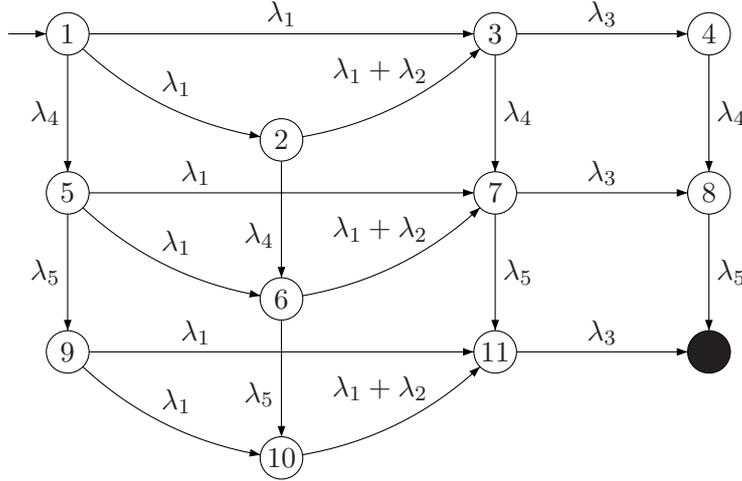


Figure 6.4: The Absorbing CTMC Semantics of Process $P_1.P_2||P_3$

Lemma 6.8. For every $P \in \mathcal{L}$, $P \Longrightarrow \text{stop}$.

Proof. We prove the lemma by induction on process $P \in \mathcal{L}$. As the base of the induction, let process $P = (\lambda)$. By SOS rule (1), $P \Longrightarrow \text{stop}$.

For the induction steps, assume that for each of arbitrary $P_1, P_2 \in \mathcal{L}$, the lemma is valid. Then it is also valid for:

1. $P = (\mu) \triangleleft (\lambda)P_1$: $P \Longrightarrow \text{stop}$, for instance, by SOS rule (2.a).
2. $P = P_1.P_2$: By SOS rules (3.a) and (3.b), $P \Longrightarrow P_2$ and $P_2 \Longrightarrow \text{stop}$. Therefore $P \Longrightarrow \text{stop}$.
3. $P = P_1 + P_2$: By repeated applications of SOS rules (4.a) $P \Longrightarrow (\lambda) + P_2$ for some $\lambda \in \mathbb{R}_+$, and by SOS rules (4.c) $(\lambda) + P_2 \Longrightarrow \text{stop}$. Therefore $P \Longrightarrow \text{stop}$.
4. $P = P_1||P_2$: By repeated applications of SOS rules (5.a) $P \Longrightarrow \text{stop}||P_2$, and then by repeated applications of SOS rules (5.b) $\text{stop}||P_2 \Longrightarrow \text{stop}$. Therefore $P \Longrightarrow \text{stop}$.

□

Since every process $P \in \mathcal{L}$ can reach the process stop , $|\text{Reach}(P)|$ is finite.

Lemma 6.9. For every $P \in \mathcal{L}$, $\mathcal{M}_P = (\mathcal{S}, \mathbf{R}, \vec{\pi})$ is an underlying CTMC of an acyclic phase-type representation.

Proof. For \mathcal{M}_P to be an underlying CTMC of a PH representation, it must be finite, has a single absorbing state, and the rest of its states are transient.

Since there is only a single process stop in language \mathcal{L} and we associate this process with the absorbing state, to show that \mathcal{M}_P contains a single absorbing state, we show $P \Longrightarrow \text{stop}$. Hence, by Lemma 6.8, \mathcal{M}_P is finite and contains a single absorbing state. By the same lemma, it is immediate that each state in \mathcal{M}_P can reach the absorbing state, which implies that it does not contain any non-transient state.

The absence of a recursion operator in the grammar of language \mathcal{L} implies that each $P \in \mathcal{L}$ is acyclic. Therefore \mathcal{M}_P is acyclic too, and \mathcal{M}_P is an underlying CTMC of an APH representation. □

Now that we are certain that the absorbing CTMC semantics of any process $P \in \mathcal{L}$ always represents an APH distribution, we formalize their relationship in the following definition.

Definition 6.10. For a CCC process $P \in \mathcal{L}$, let $\text{PH}(P)$ be the PH distribution associated with $\mathcal{M}_P = (\mathcal{S}, \mathbf{R}, \vec{\pi})$.

In the following lemma, we prove a stronger assertion than the fact that the absorbing CTMC semantics of any process $P \in \mathcal{L}$ always represents an APH distribution. The lemma establishes that for any APH distribution having no mass at $t = 0$, we can generate a corresponding CCC process only by using the disabling operator.

Lemma 6.11. Let PH be an acyclic phase-type distribution having no mass at $t = 0$. There is a process P , which is generated only by using the disabling operator, such that $\text{PH} = \text{PH}(P)$.

Proof. PH must have an APH representation. We can transform this representation to a Cox distribution by using the spectral polynomial algorithm (see Section 3.3.3) and Theorem 3.5. Let $(\vec{e}_1, \mathbf{Cx}([\lambda_1, p_1], [\lambda_2, p_2], \dots, \lambda_n))$ be the obtained Cox representation, then

$$\text{PH} = \text{PH}(\vec{e}_1, \mathbf{Cx}([\lambda_1, p_1], [\lambda_2, p_2], \dots, \lambda_n)).$$

Let process $P = P_1$ be defined as follows:

1. $P_n = (\lambda_n)$,
2. For $1 \leq i < n$,

$$P_i = (\mu) \triangleleft (\lambda) P_{i+1},$$

where $\mu = (1 - p_i)\lambda_i$ and $\lambda = p_i\lambda_i$.

From the form of the Cox representation described in Section 3.1.2, \mathcal{M}_P is the underlying CTMC of Cox representation $(\vec{e}_1, \mathbf{Cx}([\lambda_1, p_1], [\lambda_2, p_2], \dots, \lambda_n))$. Therefore $\text{PH} = \text{PH}(\vec{e}_1, \mathbf{Cx}([\lambda_1, p_1], [\lambda_2, p_2], \dots, \lambda_n)) = \text{PH}(P)$. \square

In the beginning of Section 6.1, we started the development of our simple calculus by laying out the intuitive interpretation of each operator of language \mathcal{L} . This interpretation is actually a goal, and we have worked out the semantics to achieve it. In the following, we verify whether the intuitive interpretation is matched by the semantics.

Lemma 6.12. For all processes $P, Q \in \mathcal{L}$:

1. $\text{con}(\text{PH}(P), \text{PH}(Q)) = \text{PH}(P.Q)$,
2. $\text{min}(\text{PH}(P), \text{PH}(Q)) = \text{PH}(P + Q)$, and
3. $\text{max}(\text{PH}(P), \text{PH}(Q)) = \text{PH}(P \parallel Q)$.

A proof sketch of this lemma is available in Appendix B.3.

For the case of $(\mu) \triangleleft (\lambda)P$, assume that the APH representation associated with \mathcal{M}_P is (\vec{e}_1, \mathbf{A}) . By following SOS rules (2.a) and (2.b), we conclude that the representation associated with the absorbing CTMC semantics of $(\mu) \triangleleft (\lambda)P$ is given by $(\vec{e}'_1, \mathbf{A}')$, where

$$\mathbf{A}' = \begin{bmatrix} -(\mu + \lambda) & \lambda \vec{e}_1 \\ \vec{0} & \mathbf{A} \end{bmatrix}, \quad (6.3)$$

and \vec{e}_1 is a row vector whose first component is equal to 1 of appropriate dimension.

We close this section by offering the following remarks. In the calculus, we proposed the disabling operator instead of the traditional and more general choice operator. The traditional choice operator usually proceeds by selecting one from several processes based on the set of outgoing transitions from the processes. Thus the operator provides a mechanism to branch to the starting points of several processes. The reason we proposed one but not the other is so that the PH-equivalence notion—an equivalence notion on CCC processes that we will introduce in the next section—is a congruence with respect to all operators of language \mathcal{L} . With respect to the traditional choice operator, on the other hand, PH-equivalence is not a congruence. We will show this in the end of Section 6.3.

6.3 Some Notions of Equivalence

In this section, we introduce three notions of equivalence among CCC processes. A notion of equivalence defines the circumstances in which we can deem two CCC processes to be equivalent in their behaviors. The first two, strong and weak-bisimulation equivalences, are closely related to Markovian strong bisimulation [Buc94, Hil96] and Markovian weak bisimulation [Bra02], respectively. Both of them operate on the level of the transition relations of the processes in question. The third notion operates on the level of the completion times of the processes.

6.3.1 Bisimulations

For the purpose of defining strong and weak-bisimulation relations, we first define a generalization of the function γ defined in Equation (6.2). The new function γ_c accumulates the rates of transitions from a process $P \in \mathcal{L}$ to all processes in $C \subseteq \mathcal{L}$.

Definition 6.13. *Let the function $\gamma_c : (\mathcal{L} \times 2^{\mathcal{L}}) \rightarrow \mathbb{R}_{\geq 0}$ be defined as follows*

$$\gamma_c(P, C) = \sum_{Q \in C} \sum_{(\lambda, w) \in \{(\lambda, w) \mid P \xrightarrow{\lambda, w} Q\}} \lambda, \quad (6.4)$$

where $C \subseteq \mathcal{L}$.

If \mathcal{R} is an equivalence relation—*i.e.*, a relation that is reflexive, symmetric, and transitive—on \mathcal{L} , then let \mathcal{L}/\mathcal{R} be the partitioning of \mathcal{L} induced by \mathcal{R} , and for $P \in \mathcal{L}$, let $[P]_{\mathcal{R}}$ be the partition (class) that contains P . We write $P\mathcal{R}Q$ when $(P, Q) \in \mathcal{R}$. The two notations are interchangeable.

Strong Bisimulation We define the strong-bisimulation relation in a same style as that of [BKHW05, Her02], but for the absence of labelling on processes. We explicitly restrict process `stop` to be the solitary member of its class to simplify subsequent proofs.

Definition 6.14. *An equivalence relation $\mathcal{S} \subseteq \mathcal{L} \times \mathcal{L}$ is a strong bisimulation on \mathcal{L} if and only if (1) $[\text{stop}]_{\mathcal{S}} = \{\text{stop}\}$ and (2) $(P, Q) \in \mathcal{S}$ implies that*

$$\forall C \in \mathcal{L}/\mathcal{S} : \gamma_c(P, C) = \gamma_c(Q, C). \quad (6.5)$$

Two processes P and Q are strongly bisimilar (denoted by $P \sim Q$) if there exists a strong bisimulation \mathcal{S} such that $(P, Q) \in \mathcal{S}$.

Example 6.15. Let process $P \in \mathcal{L}$ be defined by $P = ((\lambda_1) \parallel (\lambda_1)).(\lambda_2)$. Figure 6.5 shows the semantics of the process in MLTS.

A relation \mathcal{S} that identifies processes $P_1 := ((\lambda_1) \parallel \text{stop}).(\lambda_2)$ and $P_2 := (\text{stop} \parallel (\lambda_1)).(\lambda_2)$, processes P , (λ_2) , and stop with themselves, respectively, is a strong bisimulation on \mathcal{L} . This can be verified as $\gamma_c(P_1, [(\lambda_2)]_{\mathcal{S}}) = \gamma_c(P_2, [(\lambda_2)]_{\mathcal{S}}) = \lambda_1$, and for all other $C \in \mathcal{L}/\mathcal{S}$ it holds that $\gamma_c(P_1, C) = \gamma_c(P_2, C) = 0$. Therefore we conclude that $P_1 \sim P_2$.

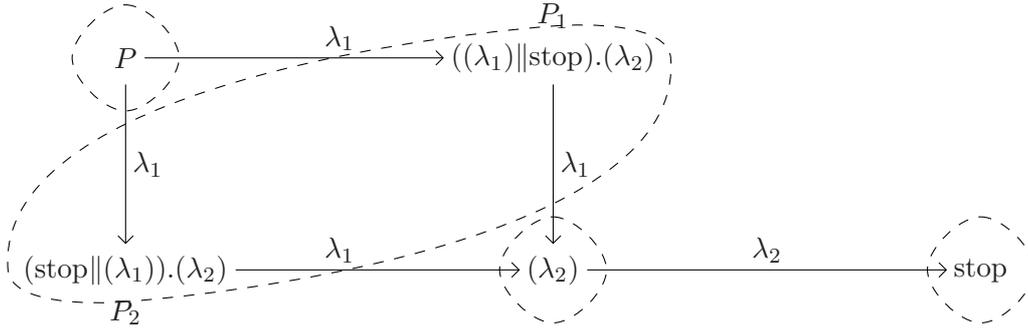


Figure 6.5: The MLTS of $P = ((\lambda_1) \parallel (\lambda_1)).(\lambda_2)$

Weak Bisimulation The weak-bisimulation relation is also defined in a similar style as that of [BKHW05, Her02]. However, the absence of the process labelling enforces us to put process stop in its own partition to avoid identifying the whole set of processes in a single equivalence class.

Definition 6.16. An equivalence relation $\mathcal{W} \subseteq \mathcal{L} \times \mathcal{L}$ is a weak bisimulation on \mathcal{L} if and only if (1) $[\text{stop}]_{\mathcal{W}} = \{\text{stop}\}$ and (2) $(P, Q) \in \mathcal{W}$ implies that

$$\forall C \in \mathcal{L}/\mathcal{W}, C \neq [P]_{\mathcal{W}} : \gamma_c(P, C) = \gamma_c(Q, C). \quad (6.6)$$

Two processes P and Q are weakly bisimilar (denoted by $P \approx Q$) if there exists a weak bisimulation \mathcal{W} such that $(P, Q) \in \mathcal{W}$.

Example 6.17. Let process $Q \in \mathcal{L}$ be defined by $Q = (2\lambda_1).((\lambda_2) \triangleleft (\lambda_1)(\lambda_2)).(\lambda_1)$. The MLTS semantics of the process Q is depicted in Figure 6.6.

A relation \mathcal{W} that identifies process $Q_1 := ((\lambda_2) \triangleleft (\lambda_1)(\lambda_2)).(\lambda_1)$ with process $Q_2 := (\lambda_1).(\lambda_2)$, processes Q , (λ_1) , and stop with themselves, respectively, is a weak bisimulation on \mathcal{L} . This can be verified as $\gamma_c(Q_1, [(\lambda_1)]_{\mathcal{W}}) = \gamma_c(Q_2, [(\lambda_1)]_{\mathcal{W}}) = \lambda_2$, and for all other $C \in \mathcal{L}/\mathcal{W}$, such that $C \neq [Q_1]_{\mathcal{W}}$, it holds that $\gamma_c(Q_1, C) = \gamma_c(Q_2, C) = 0$. Therefore we conclude that $Q_1 \approx Q_2$.

The two notions of equivalence defined above provide a compositional notion of semantics for CCC that is consistent with the structural operational semantics defined in Table 6.1. In short, both notions of equivalence are congruences, as shown in the following lemma.

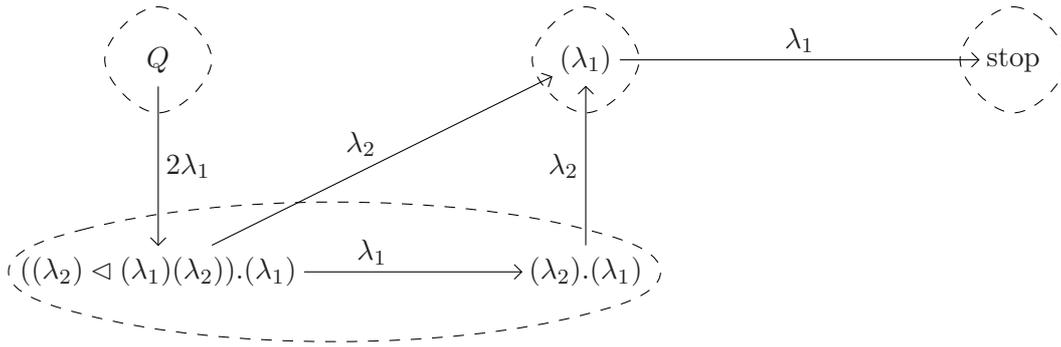


Figure 6.6: The MLTS of $Q = (2\lambda_1).((\lambda_2) \triangleleft (\lambda_1)(\lambda_2)).(\lambda_1)$

Lemma 6.18. *Each of strong bisimilarity \sim and weak bisimilarity \approx is a congruence with respect to all operators of language \mathcal{L} .*

A proof sketch of this lemma is available in Appendix B.4.

The congruent nature of both weak and strong bisimilarities is important, for it enables us to substitute a process with an equivalent one during compositions of processes. The substitution is useful if, for instance, the replacing process possesses some “better” properties—say, having simpler structure or having smaller set of reachable processes—than the replaced one.

We have mentioned that strong and weak-bisimulation relations are closely related to Markovian strong and weak-bisimulation relations as defined in [Buc94, Hil96, Bra02]. This is straightforward from the fact that the semantics of CCC processes is absorbing Markov chains. Our definitions extend the traditional ones by explicitly providing a different handling of the absorbing state.

6.3.2 PH-Equivalence

In this section, we define a new notion of equivalence among CCC processes based on the PH distributions they represent. We will also clarify the relationship between this notion of equivalence and the previously defined weak and strong bisimilarities.

Definition 6.19. *Two processes $P, Q \in \mathcal{L}$ are PH-equivalent (denoted by $P \approx_{\text{PH}} Q$) if and only if*

$$\text{PH}(P) = \text{PH}(Q). \quad (6.7)$$

Example 6.20. *Consider process P in Example 6.15 and process Q in Example 6.17. Once we obtain the absorbing CTMC semantics of both processes, we can use Algorithm 3.13 to obtain the canonical representations of $\text{PH}(P)$ and $\text{PH}(Q)$, and, at the same time, to reduce the size of the representations. The algorithm produces the same Cox representation for both, hence $\text{PH}(P) = \text{PH}(Q)$, and therefore $P \approx_{\text{PH}} Q$. Assuming that $\lambda_2 < \lambda_1$, Figure 6.7 depicts the Cox representation we obtain.*

PH-equivalence is also a congruence, as shown in the following lemma. Thus we are allowed to interchange processes that are equivalent during compositions of processes. This is especially important, because, then, we can use the reduction algorithm of

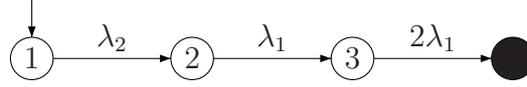


Figure 6.7: Minimized Canonical Representation of $\text{PH}(P)$ and $\text{PH}(Q)$

APH representations in every step of the composition to keep the size of intermediate processes almost surely minimal. We will discuss this in greater detail in the next section.

Lemma 6.21. *PH-equivalence (\approx_{PH}) is a congruence with respect to all operators of language \mathcal{L} .*

Proof. That \approx_{PH} is a congruence with respect to sequential, choice, and parallel operators is a straightforward consequence of Definition 6.19 and Lemma 6.12.

We are left with the disabling operator: we show that $P_1 \approx_{\text{PH}} P_2$ implies $(\mu) \triangleleft (\lambda)P_1 \approx_{\text{PH}} (\mu) \triangleleft (\lambda)P_2$

For processes $P_1, P_2 \in \mathcal{L}$, let $F_1(t)$ and $F_2(t)$ be the distribution functions of $\text{PH}(P_1)$ and $\text{PH}(P_2)$, respectively. Assuming that $P_1 \approx_{\text{PH}} P_2$, $\text{PH}(P_1) = \text{PH}(P_2)$ and $F_1(t) = F_2(t)$, for all $t \in \mathbb{R}_{\geq 0}$.

Let $\nu = \lambda + \mu$, and let $G(t)$ be the distribution function of the exponential distribution with rate ν .

From Equation (6.3), the distribution functions of $\text{PH}((\mu) \triangleleft (\lambda)P_1)$ and $\text{PH}((\mu) \triangleleft (\lambda)P_2)$ are given by $\frac{\mu}{\nu}G(t) + \frac{\lambda}{\nu}[G * F_1](t)$ and $\frac{\mu}{\nu}G(t) + \frac{\lambda}{\nu}[G * F_2](t)$, respectively. Since $F_1(t) = F_2(t)$ for all $t \in \mathbb{R}_{\geq 0}$, $\frac{\mu}{\nu}G(t) + \frac{\lambda}{\nu}[G * F_1](t) = \frac{\mu}{\nu}G(t) + \frac{\lambda}{\nu}[G * F_2](t)$ for all $t \in \mathbb{R}_{\geq 0}$.

Therefore, $\text{PH}((\mu) \triangleleft (\lambda)P_1) = \text{PH}((\mu) \triangleleft (\lambda)P_2)$ and $(\mu) \triangleleft (\lambda)P_1 \approx_{\text{PH}} (\mu) \triangleleft (\lambda)P_2$. \square

We have defined three notions of equivalence for CCC processes. The interrelation between the three notions is expressed in the following lemma.

Lemma 6.22. $\sim \subset \approx \subset \approx_{\text{PH}}$.

Proof. That $\sim \subset \approx$ follows directly from the definitions of \sim and \approx .

In the rest of the proof, we show that $\approx \subset \approx_{\text{PH}}$, namely for all $P, Q \in \mathcal{L}$, $P \approx Q$ implies $P \approx_{\text{PH}} Q$. Assume that $P \approx Q$, and let $(\vec{\alpha}, \mathbf{A})$ and $(\vec{\beta}, \mathbf{B})$ be the APH representations associated with \mathcal{M}_P and \mathcal{M}_Q , respectively.

The conditions we specify in the definition of weak bisimulation (and hence its impacts on the underlying CTMC semantics) can be shown to match the definition of lumpability [KS76, Buc94, Hil96]. This together with Lemma 3.17 imply $\text{PH}(\vec{\alpha}, \mathbf{A}) = \text{PH}(\vec{\beta}, \mathbf{B})$. Therefore $P \approx_{\text{PH}} Q$. \square

Figure 6.8 illustrates the fact that PH-equivalence is not a congruence with respect to the traditional choice operator. Let \pm denote the traditional choice operator. Since $\text{PH}(P) = \text{PH}(P')$, $P \approx_{\text{PH}} P'$. However, from the figure it is evident that $\text{PH}(P \pm Q) \neq \text{PH}(P' \pm Q)$. Therefore $(P \pm Q) \not\approx_{\text{PH}} (P' \pm Q)$.

6.4 Equivalence Checking and Process Reduction

In the previous section, we have introduced three notions of equivalence for CCC processes. In this section, we describe several methods to determine whether two CCC processes are equivalent.

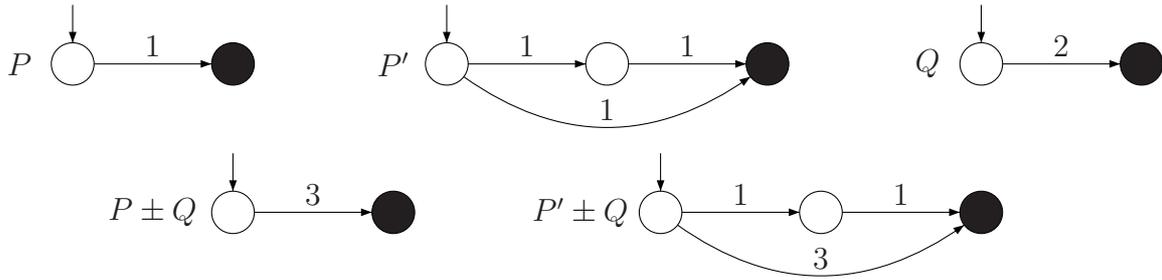


Figure 6.8: PH-equivalence and the Traditional Choice Operator

6.4.1 Algorithmic Considerations

Since the strong and weak-bisimulation relations we described in the previous section are slight variations of the Markovian strong and weak-bisimulation relations, respectively, we can use existing algorithms for checking Markovian strong and weak-bisimulations to decide strong and weak bisimilarities on CCC processes with minor changes.

Checking strong bisimulation on a CTMC can be carried out by an algorithm with time complexity $\mathcal{O}(m \log n)$ [DHS03], where m is the number of transitions, and n is the number of states of the CTMC. The algorithm is based on the partition-refinement approach described in [PT87]. The same partition-refinement-based algorithm can also be used to check weak bisimulation on a CTMC [DHS03], when it is coupled with transitive-closure computation. This algorithm is of time complexity $\mathcal{O}(n^3)$, where n is the number of states of the CTMC. Each of these algorithms computes the largest Markovian bisimulation, in the sense that it produces an optimal aggregation from the possible lumpings of the given CTMC.

New algorithms for checking strong and weak bisimulations on interactive Markov chains (IMC) [Her02] that exploit the acyclicity of the models are presented in [CHZ08]. These algorithms can also be used to check Markovian strong and weak bisimulations. Assuming that the given CTMC is acyclic, the algorithm for checking strong bisimulation is of time complexity $\mathcal{O}(m)$, where m is the number of transitions of the CTMC. The theoretical time complexity of the new algorithm for checking weak bisimulation on an acyclic CTMC remains cubic in the number of states. In practice, however, the new algorithm achieves a significant speed-up.

For the case of PH-equivalence, the following lemma provides a mechanism to determine whether two processes $P, Q \in \mathcal{L}$ are PH-equivalent.

Lemma 6.23 ([Wol08]). *Let $(\vec{\alpha}, \mathbf{A})$ and $(\vec{\beta}, \mathbf{B})$ be two phase-type representations of size k and respectively l , where $k \geq l$. $\text{PH}(\vec{\alpha}, \mathbf{A}) = \text{PH}(\vec{\beta}, \mathbf{B})$ if and only if their first $2k$ moments agree.*

The lemma establishes that in order to determine whether two PH representations have the same PH distribution, we just have to check whether their first $2k$ moments are pair-wise equal, where k is the larger size of the two PH representations. Therefore, given two processes $P, Q \in \mathcal{L}$, where $|\text{Reach}(P)| \geq |\text{Reach}(Q)|$, their PH-equivalence can be ascertained by obtaining their absorbing CTMC semantics \mathcal{M}_P and \mathcal{M}_Q , and then comparing the first $2(|\text{Reach}(P)| - 1)$ moments of their associated APH representations.

The i -th moment of a PH representation $(\vec{\alpha}, \mathbf{A})$ (cf. Section 2.5.1) is

$$m_i = (-1)^i i! \vec{\alpha} \mathbf{A}^{-i} \vec{e}.$$

Thus, to compute the i -th moment, we need a matrix inversion and followed by $i - 1$ vector-matrix multiplications. Since matrix inversion can be unstable, it is best to avoid it. As described in [Wol08], the first i moments can be instead obtained as follows. The first moment can be obtained by solving the system of equations

$$\vec{\gamma}^{[1]} \mathbf{A} = -\vec{\alpha},$$

and the moment is $m_1 = \vec{\gamma}^{[1]} \vec{e}$. Then, for $2 \leq j \leq i$, we iteratively solve the system of equations

$$\vec{\gamma}^{[j]} \mathbf{A} = -j \vec{\gamma}^{[j-1]},$$

and the corresponding moment is $m_j = \vec{\gamma}^{[j]} \vec{e}$.

Since we are dealing with acyclic PH representations, PH-generator \mathbf{A} is an upper-triangular matrix, and to solve the above systems of equations we only need to apply backward substitution, which is of time complexity $\mathcal{O}(k^2)$. Therefore, to determine whether two APH representations are PH-equivalent is of time complexity $\mathcal{O}(k^3)$.

6.4.2 Compositional Considerations

One of the main practical advantages of a notion of equivalence is that it may provide us with a mechanism to aggregate models while still retaining behaviors that we are interested in. A congruent notion of equivalence, moreover, is more desirable, for it also enables us to substitute a model with an equivalent one during compositions of models. By using an aggregated model instead of the original one during model compositions, we can keep the size of the resulting model small. PH-equivalence is a congruent notion of equivalence on CCC processes.

Given a process $P \in \mathcal{L}$, Algorithm 3.13 can be used to reduce the state space of its associated APH representation. The algorithm works on the corresponding ordered bidiagonal representation. Once the reduced representation is obtained, we can transform it to its Cox representation, and the procedure described in the proof of Lemma 6.11 can be utilized to obtain a process $\text{Red}(P) \in \mathcal{L}$, which is called the reduced process of P . Now, the following lemma is straightforward.

Lemma 6.24. *For all processes $P, Q \in \mathcal{L}$:*

1. $\text{PH}(\text{Red}(P)) = \text{PH}(P)$,
2. $\text{PH}(\text{Red}(P.Q)) = \text{PH}(P.Q)$,
3. $\text{PH}(\text{Red}(P + Q)) = \text{PH}(P + Q)$, and
4. $\text{PH}(\text{Red}(P \parallel Q)) = \text{PH}(P \parallel Q)$.

Hence, the reduction algorithm can be used to reduce the size of processes in each composition step. In Chapter 5, we have shown that given APH representations whose APH distributions are triangular ideal (*i.e.*, PH distributions whose triangular order is equal to their algebraic degree), the application of the convolution, minimum, or maximum operations on them when accompanied by the reduction algorithm produces

representations that are almost surely minimal. Therefore, the reduction algorithm can be used to keep the processes obtained after sequential, choice, and parallel compositions almost surely minimal.

6.5 Conclusion

Our aim was to develop a stochastic process calculus that captures the mechanism for generating and manipulating APH representations. We have built a stochastic process calculus CCC by specifying the syntax and semantics of a language with four operators: the disabling (\triangleleft), sequential (\cdot), choice ($+$), and parallel (\parallel) operators.

For any arbitrary APH representation, we showed that the disabling operator together with the basic processes are enough to generate a process that describes the representation. The APH distribution of the representation then describes the completion time of the process. Theorem 2.30, on the other hand, states that the whole set of APH distributions can also be generated from exponential distributions together with convolution and finite mixture operations. The finite mixture operation can be accomplished by the traditional choice operation, namely the operation that selects one from several processes based on the set of outgoing transitions from the processes. This operation is straightforward. Moreover, the results of the minimum and maximum operations can be flattened onto this operation. However, as shown in Section 6.3, with respect to this traditional choice operator, PH-equivalence is not a congruence.

The three operations on APH representations that we proposed in the previous chapter, *i.e.*, convolution, minimum, and maximum operations, were shown to correspond exactly to the sequential (\cdot), choice ($+$), and parallel (\parallel) operators, respectively.

On processes generated by the calculus, we defined three notions of equivalence. A notion of equivalence defines the circumstances in which we can deem two CCC processes to be equivalent in their behaviors. Two of the notions, namely strong and weak-bisimulation equivalences, are more or less standard, and we have defined them in a similar way they are usually defined in other stochastic process calculi. These two notions identify processes by their branching structures. The third notion, PH-equivalence, on the other hand, identifies processes on a more lower level, namely on the distributions of their completion times. We have shown that PH-equivalence is coarser than both strong and weak bisimulation.

We have shown that each of these three notions is a congruence with respect to all operators of the language of CCC. This means that we are allowed to substitute a process with an equivalent one during the composition of processes. These substitutions can be useful if the replacing process is an aggregation of—and hence is smaller in size than—the replaced one.

In Chapter 5, we have stated that if we restrict ourselves to use only APH distributions whose triangular order is equal to their algebraic degree in a stochastic modelling formalism that is equipped with the three operations, we will only deal with models that are almost surely minimal. Overall, this is the main result of the chapter: the congruent nature of the PH-equivalence and the almost surely minimal property of the three operations implies that we have developed a stochastic process calculus that always defines processes that are almost surely minimal.

Chapter 7

Case Studies

We implemented the reduction algorithm described in Algorithm 3.13 in C++ together with the three operations defined on APH representations. The implementation allows for transformations from one APH canonical form to the others. In order to handle the sensitivity of the initial probability distributions of ordered bidiagonal representations in relation to the APH distributions they represent, we resorted to rational arithmetic using the GNU Multiple Precision Arithmetic (GMP) library [GMP08] in the implementation.

Contribution To demonstrate the practical potential of the reduction algorithm and the three operations, we present three case studies in this chapter. In the first case study, we extend the set of Markovian probability distributions used as basic events in static fault trees to cover the whole set of APH distributions. At the same time, we observe that in practice the exponential growth of minimal representations of the maximum of Erlang distributions is manageable when the number of components is small. In the second, we study several models of fault-tolerant parallel processors by analyzing their reliability using dynamic fault trees. We show order-of-magnitude reductions achieved by the algorithm on these models. In the third case study, we show how the reduction algorithm can be useful in an analysis of stochastic delay propagations in a railway network.

Related Work In the first case study, the 3P2M model is a slight modification of a model with the same name first proposed in [DD96]. In the second case study, the FTTP model was first described in [HLD88]. In [DBB92], the model was analyzed using dynamic fault trees. The authors of [BCS07a] used the same model to demonstrate the use of their compositional method to produce CTMC models of dynamic fault trees. In the third case study, the original model of delay propagations in the railway network, specifically the model derived from the Dutch intercity railway network, was presented in [GyK00]. In [MM07], PH distributions were used in the analysis of the model. The authors of the paper showed that operations on the stochastic models of the delay propagation correspond to standard operations on PH distributions.

Structure The chapter is organized as follows: Sections 7.1, 7.2, and 7.3 present the first, second, and third case studies, respectively. In Section 7.4, we summarize and conclude the chapter.

7.1 Fault Trees with PH Distributions

In this section, we analyze the reliability of a processors-and-memories (3P2M) system [DD96] by using static fault trees [HK92, DBB93]. Figure 7.1 depicts a high-level model of the 3P2M system. The system consists of three identical processors (P1–P3) and two identical memory modules (M1 and M2), all connected by a bus (B). The processors and the memory modules work independently of each other, and they, together with the bus, over time fail. The failure of each component is governed by a continuous probability distribution. The system fails when either all processors fail or all memory modules fail or the bus fails.

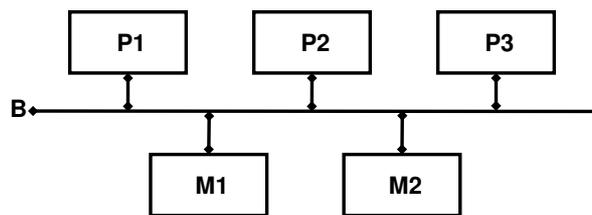


Figure 7.1: Three Processors, Two Memories, and a Bus (3P2M) Model

The fault tree model of the 3P2M system is depicted in Figure 7.2. A fault tree model consists of several basic events and gates. A basic event represents the failure of some basic, indivisible component, while a gate represents and determines the relationship and interdependency between several basic events.

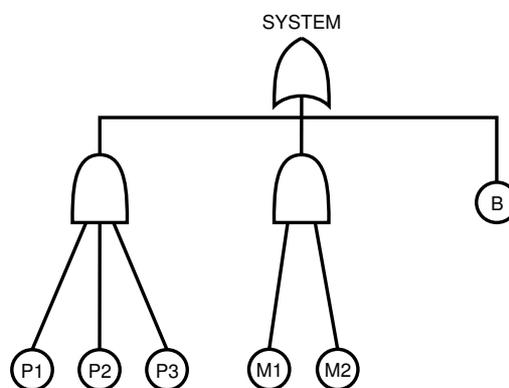


Figure 7.2: The Fault-Tree of the 3P2M Model

In the standard or static fault trees, there are three types of gates: OR, AND, and VOTING gates. The OR and AND gates are the standard logic gates, and they are depicted in the same way. An instance of the VOTING gate is depicted in Figure 7.6(a). A VOTING gate is an AND gate with k/m inscription on it. A k/m VOTING gate fails when at least k out of its m inputs fail.

In Section 3.5, we have demonstrated the use of the reduction algorithm in minimizing the CTMCs obtained from fault tree models when all basic events fail according to exponential distributions. In this section, we illustrate the possibility of employing more complex Markovian distributions to govern the basic events. The possibility

arises from the fact that OR and AND gates in the fault trees correspond exactly to the minimum and respectively maximum operators we have discussed in the previous chapters. Hence, they also correspond to the choice (+) and respectively the parallel (||) operators of CCC.

Let P_i denote a CCC process describing the time to failure of processor P_i , for $i = 1, 2, 3$. Further, let M_i be a CCC process describing the time to failure of memory module M_i , for $i = 1, 2$, and similarly with B , the process describing the time to failure of the bus B . Then the fault tree model in Figure 7.2 can be expressed by CCC process

$$(P_1 || P_2 || P_3) + (M_1 || M_2) + B. \quad (7.1)$$

To analyze the reliability of the 3P2M model, we conduct several experiments. In each experiment, the time to failure of each component is governed by Erlang distributions of a particular phase (number of states). The mean value of the Erlang distributions governing each component, however, are kept the same in all experiments, which means that the rates must be adjusted accordingly. The mean failure times of each processor, each memory, and the bus are set to 5, 3, and 7 years, respectively. Table 7.1 lists the parameters of the Erlang distributions used in the experiments.

Table 7.1: Erlang Distributions Used in the Experiments

Phase	Processors	Memories	Bus
1	$\text{Exp}(\frac{1}{5})$	$\text{Exp}(\frac{1}{3})$	$\text{Exp}(\frac{1}{7})$
5	$\text{Erl}(\frac{5}{5}, 5)$	$\text{Erl}(\frac{5}{3}, 5)$	$\text{Erl}(\frac{5}{7}, 5)$
10	$\text{Erl}(\frac{10}{5}, 10)$	$\text{Erl}(\frac{10}{3}, 10)$	$\text{Erl}(\frac{10}{7}, 10)$
20	$\text{Erl}(\frac{20}{5}, 20)$	$\text{Erl}(\frac{20}{3}, 20)$	$\text{Erl}(\frac{20}{7}, 20)$
50	$\text{Erl}(\frac{50}{5}, 50)$	$\text{Erl}(\frac{50}{3}, 30)$	$\text{Erl}(\frac{50}{7}, 50)$
100	$\text{Erl}(\frac{100}{5}, 100)$	$\text{Erl}(\frac{100}{3}, 100)$	$\text{Erl}(\frac{100}{7}, 100)$

Table 7.2 summarizes the result of the experiments. We have six 3P2M models, where we vary the phases of the Erlang distributions governing the basic events, ranging from 1, which corresponds to exponential distributions, to 100. The second column of the table (Original) describes the number of states in the resulting CTMC models when they are generated without any size reduction whatsoever. The third column (Inter.) corresponds to the number of states in the largest intermediate CTMC models the reduction algorithm encounters while minimizing each of the six 3P2M models. Recall that after carrying out each operation in Equation (7.1), the reduction algorithm can be used to reduce the resulting intermediate model. In this way, the size of the results of subsequent operations can be kept small. The state spaces shown in the column are usually the intermediate results prior to the last operation, which in this case is the last choice operation. The fourth column (Final) corresponds to the number of states in the final CTMC models. Compared to the original state spaces, the size of the final state spaces is orders-of-magnitude smaller. While the size of an original model grows multiplicatively in the sizes of its components, the size of a reduced representation grows additively in the sizes of its components.

The computation times (in seconds) for transforming, reducing, and composing the models are shown in the last three columns of Table 7.2. In the “SPA” column,

Table 7.2: State Space Reduction of the 3P2M Models

Phase	Original	Inter.	Final	Comp. Time (sec.)		
				SPA	Red.	Proc.
1	21	6	6	<1	<1	<1
5	37625	450	114	<1	<1	<1
10	1596000	1950	249	4	18	<1
20	81488000	8100	519	35	265	1
50	$>1.72 \cdot 10^{10}$	51750	1329	10999	800	16
100	$>1.05 \cdot 10^{12}$	208500	2679	219180	10086	98

we provide the computation time spent to transform the models to ordered bidiagonal representations. The “Red.” column describes the computation time spent by the reduction algorithm in reducing the models, namely eliminating the removable states once SPA is completed. The rest of the computation time is expended in other processing, such as building all components, carrying out the minimum and maximum operations, and storing the results in files. This is listed in the column “Proc.”. Compared to the SPA and reduction times, the processing time in most cases is negligible. On the other hand, SPA consumes most of the computation time. The computation times required by both SPA and reduction grow extremely fast—faster than $\mathcal{O}(n^3)$. This is due to the fact that the implementation uses rational numbers. The storage requirement of the used rational numbers grows in size over time, and hence also in their processing time.

With the current implementation of the reduction algorithm, we have seen the limit of the scalability of the algorithm in dealing with 3P2M models or similar models that involve the minimum and maximum of highly structured APH representations. The computation time of SPA for the largest model in Table 7.2 is already more than 60 hours. As a rule of thumb, the algorithm should only be used when the largest intermediate model is no more than around 300000 states.

Figure 7.3 depicts the distributions of the time (in years) to failure in the six 3P2M models. The reliability of the system is actually not improved by introducing more states in each component’s Erlang distributions. Instead, the failure time becomes more “precise” as the probability mass increases in the range of 3 to 4 years, and approaching 3 years as the number of states becomes larger. This is to be expected, as each memory module fails with mean time 3 years, and the top level event of the fault tree is an OR gate, which corresponds to the minimum operation.

As a concluding remark, we would like to reemphasize the fact that minimal representations of the maximum of Erlang distributions grow exponentially with the number of components. As we have observed in the case study, for a particular number of components, the minimal representation grows additively in the sizes of the components. Hence, when the number of components is small, the size of the resulting minimal representations is actually manageable.

The 3P2M model we use is a modification of the original one, in which we replace a 2/3 VOTING gate on the processors by an AND gate. A VOTING gate can actually always be represented by a combination of AND and OR gates. In the case of

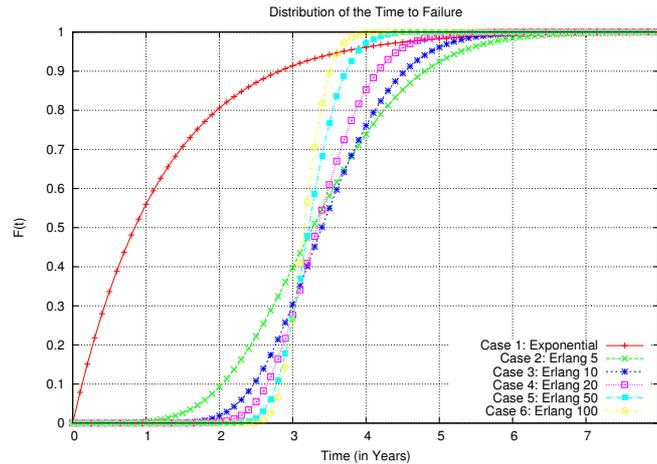


Figure 7.3: The Distribution of the Time to Failure of the 3P2M Models

a 2/3 VOTING gate on three processors, an equivalent representation is depicted in Figure 7.4.

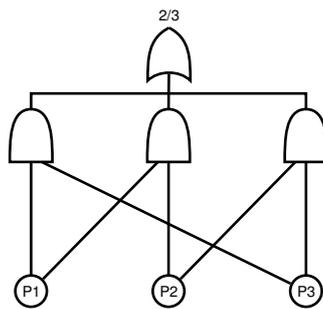


Figure 7.4: Another Fault-Tree Representation of 2/3 VOTING Gate

The reason why we modified the original model is because a VOTING gate or even this new representation cannot be accomplished by our three operations. $P = (P_1 \parallel P_2) + (P_1 \parallel P_3) + (P_2 \parallel P_3)$, for instance, is not a correct CCC process to describe a 2/3 VOTING gate on three processors. This is because there is no dependency between any two similarly named sub-processes in process P , while a dependency exists in the basic events of the VOTING gate. The two sub-processes P_1 in P have the same behavior, but they are independent of each other, while the basic event feeding the first and the second AND gates in Figure 7.4 is the same event, P_1 .

7.2 Fault-Tolerant Parallel Processors

In this section, we analyze the reliability of several fault-tolerant parallel processors (FTPP) systems by using dynamic fault trees (DFT) [MDCS98, BCS07b, BCS07a]. Several variations of FTPP are presented in [DBB92]. Figure 7.5 depicts one of the variants. An FTPP consists of a number of network elements (NE), which are fully connected to each other. There are the same number of groups of processors as the number of NES. In the figure we have four NES and four groups of processors: T1, T2, T3, and T4. In

each of these groups, one processor—that on NE4, namely the processor with subscript s —is originally powered down and is used as a spare unit.

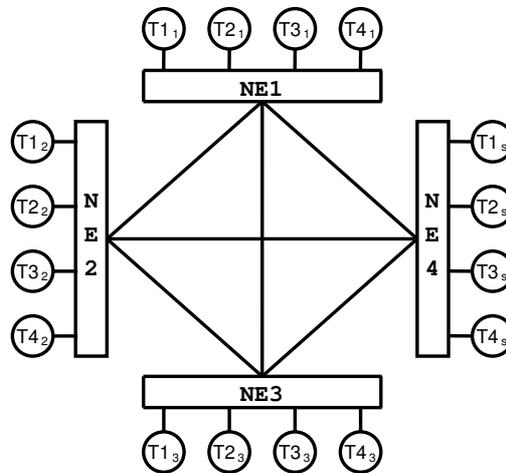


Figure 7.5: A Fault-Tolerant Parallel Processors (FTPP) Model

Each NE is attached to the same number of processors as the number of groups. Overall in the system, there are as many processors in a single group as the number of NES. At least two processors per group are required to be operational for the whole system to be considered operational, otherwise the system is considered failed. Each processor and each NE may experience failures that are governed by exponential distributions. Furthermore a failed NE brings down all the processors connected to it.

In this case study, several FTPP systems obtained by varying the number of NES—and hence also the number of groups of processors—are modelled by using dynamic fault trees. Dynamic fault trees extend the static fault trees with several additional gates. The dynamic fault tree model of FTPP with 4 NES—based on [DBB92]—is depicted in Figure 7.7. Aside from static gates, namely an OR and four VOTING gates, the dynamic fault-tree model contains two other types of gates, from which the “dynamic” part arises. These are the SPARE gate (Figure 7.6(b)) and the functional dependency (FDEP) gate (Figure 7.6(c)).

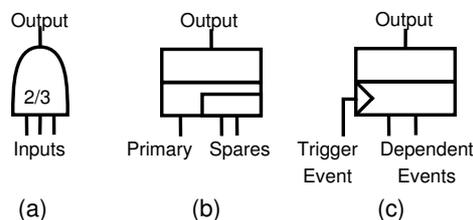


Figure 7.6: VOTING, SPARE, and FDEP Gates

A SPARE gate consists of a single *primary input* and several other inputs called *spares*. The primary input is up from the beginning, while the spares are in stand-by mode. Upon the failure of the primary input, one of the available spares is activated, and it replaces the primary input. If this replacing spare fails, another available spare will replace it, and so on, until all spares are exhausted, at which time the gate itself

fails. All inputs to a SPARE gate must be basic events. A FDEP gates consists of a *trigger input* and a set of *dependent inputs*. The trigger input is connected to a trigger event. A dependent input is connected to the basic event of a component. When the trigger event occurs, all dependent components of the gate become inaccessible and unusable. The output of a FDEP gate is not used in the analysis of a dynamic fault tree model; what is important is that the gate imposes some dependencies among its dependent components.

Once the dynamic fault-tree models are built, we use a tool called CORAL [BCS07a] to obtain the corresponding CTMC models. CORAL has a formal semantics that maps dynamic fault trees to I/O-IMCs [BCS07a]. In many cases, including ours, the obtained I/O-IMCs can be converted to CTMCs. In all models, the failures of each NE and each processor are governed by exponential distributions with rates $1.7 \cdot 10^{-5}$ per hour and $1.1 \cdot 10^{-4}$ per hour, respectively. The spare components connected to a SPARE gate are *cold spares*, *i.e.*, they do not degrade while in stand-by mode. Each of the obtained CTMC models describes the distribution of the time to the system's failure.

The obtained CTMC models represent PH distributions, and they are all APH representations. We use Algorithm 3.13 to reduce the state spaces of the APH representations. Table 7.3 summarizes the result of the experiment. We have three FTTP models. For each model, the table provides the size of the state spaces before (Original) and after (Final) the reduction algorithm is applied. Note that the state spaces have been minimized by a Markovian weak-bisimulation-based lumping prior to the reduction algorithm.

Table 7.3: State-Space Reduction of the FTTP Models

#NE	Original		Final		Comp. Time (sec.)	
	States	Transitions	States	Transitions	SPA	Reduction
4	64	304	20	39	1	<1
5	390	2956	40	76	92	13
6	1727	17211	63	121	18062	944

In general, the reduction algorithm produces state spaces, which are orders-of-magnitude smaller than the original ones. The models prior to the reduction algorithm are obtained by composing smaller components. During the composition, the size of the intermediate representations blows up fast, because they are the cross products of the components' representations. However, when the smaller components are stochastically similar, the composition alters the representations drastically, while the stochastic behaviors actually remain the same or are altered only slightly. In these circumstances, we should normally find many duplicated states with similar total outgoing rates, in which case the reduction algorithm should perform well.

The computation time (in seconds) required for the transformation to ordered bidiagonal representations by the spectral polynomial algorithm prior to the actual reduction is shown in the column before the last (SPA). The computation time of the actual reduction (Reduction) is shown in the last column of the table. It is evident from the table that, as in the previous case study, SPA consumes most of the computation time in the reduction procedure.

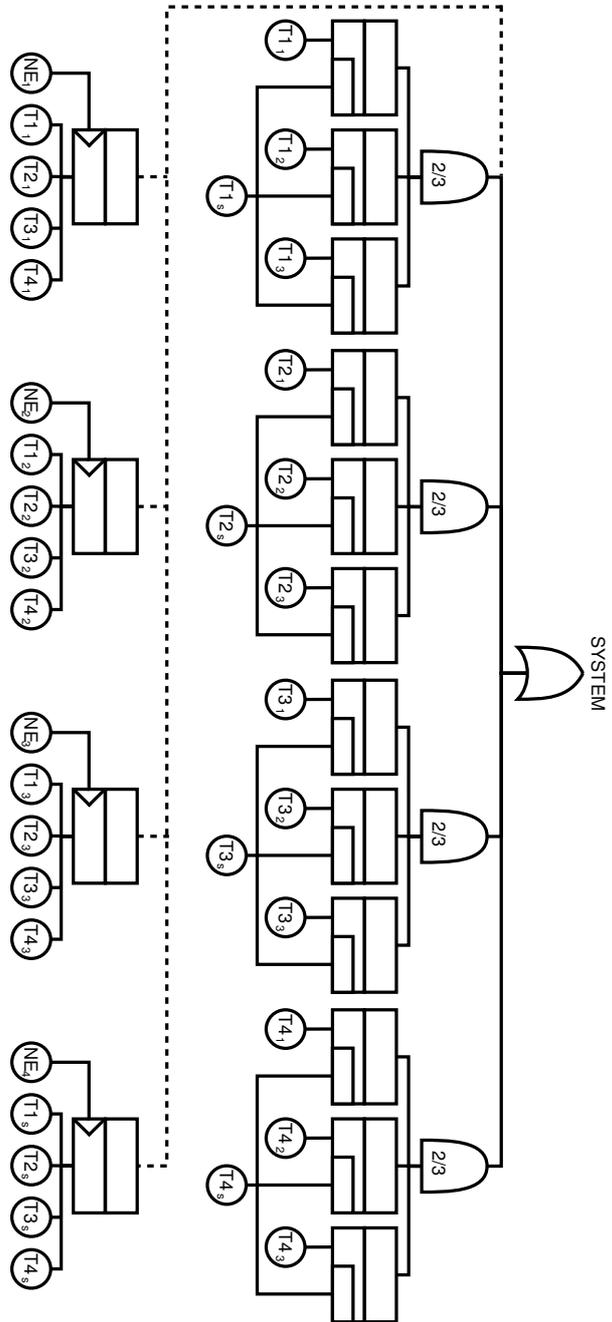


Figure 7.7: The Dynamic Fault-Tree Representation of the FTTP Model

We only analyze FTTP models having up to 6 NES in this case study. This is because to obtain CTMC models for FTTPs with more NES by using CORAL requires an enormous amount of computation time. However, observing the size of the reduced representations and the required computation time in Table 7.3, we expect that the algorithm would be able to reduce FTTP models with larger number of NES in some reasonable computation time, if the CTMC models were available.

Figure 7.8 depicts the distributions of the time (in hours) to failure in the three FTTP models. The reliability of a system can be improved by introducing more redundancy to the system, for instance, in our FTTP case, by adding more NES or more processors. As can be witnessed in the figure, indeed the more NES an FTTP system has, the slower it is failing.

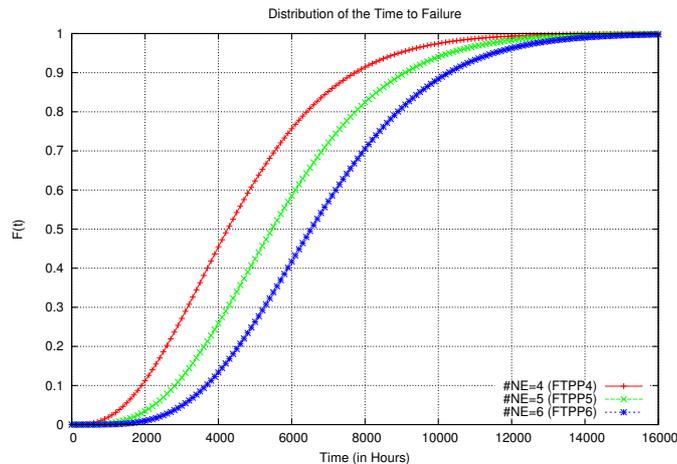


Figure 7.8: The Distribution of the Time to Failure of the FTTP Models

7.3 Delay in a Railway Network

In this section, we analyze the propagation of delays (deviations from the scheduled times) in a railway network by modelling the network in CCC. The model is based on the railway network model of [MM07]. All non-stochastic parameters of the network model are exactly as they are specified there. Most of these parameters are provided in Figure 7.9, Table 7.4, and Table 7.5. The network model of that paper itself is an adaptation of the model of [GyK00].

Figure 7.9 depicts a part of the Netherlands intercity (long-distance) railway network that connects 10 cities, represented by the ovals. The network consists of four lines, numbered from 1 to 4. In the figure, an oval is labelled with the identity of the line(s) passing through the city it represents. Line 1, for instance, goes through Amsterdam, Amersfoort, Zwolle and Groningen.

Each line is divided into several segments. A segment is the part of a line between two connected cities. Since the whole network is double track, there are two segments between two connected cities, one for each direction. Columns 1–4 of Table 7.4 provide the information of the lines (Line), the segments (Seg.), the origin city (Orig.) of the segments, and the destination city (Dest.) of the segments, respectively. From the table, we see that line 1, for instance, consists of 6 segments: segment 01 from Amsterdam to Amersfoort, segment 02 from Amersfoort to Zwolle, segment 03 from Zwolle

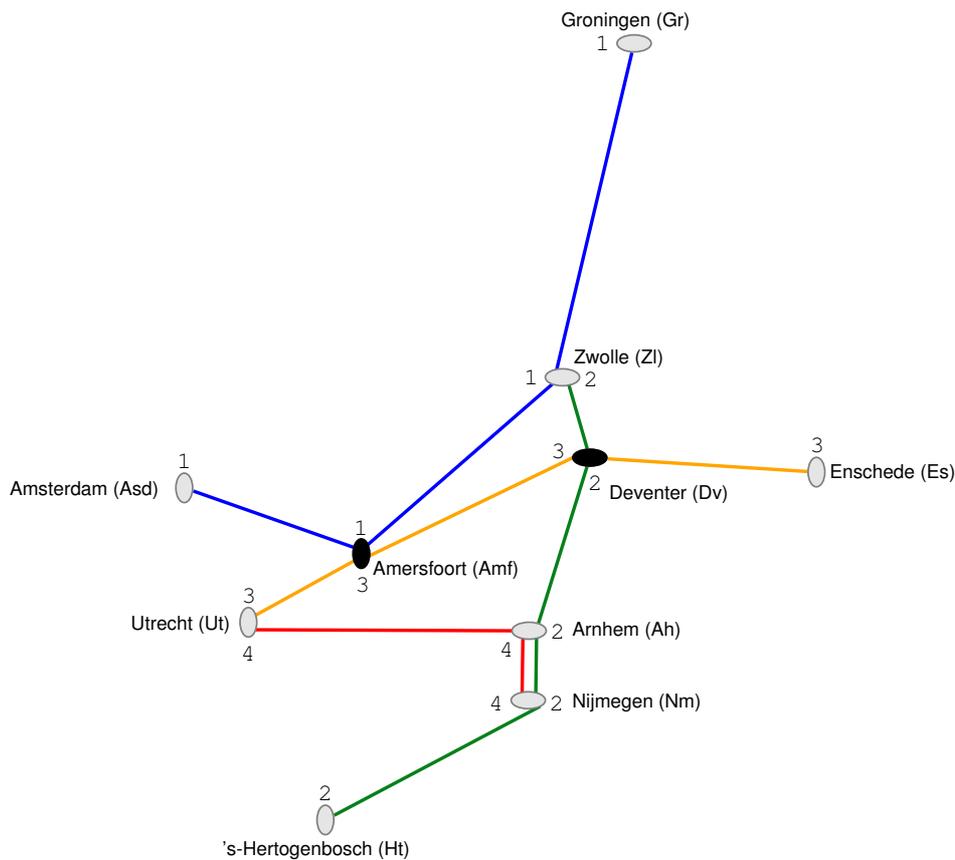


Figure 7.9: A Part of the Netherlands Intercity Railway Network

to Groningen; and on the opposite direction, segment 51 from Groningen to Zwolle, segment 52 from Zwolle to Amersfoort, segment 53 from Amersfoort to Amsterdam.

The fixed schedule of the trains passing through the segments is shown in Table 7.4 columns 5–8. Column 5 (Dep.) lists the departure time of trains from the origin cities, showing the minute after the hour. Column 6 (Run.) lists the running time (in minutes), namely the time it takes for trains to travel from the origin to destination cities. Column 7 (Stop.) lists the least amount of time (in minutes) trains must stop in the destination cities before departing to the next cities in the lines. Column 8 (Buff.) lists some additional buffer time (in minutes) trains may stop in the destination cities before departing to the next cities.

As an example, take line 1 segment 01. A train is expected to depart from Amsterdam at 34 minutes after the hour; the time it takes to travel to the destination city Amersfoort is 32 minutes; in Amersfoort, before continuing the trip to Zwolle, the train must stop for at least 2 minutes with a—possibly additional—buffer time of at most 2 minutes. In some segments, for instance segment 03 of line 1, the buffer times are large. These are for shielding the return trips on the corresponding lines from delays caused by late arrivals.

In Table 7.5, the synchronizations between several segments of different lines for the purpose of transfer connections are listed. During a synchronization, a train arriving at the destination city of a feeder segment (columns 1–2) may have passengers

Table 7.4: Information on Lines and Segments in the Railway Network

Line	Seg.	Orig.	Dest.	Dep.	Run.	Stop.	Buff.	t	r
1	01	Asd	Amf	34	32	2	2	29	5
1	02	Amf	Zl	10	36	2	1	32	5
1	03	Zl	Gr	49	65	5	39	59	11
1	51	Gr	Zl	38	66	2	2	59	9
1	52	Zl	Amf	48	36	2	2	32	6
1	53	Amf	Asd	28	31	5	5	28	8
2	01	Zl	Dv	23	19	2	0	17	2
2	02	Dv	Ah	44	35	2	0	32	3
2	03	Ah	Nm	21	15	2	0	14	1
2	04	Nm	Ht	38	29	5	14	26	8
2	51	Ht	Nm	26	28	2	0	25	3
2	52	Nm	Ah	56	14	2	0	13	1
2	53	Ah	Dv	12	35	2	0	32	3
2	54	Dv	Zl	49	20	5	9	18	7
3	01	Ut	Amf	52	14	2	2	13	3
3	02	Amf	Dv	10	37	2	0	33	4
3	03	Dv	Es	49	43	5	21	39	9
3	51	Es	Dv	58	43	2	1	39	5
3	52	Dv	Amf	44	40	2	1	36	5
3	53	Amf	Ut	27	15	5	5	14	6
4	01	Ut	Ah	20	34	2	3	31	6
4	02	Ah	Nm	59	12	5	5	11	6
4	51	Nm	Ah	21	13	2	3	12	4
4	52	Ah	Ut	39	34	5	2	31	5

who need to transfer to a train departing from the origin city of a connecting segment (column 3–4). The last column of the table lists the time (in minutes) needed for the transfer. The listed time must be made available between the arrival of the feeder train to the departure of the connecting train. This time duration can overlap with the stop time if the feeder train arrives beforehand. As an example, from the fifth row of the table we conclude that a train arriving in Deventer from Enschede at segment 51 of line 3 feeds a transfer to the connecting train departing from Deventer to Zwolle at segment 54 of line 2 with a transfer time of 2 minutes.

Several synchronizations are two ways, *i.e.*, both trains are simultaneously a feeder and a connecting train. This is the case for trains travelling on lines 1 and 3 synchronizing in Amersfoort (row 1 and 4) and trains travelling on lines 3 and 1 synchronizing in Deventer (row 2 and 7).

The data we described so far specify the schedule that the trains travelling on the networks are expected to meet. In this sense, they are deterministic. In real life, however, the actual departure, running, and stopping times deviate from their scheduled times. If the actual running time of a train in some segment exceeds its scheduled running time, a *primary* delay occurs. Because of the interdependence of segments

Table 7.5: Synchronization in the Railway Network

Feeder		Connecting		Transfer Time
Line	Segment	Line	Segment	
1	01	3	02	2
1	52	3	53	2
2	01	3	03	2
3	01	1	02	2
3	51	2	54	2
3	51	2	02	2
3	52	1	53	2

on the same or different lines, this delay may be passed on and may cause delay in other segments, which is referred to as a *secondary* delay [MM07]. The purpose of the model is to analyze the propagation of delays by modelling the deviations from the scheduled times as random variables. In the following analysis, we follow closely the method of [MM07].

In the penultimate column of Table 7.4, the minimal running times (in minutes) of all segments in the railway network are listed. To each segment we assign a running-time distribution, which is described by $t+T$, where t is the segment's minimal running time and T is a continuous nonnegative random variable describing the actual time needed *in addition* to the minimal running time. The distributions of the random variables T are parameterized by the values r listed in the last column of the table. For a segment, the value of r is chosen such that it is constituted by the segment's buffer time and an additional (around) 10% of the minimal running time. Overall, $t+r$ should be equal to the sum of the expected running time and the expected buffer time for each segment, except for line 1 segment 03, line 2 segment 04, and line 3 segment 03, where the buffer times are shortened to 5 minutes. We can see in the table that there are 10 different values of r . In our models, then, we have 10 random variables T and each describes a basic additional running-time distribution.

Now that we have the running-time distribution of each segment in the network, we need to model the delay experienced by a train travelling on the segment. Given T and r , when $T-r > 0$, then the actual running time is more than the expected time specified in the schedule. In this case, a delay will occur. Let $D \geq 0$ be a random variable describing the departure delay at station A in the beginning of a segment. This delay is nonnegative, because even if a train arrives early at A, it will depart exactly on the scheduled time from A. The departure delay at station B in the beginning of the next segment on the line is given by

$$[D + T - r]^+,$$

where $[x]^+ = \max\{x, 0\}$. Note that the minimal running time and the stopping time play no role in the delay at all. In the case of synchronizations, let segment i be a feeder segment while segment j be a connecting segment, then the departure delay in the beginning of segment $i+1$ is

$$[D_i + T_i - r_i]^+,$$

while the departure delay in the beginning of segment j is

$$\max\{[D_i + T_i - r_i]^+, [D_j + T_j - r_j]^+\}.$$

In the experiments of this case study, all random variables will be governed by PH distributions. As we described in Section 2.5.5, the sum of two PH random variables $X + Y$ corresponds to the convolution of their PH distributions, and, in turn, to the convolution of their PH representations. The maximum of two PH random variables $\max\{X, Y\}$, on the other hand, corresponds to the maximum of their PH distributions. For the excess beyond a nonnegative number $[X - r]^+$, where X is a random variable governed by a PH distribution with representation $(\vec{\alpha}, \mathbf{A})$, we have

$$\Pr([X - r]^+ \geq t) = 1 - \vec{\alpha}e^{\mathbf{A}(t+r)}\vec{e} = 1 - \vec{\alpha}e^{\mathbf{A}r}e^{\mathbf{A}t}\vec{e}, \quad t \in \mathbb{R}_{\geq 0}.$$

Hence the distribution of $[X - r]^+$ is given by $\text{PH}(\vec{\alpha}e^{\mathbf{A}r}, \mathbf{A})$.

Let P_i be a CCC process describing the departure delay of segment i , and P'_i be a CCC process describing the arrival delay at the station in the end of segment i . Furthermore, for a CCC process P describing the PH distribution of a random variable X , let $[P]_r$ be the CCC process describing the PH distribution of $[X - r]^+$. Let S_r , for $r = 1, \dots, 9, 11$, denote the processes describing the basic running-time distributions, which correspond to the 10 different values of r (cf. the last column of Table 7.4). Table 7.6 lists the CCC process definitions of all the departure (and several arrival) delays of all segments in the railway network. The propagations of delays from one segment to the next on the line are accomplished by the sequential (\cdot) operator in CCC. The synchronizations of several segments, on the other hand, are accomplished by the parallel (\parallel) operator.

Table 7.6: CCC Process Definitions of the Delays

Line	Forward	Return
1	$P_{1-02} = [S_5]_5 \parallel [S_3]_3$ $P_{1-03} = [S_5.P_{1-02}]_5$ $P'_{1-03} = [S_{11}.P_{1-03}]_{11}$	$P_{1-52} = [S_9]_9$ $P_{1-53} = [S_6.P_{1-52}]_6 \parallel [S_5.P_{3-52}]_5$ $P'_{1-53} = [S_8.P_{1-52}]_8$
2	$P_{2-02} = [S_2]_2 \parallel P_{3-52}$ $P_{2-03} = [S_3.P_{2-02}]_3$ $P_{2-04} = [S_1.P_{2-03}]_1$ $P'_{2-04} = [S_8.P_{2-04}]_8$	$P_{2-52} = [S_3]_3$ $P_{2-53} = [S_1.P_{2-51}]_1$ $P_{2-54} = [S_3.P_{2-52}]_3 \parallel P_{3-52}$ $P'_{2-54} = [S_7.P_{2-53}]_7$
3	$P_{3-02} = [S_3]_3 \parallel [S_5]_5$ $P_{3-03} = [S_4.P_{3-02}]_4 \parallel [S_2]_2$ $P'_{3-03} = [S_9.P_{3-02}]_9$	$P_{3-52} = [S_5]_5$ $P_{3-53} = [S_5.P_{3-52}]_5 \parallel [S_6.P_{1-52}]_6$ $P'_{3-53} = [S_6.P_{3-52}]_6$
4	$P_{4-02} = [S_6]_6$ $P'_{4-02} = [S_6.P_{4-02}]_6$	$P_{4-51} = [S_6.P_{4-02}]_6$ $P_{4-52} = [S_4.P_{4-51}]_4$ $P'_{4-52} = [S_5.P_{4-52}]_5$

Observe that in Table 7.6, the departure delays of the return trips do not depend on the departure delays of the forward trips, except in line 4. In our models, we assume that the beginning of the return trips in all lines, except in line 4, are always

on schedule. Indeed, the buffer times in the segments where the return trips begin, as shown in Table 7.4, are set very large precisely in order to avoid delay propagations from late arrivals in the end of the forward trips.

There are two series of experiments in this case study: Weibull and Erlang. In the Weibull experiments, each of the 10 random variables T is distributed according to a Weibull distribution [Wei51]. A random variable X is governed by a Weibull distribution if its distribution function is given by

$$F(t) = \Pr(X \leq t) = \begin{cases} 1 - e^{-\left(\frac{t}{\lambda}\right)^k}, & t \in \mathbb{R}_{\geq 0}, \\ 0, & \text{otherwise,} \end{cases} \quad (7.2)$$

where $k \in \mathbb{R}_+$ and $\lambda \in \mathbb{R}_+$ are called the *shape* and *scale*, respectively, of the distribution. The parameters of the Weibull distributions are determined by the values of r : 1, 2, and 3 correspond to Weibull distributions with shape 2.4; 4, 5, and 6 correspond to Weibull distributions with shape 1.8; and the rest (*i.e.*, 7, 8, 9, and 11) correspond to Weibull distributions with shape 1.4. Furthermore, the scale of the Weibull distribution that corresponds to a particular value of r is chosen such that r roughly corresponds to the 80-th percentile of the Weibull distribution, namely the value of λ when $F(r) = 0.8$. Hence, broadly speaking, 80% of the time, the scheduled running time is met. From Equation (7.2), this is obtained by solving

$$\lambda = \frac{r}{\left(-\log(1 - 0.8)\right)^{\frac{1}{k}}} = \frac{r}{e^{\frac{0.4758849953}{k}}}.$$

Figure 7.10 depicts the distribution and the density functions of the used Weibull distributions. The figures also shows the scales of the Weibull distributions obtained by the previous equation.

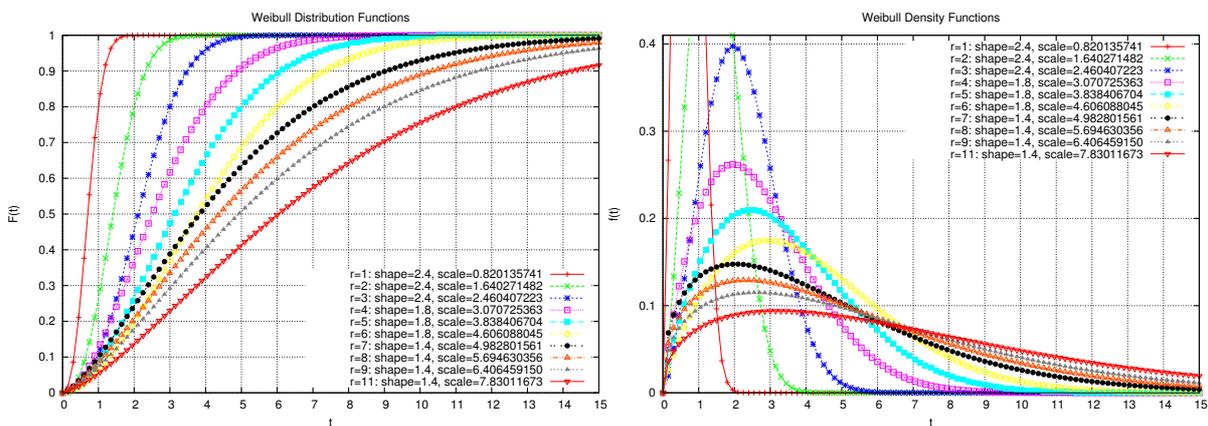


Figure 7.10: The distribution (left) and the density (right) functions of the Weibull distributions used in the Weibull experiments.

Once the Weibull distributions are obtained, they are then approximated by APH representations of size 5, 10, 15, 20, and 30. To carry out these approximations, we use the—currently state-of-the-art—phase-type fitting tool called G-FIT [TBT06]. G-FIT performs approximations by fitting hyper-Erlang distributions to arbitrary traces of probability distributions. Hyper-Erlang distributions are formed by mixtures of several Erlang distributions of different rates and phases.

Here, our model differs from the original model of [MM07]. In our model, we associate a different basic running-time distribution *only* with each distinct value of r . In the original model, on the other hand, a different basic running-time distribution is associated with each value of r . For values of r that occur more than once, Weibull distributions with different shapes are created. Thus there are 24 basic running-time distributions in the original model. We take our approach to save time in generating the basic running-time distributions. We also notice that the same basic running-time distribution is only used in sequences, never in parallels in all processes. Hence, repeatable structures arising from these basic running-time distributions when they are composed will not favorably affect our reduction algorithm.

In the Erlang experiments, we use Erlang distributions instead of Weibull distributions. The phase (size) of the Erlang distributions ranges from 5 to 50. Similar to the Weibull experiment, the rate of an Erlang distribution that corresponds to a particular value of r is chosen such that r roughly corresponds to the 80-th percentile of the Erlang distribution. In Table 7.7, for each phase, we list the rates of the Erlang distributions that correspond to the different values of r .

Table 7.7: Erlang Distributions Used in the Erlang Experiments

Phase	Rates of the Erlang distributions when r is equal to									
	1	2	3	4	5	6	7	8	9	11
5	6.73	3.37	2.25	1.69	1.35	1.13	0.97	0.85	0.75	0.62
10	12.52	6.26	4.18	3.13	2.51	2.09	1.79	1.57	1.40	1.14
15	18.13	9.07	6.05	4.54	3.63	3.03	2.59	2.27	2.02	1.65
20	23.64	11.81	7.88	5.91	4.73	3.94	3.38	2.96	2.63	2.15
30	34.49	17.25	11.50	8.63	6.90	5.75	4.93	4.32	3.84	3.14
50	55.84	27.92	18.62	13.96	11.17	9.31	7.98	6.98	6.21	5.08

Table 7.8 and Table 7.9 summarize the results of the Weibull and Erlang experiments, respectively. In both tables, we present information related to the state spaces of the resulting APH representations associated with the delays of several segments in the railway lines and their distributions at time point 3 (in minutes).

The first column of both tables lists the size of all APH representations used in the experiments for the 10 different basic distributions. In the second column, we list the processes that we are interested in; they are those corresponding to the arrival delays in the end of the last segments in forward and return journeys. We omit those related to line 4, because its segments do not involve in any synchronization. This means that their associated processes consist only of convolution operations, and therefore are not affected by the reduction algorithm (cf. Section 5.3.1). We produce two types of state spaces for each process: without and with using our reduction algorithm (columns 3 and 4, respectively). The computation times (in seconds) of the reduction procedure are listed in columns 5 and 6, separating the transformation (SPA) and the actual reduction parts. The last column of both tables provides the probability that the delay associated with each process is no more than 3 minutes.

We observe in Table 7.8 that the reduction algorithm has a significant impact on the size of the resulting state spaces of the listed processes. In several cases, it achieves

Table 7.8: Summary of the Result of the Weibull Experiments

APH Size	Process	State Spaces		Comp. Time (sec.)		Pr($D \leq 3$)
		No Red.	Red.	SPA	Red.	
5	P'_{1-03}	45	37	< 1	< 1	0.88397496
	P'_{1-53}	125	80	1	< 1	0.85873866
	P'_{2-04}	50	42	< 1	< 1	0.90471570
	P'_{2-54}	100	76	1	< 1	0.90658716
	P'_{3-03}	250	152	15	7	0.89049447
	P'_{3-53}	125	82	1	< 1	0.87504607
10	P'_{1-03}	140	75	1	< 1	0.99209628
	P'_{1-53}	450	140	14	5	0.99276604
	P'_{2-04}	150	84	1	< 1	0.99450540
	P'_{2-54}	350	156	22	7	0.99495819
	P'_{3-03}	1450	408	629	258	0.99341906
	P'_{3-53}	450	145	15	5	0.97368998
15	P'_{1-03}	285	76	1	< 1	0.98547602
	P'_{1-53}	975	305	251	83	0.98331980
	P'_{2-04}	300	69	< 1	< 1	0.98993435
	P'_{2-54}	750	89	4	< 1	0.98991878
	P'_{3-03}	4350	264	89	47	0.98664726
	P'_{3-53}	975	306	251	83	0.95686298
20	P'_{1-03}	480	72	1	< 1	0.98581325
	P'_{1-53}	1700	271	214	52	0.98307950
	P'_{2-04}	500	70	1	< 1	0.99118148
	P'_{2-54}	1300	90	3	< 1	0.99046329
	P'_{3-03}	9700	261	82	45	0.98828893
	P'_{3-53}	1700	271	214	52	0.94808065
30	P'_{1-03}	1020	78	1	< 1	0.84113275
	P'_{1-53}	3750	482	1046	502	0.86971871
	P'_{2-04}	1050	58	< 1	< 1	0.92899385
	P'_{2-54}	2850	70	1	< 1	0.97276211
	P'_{3-03}	30750	241	55	27	0.85327802
	P'_{3-53}	3750	484	1037	502	0.94733634

orders-of-magnitude reduction. Referring back to the process definition in Table 7.6, we see that P'_{3-03} and P'_{1-53} are two of the most complex and involved processes: they both contain two parallel composition (or maximum) operations. Without the use of the reduction algorithm, indeed these processes have the largest state spaces for all different sizes of the basic running-time distributions. The impact of the reduction algorithm on these processes is impressive but somewhat irregular in size. The reason for this irregularity is, in several cases, the hyper-Erlang representations produced by G-FIT are far from minimal, and, therefore, often can be reduced by the reduction algorithm even before they are used in further compositions. Furthermore, the hyper-Erlang representations have a nice structure with many states having similar total outgoing rates. As a result, some processes may have smaller final state spaces when the size of the representations of the basic running-time distributions is increased. Even though the produced hyper-Erlang representations can be reduced, G-FIT guarantees that they are the best in the given number of states, as the tool searches for the best fitting from all possible configurations or hyper-Erlang structures.

The computation times of the reduction procedure are dominated by the transformations (SPA), although the difference between the computation times of the transformation and reduction is not as large as that of the previous two case studies. In several cases, the computation times are shorter when the original state space is larger, for instance process P'_{3-03} in size 30 is obtained quicker than it is in size 10, 15, or 20. This is due to the above-mentioned irregularity of the size of the reduction's results.

The resulting delay distributions of the processes at time point 3 (in minutes) follow no apparent patterns with the varying sizes of the APH representations used to approximate the basic running-time distributions. This means that fitting traces to APH representations of larger size does not necessarily produce better approximations. This is because the stopping criterion for the iterative procedure in almost every fitting tools including G-FIT is based on the difference between the values of some variables (such as the approximated parameters or the likelihood) in subsequent iterations, instead of the absolute value of the optimized measure of closeness. The absence of patterns also forbids us to predict the location of the "precise" probabilities.

In the Erlang experiments (cf. Table 7.9) we also obtain significant reductions in the size of the state spaces of the processes. Furthermore, there are clear patterns in how the reduced state spaces grow with the increasing size of the basic running-time distributions. This can be explained by the regularity of the APH representations of Erlang distributions used as the basic running-time distributions. Similar to the first experiments, P'_{3-03} and P'_{1-53} experience the largest reductions.

In the experiments, the size of the original state spaces and the number of states that can be removed from them by the reduction algorithm mostly determine the computation times needed. Indeed, we see that the reduction of process P'_{3-03} takes the most time. In several cases, however, the reductions take more time than the transformations do. Investigating the cases more carefully, we found that in these cases, the states that can be removed are concentrated in the end of the chains of the ordered bidiagonal representations. Hence, the need to solve the systems of linear equations (when removing the states) arises in the end, and, as a result, the systems are big and take much more time to solve.

Contrary to the Weibull experiments, the resulting delay distributions of the processes at time point 3 (in minutes) follow a particular pattern, namely the larger the size of the used APH representations of the basic running-time distributions, the

Table 7.9: Summary of the Result of the Erlang Experiments

APH Size	Process	State Spaces		Comp. Time (sec.)		Pr($D \leq 3$)
		No Red.	Red.	SPA	Red.	
5	P'_{1-03}	45	29	< 1	< 1	0.91717075
	P'_{1-53}	125	53	< 1	< 1	0.90995698
	P'_{2-04}	50	34	< 1	< 1	0.94416608
	P'_{2-54}	100	48	< 1	< 1	0.94609606
	P'_{3-03}	250	74	< 1	< 1	0.92679645
	P'_{3-53}	125	53	< 1	< 1	0.92064101
10	P'_{1-03}	140	59	< 1	< 1	0.94460584
	P'_{1-53}	450	108	2	< 1	0.94866633
	P'_{2-04}	150	69	< 1	< 1	0.97077302
	P'_{2-54}	350	98	1	< 1	0.97382524
	P'_{3-03}	1450	154	5	< 1	0.95994336
	P'_{3-53}	450	108	2	< 1	0.96052168
15	P'_{1-03}	285	89	< 1	< 1	0.96195133
	P'_{1-53}	975	163	7	1	0.96800000
	P'_{2-04}	300	104	< 1	< 1	0.98338193
	P'_{2-54}	750	148	2	1	0.98621536
	P'_{3-03}	4350	234	21	2	0.97482432
	P'_{3-53}	975	163	7	1	0.97811180
20	P'_{1-03}	480	119	< 1	< 1	0.97260921
	P'_{1-53}	1700	218	21	1	0.97888762
	P'_{2-04}	500	139	2	< 1	0.99010732
	P'_{2-54}	1300	198	15	1	0.99239673
	P'_{3-03}	9700	413	52	3	0.98340104
	P'_{3-53}	1700	218	21	1	0.98628849
30	P'_{1-03}	1020	143	2	< 1	0.98523706
	P'_{1-53}	3750	256	69	63	0.99018536
	P'_{2-04}	1050	173	2	< 1	0.99618049
	P'_{2-54}	2850	253	40	35	0.99738696
	P'_{3-03}	30750	357	115	211	0.99255341
	P'_{3-53}	3750	256	69	63	0.99433131
50	P'_{1-03}	2700	239	30	14	0.99486109
	P'_{1-53}	10250	428	481	492	0.99721737
	P'_{2-04}	2750	289	32	14	0.99926165
	P'_{2-54}	7750	423	558	414	0.99961664
	P'_{3-03}	135250	599	1562	3083	0.99812439
	P'_{3-53}	10250	428	439	492	0.99864256

higher the probabilities. This is because we are using Erlang distributions for the basic running-time distributions: the larger the size, the steeper the distribution functions around the means, and, in turn, the faster the 80-th percentile is reached.

Figure 7.11 depicts the distributions of the arrival delays (in minutes) at the end of several segments produced in the Weibull and Erlang experiments. The size of each APH representation used in the experiments for the 10 basic distributions is 15. Comparing the results of the two experiments, we see that the delay distributions in Weibull experiments are relatively larger than in the Erlang experiments. This means that the arrival delays are probabilistically shorter in the former than in the latter. This can be explained by the fact that the Weibull distributions we used are faster in reaching the 80-th percentile than the corresponding Erlang distributions.

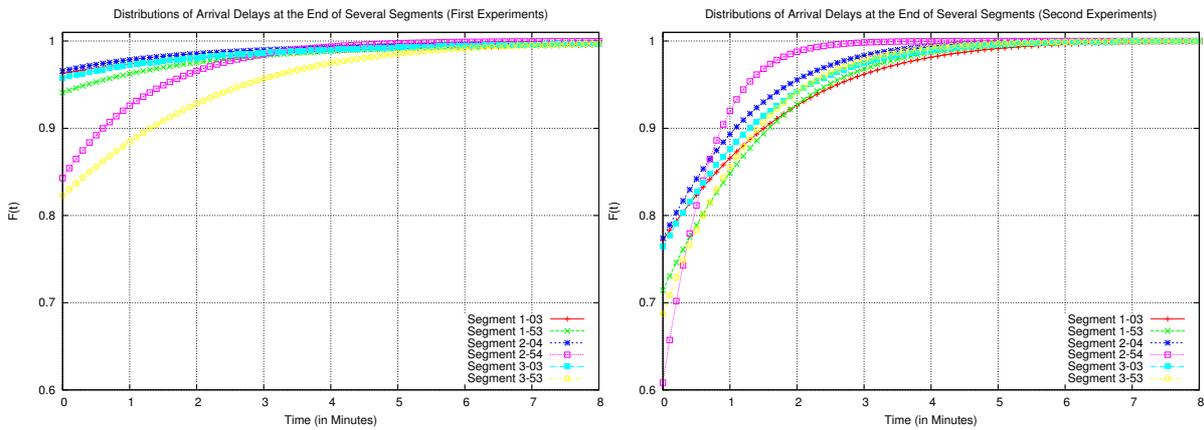


Figure 7.11: The distributions of the arrival delays at the end of several segments of the railway lines produced in the Weibull (left) and the Erlang (right) experiments.

In Figure 7.12, we show the distributions of the arrival delays (in minutes) at the end of segment 53 of line 3 produced in the Weibull and Erlang experiments. The size of APH representations for the 10 basic distributions ranges from 5 to 30 and 50, respectively. As we have observed before, the resulting distributions are more regular in the Erlang than in the Weibull experiments. In the Erlang experiments, larger size results in faster distributions, while in the Weibull experiment, there is no such pattern. However, as in the previous case, the delay distributions in the Weibull experiments are relatively larger than in the Erlang experiments, except for the case of size 5.

To close this section, we would like to offer several remarks. First, in the original case study of the delay propagation in [MM07], APH representations are fitted to the Weibull distributions by using the fitting tool EMPHT [ANO96]. We have also carried out experiments using this tool. Using EMPHT, traces can be fitted by APH representations of triangular form or by Cox representations. In both of these cases, the representations of the fitting results are highly irregular and unstructured. In almost all fitting results, we found that all states have distinct total outgoing rates. As a result, when a maximum operation is applied to the representations, the representation of the maximum is already minimal, and therefore cannot be profitably affected by our reduction algorithm (see Section 5.3.2).

Second, in the first series of experiments we did not provide results for size 50. The reason for this is the time to obtain the hyper-Erlang representations of size 50 from G-FIT is simply immense. This is apparently because to produce the best fittings, the

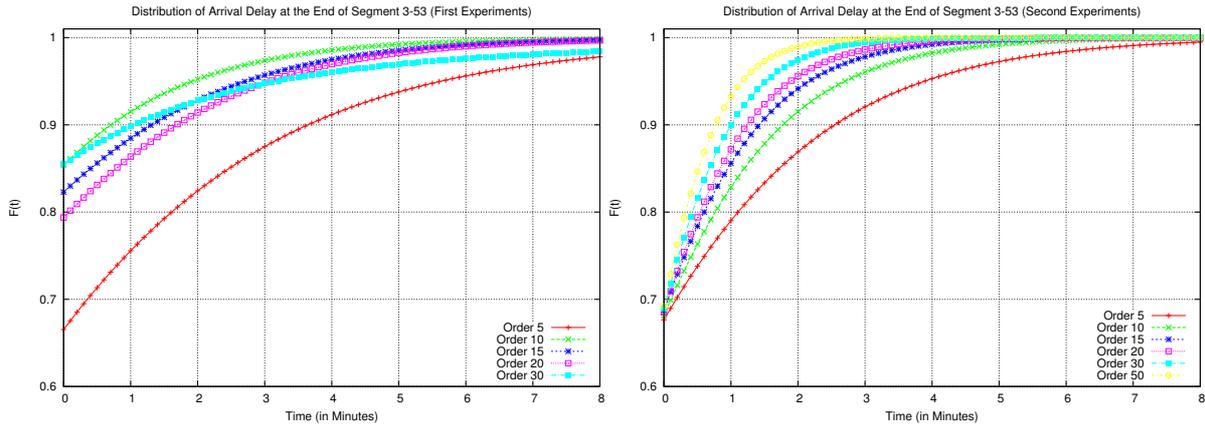


Figure 7.12: The distributions of the arrival delays at the end of segment 53 of line 3 produced in the Weibull (left) and the Erlang (right) experiments.

tool must search all possible configurations, whose number increases drastically with the number of states of the used hyper-Erlang representations.

Third, when we compare the results of our experiments to those of the original experiments [MM07], we find that the probabilities for the delay being at most 3 minutes (*i.e.*, $\Pr(D \leq 3)$) differ for processes P'_{2-54} and P'_{3-53} . For these processes, the original experiments obtained probabilities 0.739 and 0.277, respectively, while in our experiments with size 5, we obtain probabilities 0.90658716 and 0.87504607, respectively. So far, we do not know the reason behind these discrepancies.

7.4 Conclusion

In this chapter, we have demonstrated the practical potential of the reduction algorithm and the three operations by using three case studies.

In the first case study, we demonstrated the possibility of extending the static fault-tree formalism by broadening the domain of the probability distributions of basic events to include APH distributions. The possibility arises because the minimum and maximum operations on APH representations correspond exactly to OR and respectively AND gates in fault trees. Furthermore, in this setting, we showed the role the reduction algorithm plays in reducing the size of the state spaces of several variants of the 3P2M model. We clarified that although minimal representations of the maximum of Erlang distributions grow exponentially with the number of components, the prospect is not completely daunting: for a particular number of components, the minimal representations grow additively in the sizes of the components, and when the number of components are small, the size of the minimal representations is actually manageable.

We repeat that with the current implementation of the reduction algorithm, we have seen the limit of the scalability of the algorithm in dealing with 3P2M models or similar models that involve the minimum and maximum of highly structured APH representations. As a rule of thumb, the algorithm should only be used when the largest intermediate model is no more than around 300000 states.

In the second case study, we turned our attentions to models that are not lim-

ited to the results of the three operations. We demonstrated that the reduction algorithm is also useful in this case. The CTMCs of the FTTP models are obtained by applying a compositional semantics of dynamic fault trees [BCS07a], in which in every step of the composition, state-space aggregation algorithm (based on Markovian weak-bisimulation equivalence) is applied. The original models in Table 7.3 are the smallest models possible under weak-bisimulation equivalence. That our reduction algorithm manages to reduce these models even further is an encouraging prospect for models obtained in similar fashions.

In the third case study, we studied a stochastic model of delay propagations in a part of the Dutch railway network. The stochastic model is built through a method that is based on the Max-Plus algebra [Gov07]. Since the operations on the algebra can be precisely captured by the maximum and convolution operations on PH distributions [MM07], PH distributions are a perfect ingredient for such models. We show that the reduction algorithm is useful in keeping the size of the intermediate and final models small when the basic delays are distributed according to the nicely structured Erlang distributions. Even more encouraging, however, is the fact that the reduction algorithm is also usable when the basic delays are obtained through phase-type fittings.

As final thoughts, we point out the key strength and weakness of the algorithm based on what we have learned from the case studies. The major strength of the algorithm is the possibly enormous reduction it can achieve. When the reduction algorithm is coupled with the three operations we previously defined, we can even be certain that the results are almost surely minimal. We also observe that in this scenario, the effect of the algorithm is even more considerable when APH representations involved in the operations have more states with similar total outgoing rates. This can stand as a rough guideline on when to use or not to use the algorithm.

On the other hand, the major weakness of the reduction algorithm is its demanding computation times. This arises from the need to obtain exact representations, and then in our resorting to exact arithmetic in the prototypical implementation. In the future, we intend to investigate the effects of floating-point roundings on the transformations to ordered bidiagonal representations, on the reduction algorithm, and on the resulting distribution functions. We will also study the possibility of estimating and bounding the deviation errors between the original and the resulting APH distributions in such a circumstance.

Remarks All experiments in the case studies were run on PCs with Pentium 4, 2.66 GHz processor with 1 GB RAM running Linux 2.6.27-9.

Chapter 8

Conclusion

In this thesis, we have proposed a new algorithm to reduce the state space of acyclic phase-type representations. In each iteration, the algorithm requires quadratic time (in the current number of states), to reduce the representation by at least one state, as long as no further reduction is possible. This means that the algorithm is at worst of cubic time complexity in the size of the state space. Although the algorithm does not always return the minimal representation of an input acyclic phase-type distribution, we showed that when the input distribution is triangular ideal (*i.e.*, the triangular order of the distribution is equal to its algebraic degree), the algorithm can always reduce it to its minimal representation.

We have also investigated the effect of the convolution, minimum, and maximum operations on ordered bidiagonal representations that represent triangular-ideal acyclic phase-type distributions. We proved that given two such arbitrary ordered bidiagonal representations, the three operations almost always produce a triangular-ideal acyclic phase-type distribution. In other words, if we are to restrict ourselves to use only triangular-ideal acyclic phase-type distributions in our stochastic modelling formalism that is equipped with the three operations, we will only have to deal with models that are almost surely minimal.

We have provided a modelling framework that captures acyclic phase-type representations and their manipulation by the three defined operations in the form of a simple stochastic process calculus, CCC. We have applied the framework in analyzing two of the case studies. Since the three operations arise often in many stochastic models—for instance maximum and convolution operations in the Max-Plus algebra—the framework provides a convenient environment for stochastic modelling and analysis. In fact the model in the third case study (cf. Section 7.3) is built by a method that is based on the Max-Plus algebra.

These are the contributions of this thesis. And through these, we hope that we have made a contribution to the general effort to overcome the state space problem the field of stochastic state-based analysis is facing. That the method is useful in practice has been demonstrated by its use in several—realistic, albeit small—case studies we provided. Further improvements on the implementation will certainly enable us to make use of the method in even larger and more complex models or case studies. In this case, we see potential benefits in parallelizing the method to address its demanding computation time.

Future Work Looking forward to the challenges ahead, we consider addressing the main practical weakness of the algorithm we discussed in the end of Chapter 7 important. Removing the burden of exact arithmetic in the prototypical implementation will certainly relieve the demanding computation-time requirement of the algorithm. As we have mentioned before, this requires further investigations on the effects of floating-point roundings on the transformations to ordered bidiagonal representations, on the reduction algorithm itself, and on the resulting distribution functions. A successful endeavor in this direction will open the possibility of an approximation method to safely bound a phase-type distribution of high order by a phase-type distribution of smaller order.

In the end of Section 7.1, we commented on the fact that in CCC, a process always refers to a structure or a representation, never to an identity. A process may contain many sub-processes with the same name, but these sub-processes must be independent of each other, and there is no way to identify them as a single sub-process. We have shown that this inability to impose a dependency between similarly named processes restricts the use of the calculus, for instance in representing the whole static fault trees. In the future, we would like to find ways to address this problem.

Another direction worth pursuing is that of incorporating the reduction method in the analysis and manipulation of the general (cyclic) phase-type representations by devising an effective and automated method for identifying reducible acyclic sub-chains and then applying the reduction method on them.

On a broader and long-term perspective, we would also like to explore the possibility of applying a similar reduction technique that is based on the Laplace-Stieltjes transform's L -terms to the general (cyclic) phase-type representations, possibly by combining it with cycle unfolding.

Appendix A

Basic Concepts

A.1 Poisson Processes

Most of the material in this section is taken from [Ros07].

A stochastic process $\{N_t \mid t \in \mathbb{R}_{\geq 0}\}$ is called a counting process if the random variable N_t represents the number of occurrences of certain events within t time units. The set of Poisson processes is a subset of counting processes.

Definition A.1. A counting process $\{N_t \mid t \in \mathbb{R}_{\geq 0}\}$ is called a Poisson process with rate $\lambda \in \mathbb{R}_+$ if

1. $N_0 = 0$,
2. the process has independent increments, i.e., the number of events that occur in disjoint intervals are independent,
3. the number of events in any interval of length $t \in \mathbb{R}_{\geq 0}$ has distribution function

$$\Pr(N_{s+t} - N_s = n) = e^{-\lambda t} \frac{(\lambda t)^n}{n!}, \quad n \in \mathbb{Z}_{\geq 0}.$$

In practice, it is often difficult to determine whether a real-life process fits to some stochastic process model. For Poisson processes, however, we have two guidelines provided by the following properties:

1. $\Pr(N_{t+h} - N_t = 1) = \lambda h + o(h)$, which is to say that at any time point, the probability that one event occurs during a small duration h of time is approximately λh , and
2. $\Pr(N_{t+h} - N_t \geq 2) = o(h)$, namely the probability that more than one events occur during the same small duration h is negligible.

Note that $o(h)$ is an o -function. A function $f(x)$ is an o -function if $\lim_{h \rightarrow 0} \frac{f(h)}{h} = 0$.

Let $t_1 < t_2 < \dots < t_n$ be the time points of the occurrences of some events governed by a Poisson process with rate λ . For a sequence $\{T_i \mid i \in \mathbb{Z}_+\}$, let $T_i = t_i - t_{i-1}$, where we define $t_0 = 0$. The sequence describes *interarrival times* of the events. Each of T_i is a random variable, and we are interested in its distribution. It can be shown (for instance in [Ros07]) that T_i , for $i \in \mathbb{Z}_+$, are independently identically distributed by

an exponential distribution with rate λ . Therefore, the interarrival times of events in a Poisson process are exponentially distributed.

Furthermore, let $S_n = \sum_{i=1}^n T_i$, for $n \in \mathbb{Z}_+$. S_n describes the arrival time of the n -th event, or the *waiting time* until the n -th event. Each of S_n is a random variable, and it is distributed by an Erlang distribution with rate λ and phase n .

A.2 Kronecker Product and Sum

Let \mathbf{A} be a matrix of dimension kl , and \mathbf{B} be a matrix of dimension mn . The *Kronecker product* of the matrices is an $(km \times nl)$ -dimension matrix

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} \mathbf{A}(1,1)\mathbf{B} & \mathbf{A}(1,2)\mathbf{B} & \cdots & \mathbf{A}(1,l)\mathbf{B} \\ \mathbf{A}(2,1)\mathbf{B} & \mathbf{A}(2,2)\mathbf{B} & \cdots & \mathbf{A}(2,l)\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}(k,1)\mathbf{B} & \mathbf{A}(k,2)\mathbf{B} & \cdots & \mathbf{A}(k,l)\mathbf{B} \end{bmatrix}.$$

For square matrices \mathbf{A} and \mathbf{B} of dimensions m and n , respectively, the *Kronecker sum* is a square matrix of dimension mn

$$\mathbf{A} \oplus \mathbf{B} = (\mathbf{A} \otimes \mathbf{I}_n) + (\mathbf{I}_m \otimes \mathbf{B}),$$

where \mathbf{I}_x is the identity matrix of dimension x .

A.3 Some Concepts from Convex Analysis

The material in this section is taken from Sections 1 and 2 of [Roc70].

A.3.1 Affine Sets

Let \vec{x} and \vec{y} be two different points in the vector space \mathbb{R}^n , the set of points forming

$$(1 - \lambda)\vec{x} + \lambda\vec{y} = \vec{x} + \lambda(\vec{y} - \vec{x}), \quad \lambda \in \mathbb{R}$$

is called *the line through \vec{x} and \vec{y}* .

A subset $M \subseteq \mathbb{R}^n$ is called an *affine set* if $(1 - \lambda)\vec{x} + \lambda\vec{y} \in M$, for every $\vec{x} \in M$, $\vec{y} \in M$, and $\lambda \in \mathbb{R}$. In general, an affine set has to contain, along with any two different points, the entire line through those points. Thus, they form an endless uncurved structure or flat.

Theorem A.2. *The subspaces of \mathbb{R}^n are the affine sets that contain the origin.*

For a set $M \subseteq \mathbb{R}^n$ and $\vec{y} \in \mathbb{R}^n$, the *translate* of M by \vec{y} is defined by

$$M + \vec{y} = \{\vec{x} + \vec{y} \mid \vec{x} \in M\}.$$

A translate of an affine set is another affine set. An affine set M is said to be *parallel* to an affine set L if $M = L + \vec{y}$, for some $\vec{y} \in \mathbb{R}^n$.

Theorem A.3. Each non-empty affine set M is parallel to a unique subspace L , which is given by

$$L = M - M = \{\vec{x} - \vec{y} \mid \vec{x} \in M, \vec{y} \in M\}.$$

The *dimension* of a non-empty affine set is defined as the dimension of the subspace parallel to it. Affine sets of dimension 0, 1, and 2 are called *points*, *lines*, and *planes*, respectively. An $(n - 1)$ -dimensional affine set in \mathbb{R}^n is called a *hyperplane*.

Theorem A.4. Let $\vec{y} \in \mathbb{R}^n$, and let \mathbf{B} be an (mn) -dimension real matrix. The set defined by

$$M = \{\vec{x} \in \mathbb{R}^n \mid \mathbf{B}\vec{x} = \vec{y}\}$$

is an affine set in \mathbb{R}^n . Every affine set can be represented in this way.

The intersection of an arbitrary collection of affine sets is also an affine set. Therefore, given $S \subseteq \mathbb{R}^n$, there exists a unique smallest affine set containing S (namely, the intersection of the collection of affine sets M such that $S \subseteq M$). This set is called the *affine hull* of S , denoted by $\text{aff}(S)$.

A set of $m + 1$ points $\{\vec{x}_0, \vec{x}_1, \dots, \vec{x}_m\}$ is *affinely independent* if $\text{aff}(\{\vec{x}_0, \vec{x}_1, \dots, \vec{x}_m\})$ is m -dimensional. Furthermore, $\vec{x}_0, \vec{x}_1, \dots, \vec{x}_m$ are affinely independent if and only if $\vec{x}_1 - \vec{x}_0, \vec{x}_2 - \vec{x}_0, \dots, \vec{x}_m - \vec{x}_0$ are linearly independent.

A.3.2 Convex Sets

A subset $C \subseteq \mathbb{R}^n$ is *convex* if $(1 - \lambda)\vec{x} + \lambda\vec{y} \in C$, for all $\vec{x} \in C$, $\vec{y} \in C$, and $0 < \lambda < 1$. Note that all affine sets are convex. The set defined by

$$\{(1 - \lambda)\vec{x} + \lambda\vec{y} \mid 0 \leq \lambda \leq 1\}$$

is called the (*closed*) *line segment* between \vec{x} and \vec{y} .

Theorem A.5. The intersection of an arbitrary collection of convex sets is convex.

A vector sum

$$\lambda_1\vec{x}_1 + \lambda_2\vec{x}_2 + \dots + \lambda_n\vec{x}_m$$

is called a *convex combination* of $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_m$ if, for $1 \leq i \leq m$, $\lambda_i \in \mathbb{R}_{\geq 0}$ and $\lambda_1 + \lambda_2 + \dots + \lambda_m = 1$.

Theorem A.6. A subset of \mathbb{R}^n is convex if and only if it contains all the convex combinations of its elements.

The intersection of all the convex sets containing a given subset $S \subseteq \mathbb{R}^n$ is called a *convex hull* of S , denoted by $\text{conv}(S)$.

Theorem A.7. For all $S \subseteq \mathbb{R}^n$, $\text{conv}(S)$ consists of all the convex combinations of the elements of S .

A set which is the convex hull of a finitely many points is called a *polytope*. If a set of $m + 1$ points $\{\vec{x}_0, \vec{x}_1, \dots, \vec{x}_n\}$ is affinely independent, its convex hull is called an *n -dimensional simplex*, and each of $\vec{x}_0, \vec{x}_1, \dots, \vec{x}_n$ is called a *vertex* of the simplex. In general, by the *dimension* of a convex set C , we mean the dimension of the affine hull of C .

Appendix B

Proofs

B.1 Lemma 3.14

To prove Lemma 3.14, we first show how to transform Equation (3.14) to Equation (3.15).

Let $\vec{\alpha}$ be a vector of dimension m and $\mathbf{Bi}(\mu_m, \mu_{m-1}, \dots, \mu_1)$ be a PH-generator of size m . Furthermore, let $\vec{\alpha}^{(j)} = \vec{\alpha} \mathbf{Bi}(\mu_m, \mu_{m-1}, \dots, \mu_1)^j$, for $j \in \mathbb{Z}_{\geq 0}$. Then

$$\begin{aligned} \vec{\alpha}^{(1)} &= [\vec{\alpha}_m, \vec{\alpha}_{m-1}, \dots, \vec{\alpha}_1] \begin{bmatrix} -\mu_m & \mu_m & \cdots & 0 & 0 \\ 0 & -\mu_{m-1} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -\mu_2 & \mu_2 \\ 0 & 0 & \cdots & 0 & -\mu_1 \end{bmatrix}, \\ &= [-\vec{\alpha}_m \mu_m, \vec{\alpha}_m \mu_m - \vec{\alpha}_{m-1} \mu_{m-1}, \dots, \vec{\alpha}_2 \mu_2 - \vec{\alpha}_1 \mu_1]. \end{aligned} \quad (\text{B.1})$$

By the associativity of matrix multiplications, we have the identities

$$\vec{\alpha}^{(j)} = \vec{\alpha}^{(j-1)} \mathbf{Bi}(\mu_m, \mu_{m-1}, \dots, \mu_1), \quad j \in \mathbb{Z}_+,$$

with $\vec{\alpha}^{(0)} = \vec{\alpha}$. From Equation (B.1), we obtain

$$\begin{aligned} \vec{\alpha}^{(1)} \vec{e}|_m &= -\vec{\alpha}_1 \mu_1, \quad m \in \mathbb{Z}_+, \text{ and} \\ \vec{\alpha}_k^{(1)} &= \vec{\alpha}_{k+1} \mu_{k+1} - \vec{\alpha}_k \mu_k, \quad 1 \leq k \leq m, \end{aligned}$$

where we define $\mu_{m+1} = 0$ and $\vec{\alpha}_{m+1} = 0$. These equations can be generalized to the following recursive equations

$$\vec{\alpha}^{(j)} \vec{e}|_m = -\vec{\alpha}_1^{(j-1)} \mu_1, \quad j, m \in \mathbb{Z}_+, \quad (\text{B.2})$$

$$\vec{\alpha}_k^{(j)} = \vec{\alpha}_{k+1}^{(j-1)} \mu_{k+1} - \vec{\alpha}_k^{(j-1)} \mu_k, \quad 1 \leq k \leq m \text{ and } j \in \mathbb{Z}_+, \quad (\text{B.3})$$

and we define $\mu_{m+1} = 0$ and $\vec{\alpha}_{m+1}^{(j)} = 0$, for $j \in \mathbb{Z}_{\geq 0}$.

Lemma B.1. For any $m \geq 2$

$$\vec{\alpha}^{(k)} \vec{e}|_m = \sum_{j=1}^k \vec{\alpha}_j c(k, j), \quad 1 \leq k < m, \quad (\text{B.4})$$

where

$$c(k, j) = \begin{cases} 0, & \text{for } j \leq 0, \\ 0, & \text{for } j > k \text{ and } k > 0, \\ 1, & \text{for } j > k \text{ and } k = 0, \\ \mu_j (c(k-1, j-1) - c(k-1, j)), & \text{otherwise.} \end{cases} \quad (\text{B.5})$$

Lemma B.2. For each $1 \leq k < m$

- (1) computing all $c(k, j)$, for $1 \leq j \leq k$, requires k multiplications and k additions, and
- (2) computing all $\vec{\alpha}^{(k)} \vec{e}|_m$, for $1 \leq j \leq k$, requires $2k$ multiplications and k additions,

given that $c(k-1, j)$, for $1 \leq j \leq k-1$, are known.

Proof. For assertion (1): From Equation (B.5), to compute all $c(k, j)$, for $1 \leq j \leq k$, we need $c(k-1, j)$, for $0 \leq j \leq k$. However, by definition $c(k-1, 0) = 0$ and $c(k-1, k) = 0$. Since $c(k, j) = \mu_j (c(k-1, j-1) - c(k-1, j))$ for the non-trivial condition, computing a single $c(k, j)$ requires one multiplications and one additions. Therefore, we need k multiplications and k additions for all k of them.

For assertion (2): Since we need to multiply $\vec{\alpha}_j$ for each obtained $c(k, j)$, we need k multiplications more. \square

Lemma B.3. Obtaining vector \vec{b} in Equation (3.15) requires $i^2 - 3i + 2$ multiplications and $\frac{i^2 - i + 2}{2}$ additions.

Proof. Vector \vec{b} is a column vector obtained by evaluating the right-hand side of Equation (3.14) $i - 1$ times, i.e.,

$$\vec{b}_{i-k} = [\vec{\beta}_1, \vec{\beta}_2, \dots, \vec{\beta}_i] \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_i)^k \vec{e}|_i, \quad 1 \leq k < i - 1,$$

$$\text{and } \vec{b}_1 = [\vec{\beta}_1, \vec{\beta}_2, \dots, \vec{\beta}_i] \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_i)^0 \vec{e}|_i = \vec{\beta}_1 + \vec{\beta}_2 + \dots + \vec{\beta}_i.$$

Since

$$[\vec{\beta}_1, \vec{\beta}_2, \dots, \vec{\beta}_i] \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_i)^k \vec{e}|_i = [\vec{\beta}_1, \vec{\beta}_2, \dots, \vec{\beta}_i]^{(k)} \vec{e}|_i, \quad 1 \leq k < i - 1,$$

from Lemma B.2(2), computing each one of them requires $2k$ multiplications and k additions. Therefore, to obtain vector \vec{b} , we need $\sum_{k=1}^{i-2} 2k = i^2 - 3i + 2$ multiplications and $\sum_{k=1}^{i-2} k + i = \frac{i^2 - i + 2}{2}$ additions. The extra i additions are from calculating \vec{b}_1 . \square

Lemma B.4. Obtaining matrix \mathbf{A} in Equation (3.15) requires $\frac{i^2 - 3i + 2}{2}$ multiplications and $\frac{i^2 - 3i + 2}{2}$ additions.

Proof. We obtain \mathbf{A} by evaluating the left-hand side of Equation (3.14) $i - 1$ times. Let $\vec{a} = \mathbf{A} \vec{\gamma}^\top$; then

$$\vec{a}_{i-k} = [\vec{\gamma}_1, \vec{\gamma}_2, \dots, \vec{\gamma}_{i-1}] \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_{i-1})^k \vec{e}|_{i-1}, \quad 1 \leq k < i - 1,$$

$$\text{and } \vec{a}_1 = [\vec{\gamma}_1, \vec{\gamma}_2, \dots, \vec{\gamma}_{i-1}] \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_{i-1})^0 \vec{e}|_{i-1} = \vec{\gamma}_1 + \vec{\gamma}_2 + \dots + \vec{\gamma}_{i-1}.$$

Since

$$[\vec{\gamma}_1, \vec{\gamma}_2, \dots, \vec{\gamma}_{i-1}] \mathbf{Bi}(\lambda_1, \lambda_2, \dots, \lambda_i)^k \vec{e}|_{i-1} = [\vec{\gamma}_1, \vec{\gamma}_2, \dots, \vec{\gamma}_i]^{(k)} \vec{e}|_{i-1}, \quad 1 \leq k < i - 1,$$

and observing the form of Equation (B.4), matrix \mathbf{A} is an upper-triangular matrix. The components of the matrix are

$$\mathbf{A}(i - k, i - j) = c(k, j), \quad 1 \leq j \leq k, 1 \leq k < i - 1,$$

and $\mathbf{A}(1, j) = 1$, for $1 \leq j \leq i - 1$.

From Lemma B.2(1), computing each one of $c(k, j)$ requires k multiplications and k additions. Therefore, to obtain matrix \mathbf{A} , we need $\sum_{k=1}^{i-2} k = \frac{i^2-3i+2}{2}$ multiplications and $\sum_{k=1}^{i-2} k = \frac{i^2-3i+2}{2}$ additions. \square

In conclusion, to transform Equation (3.14) to Equation (3.15), *i.e.*, computing matrix \mathbf{A} and vector \vec{b} , requires $\frac{3i^2-9i+6}{2}$ multiplications and $i^2 - 2i + 2$ additions. Hence, the transformation requires $\mathcal{O}(i^2)$ multiplications and $\mathcal{O}(i^2)$ additions.

B.2 Lemma 5.14

Convolution The convolution of two hypoexponential representations is a hypoexponential representation formed by concatenating the transient states of the two representations. Since a hypoexponential representation has only a single initial state, there is only one path in the representation. This single path forms the core series. But a hypoexponential representation is always minimal (cf. Lemma 2.32) and is of size that is equal to its algebraic degree. Since the size of the hypoexponential representation is equal to the size of its longest core series, the number of states in the longest core series is equal to its algebraic degree.

Minimum and Maximum For minimum and maximum, the proof is similar to that of Lemma 4.20.

Let l be the size (or the length) of the longest core series. To prove the lemma for the case of minimum and maximum, we just have to show that the algebraic degree of the distribution associated with the resulting representation of each operation is equal to the size of the longest core series l .

The algebraic degree cannot be more than l , since from the core series we can build an ordered bidiagonal representation of size l . In the rest of the proof we show that the algebraic degree is not less than l either. Let $(\vec{e}_1, \mathbf{Bi}(\nu_1, \nu_2, \dots, \nu_l))$ be the longest core series, and let $\mathbf{Bi} := \mathbf{Bi}(\nu_1, \nu_2, \dots, \nu_l)$.

Consider the polytope of the PH-generator \mathbf{Bi} , $\text{PH}(\mathbf{Bi})$. Since \mathbf{Bi} is an ordered bidiagonal PH-generator, it is PH-simple. Therefore the polytope is l -dimensional, *i.e.*, it resides in an l -dimensional affine subspace [Roc70]. Let $\psi \subseteq \{1, 2, \dots, l\}$ and $\psi \neq \emptyset$. With each ψ we associate a bidiagonal representation $q_\psi := (\vec{e}_1, \mathbf{Bi}_\psi)$ where the PH-generator \mathbf{Bi}_ψ is built by all ν_j 's such that $j \in \psi$. Let Ψ denote the collection of all such ψ . By Lemma 2.45, the associated PH distribution of each $\psi \in \Psi$ is on the boundary of the polytope.

We have shown in the proof of Lemma 4.20 that the l polytopes

$$\begin{aligned} & \text{conv}(\{\delta, q_{\{l\}}, q_{\{l-1,l\}}, \dots, q_{\{3,\dots,l-1,l\}}, q_{\{2,3,\dots,l-1,l\}}\}), \\ & \text{conv}(\{\delta, q_{\{l\}}, q_{\{l-1,l\}}, \dots, q_{\{3,\dots,l-1,l\}}, q_{\{1,3,\dots,l-1,l\}}\}), \\ & \quad \vdots \\ & \text{conv}(\{\delta, q_{\{l\}}, q_{\{l-2,l\}}, \dots, q_{\{2,\dots,l-2,l\}}, q_{\{1,2,\dots,l-2,l\}}\}), \\ & \text{conv}(\{\delta, q_{\{l-1\}}, q_{\{l-2,l-1\}}, \dots, q_{\{2,\dots,l-2,l-1\}}, q_{\{1,2,\dots,l-2,l-1\}}\}). \end{aligned}$$

is an $(l - 1)$ -dimensional polytope, *i.e.*, it resides in an $(l - 1)$ -dimensional affine subspace.

The intersection of all of the $(l - 1)$ -dimensional affine subspaces in which the smaller polytopes resides and the polytope $\text{PH}(\mathbf{Bi})$ is exactly the region containing APH distributions of algebraic degree $l - 1$ or less with poles taken from $\{-\nu_1, -\nu_2, \dots, -\nu_l\}$.

Let the underlying CTMC of the standard representation of

$$\text{the minimum } \min((\vec{e}_1, \mathbf{Bi}(\lambda_1, \dots, \lambda_m)), (\vec{e}_1, \mathbf{Bi}(\mu_1, \dots, \mu_n)))$$

$$(\text{or the maximum } \max((\vec{e}_1, \mathbf{Bi}(\lambda_1, \dots, \lambda_m)), (\vec{e}_1, \mathbf{Bi}(\mu_1, \dots, \mu_n))))$$

be $\mathcal{M} = (\mathcal{S}, \mathbf{R}, \vec{\pi})$. From the structure of the representation, we know that each path $\sigma \in \text{Paths}(\mathcal{M})$ is of length $m + n - 1$ (or $m + n$, respectively), which is less than $l - 1$. For each of the $(l - 1)$ -dimensional affine subspaces, we show that there is a path $\sigma_{out} \in \text{Paths}(\mathcal{M})$ whose distribution resides outside it.

Take an arbitrary $(l - 1)$ -dimensional affine subspace. There must be a total outgoing rate E_h —and the corresponding distribution $q_{\{h\}}$ —whose multiplicity in the set generating the $(l - 1)$ -dimensional affine subspace is $c(\mathcal{M}, E_h) - 1$. Take a path $\sigma_{out} \in \text{Paths}(\mathcal{M})$ such that there are $c(\mathcal{M}, E_h)$ states with total outgoing rate E_h in the path. The existence of this path can be ascertained by observing the form of the standard representation in Figure 5.11 (or Figure 5.12, respectively) and ignoring the initial probabilities of all states but state $(1, 1)$. The distribution associated with this path resides outside the $(l - 1)$ -dimensional affine subspace.

B.3 Lemma 6.12

We need the following definition in the proof sketch.

Definition B.5. An ordered sub-process list of $P \in \mathcal{L}$ is $\llbracket P \rrbracket = \langle P_1, P_2, \dots, P_n \rangle$, where $\text{Reach}(P) = \{P_1, P_2, \dots, P_n\}$ and

$$\forall 1 < i \leq n : \forall 1 \leq j < i : P_j \notin \text{Reach}(P_i).$$

If $\llbracket P \rrbracket = \langle P_1, P_2, \dots, P_n \rangle$ is an ordered sub-process list of $P \in \mathcal{L}$ then $P_1 = P$ and $P_n = \text{stop}$.

Example B.6. Take process $Q = (2\lambda_1).((\lambda_2) \triangleleft (\lambda_1)(\lambda_2)).(\lambda_1)$ depicted in Figure 6.6. Then $\langle Q, ((\lambda_2) \triangleleft (\lambda_1)(\lambda_2)).(\lambda_1), (\lambda_1), (\lambda_2).(\lambda_1), \text{stop}) \rangle$ is not ordered sub-process list of Q , because $(\lambda_1) \in \text{Reach}((\lambda_2).(\lambda_1))$. $\llbracket Q \rrbracket = \langle Q, ((\lambda_2) \triangleleft (\lambda_1)(\lambda_2)).(\lambda_1), (\lambda_2).(\lambda_1), (\lambda_1), \text{stop}) \rangle$, on the other hand, is an ordered sub-processes list of Q . In fact this is the only ordered sub-process list of Q .

Let \mathcal{M}_P and \mathcal{M}_Q be the absorbing CTMC semantics of processes P and Q , respectively, where $|Reach(P)| = m + 1$ and $|Reach(Q)| = n + 1$. The state spaces of \mathcal{M}_P and \mathcal{M}_Q can be reordered in the same way as the ordered sub-process lists $\llbracket P \rrbracket$ and $\llbracket Q \rrbracket$. Now, the associated APH representations of \mathcal{M}_P and \mathcal{M}_Q can be expressed as $(\vec{e}_1|_m, \mathbf{A})$ and $(\vec{e}_1|_n, \mathbf{B})$, respectively, where each of the PH-generators \mathbf{A} and \mathbf{B} is an upper-triangular matrix. PH-generator \mathbf{A} encodes function $\gamma(P_i, P_j)$, for $1 \leq i, j \leq m$, and similarly with PH-generator \mathbf{B} .

Convolution To show that $\text{con}(\text{PH}(P), \text{PH}(Q)) = \text{PH}(P.Q)$, we just have to show that the associated APH representation of $\mathcal{M}_{P.Q}$ is (from Equation (2.18)) $(\vec{e}_1|_{m+n}, \mathbf{G})$, where

$$\mathbf{G} = \begin{bmatrix} \mathbf{A} & \vec{A}\vec{e}_1|_n \\ \vec{0} & \mathbf{B} \end{bmatrix}.$$

Intuitively, rule (2.a) maintains that process $P.Q$ proceeds as process P , which is described by sub-matrix \mathbf{A} in the PH-generator \mathbf{G} . Rule (2.b), on the other hand, maintains that if any $P_i \in \llbracket P \rrbracket$, for $1 \leq i < m$, has a transition to $P_m \in \llbracket P \rrbracket$ —described by the sub-matrix $\vec{A}\vec{e}_1|_m$ —then process $P.Q$ proceeds as process Q , which is described by sub-matrix \mathbf{B} .

Minimum To show that $\text{min}(\text{PH}(P), \text{PH}(Q)) = \text{PH}(P + Q)$, we just have to show that the associated APH representation of \mathcal{M}_{P+Q} is (from Equation (2.19)) $(\vec{e}_1|_{mn}, \mathbf{G})$, where

$$\mathbf{G} = [\mathbf{A} \oplus \mathbf{B}].$$

Let $|Reach(P)| = m + 1$ and $|Reach(Q)| = n + 1$. We fix the value of n , and we perform induction on m .

Induction basis $m = 1$: Process P must be of the form (λ) . Intuitively, repeated applications of SOS rule (4.b) in Table 6.1 result in processes $(\lambda) + Q$, $(\lambda) + Q_1$, \dots , $(\lambda) + Q_n$. However, for each process $(\lambda) + Q_i \in \langle (\lambda) + Q_1, (\lambda) + Q_2, \dots, (\lambda) + Q_n \rangle - \{(\lambda) + Q_n\}$, we can apply SOS rule (4.c), which results in a transition from $(\lambda) + Q_i$ to stop with rate λ . This is reflected in the following PH-generator of the associated APH representation of $\mathcal{M}_{(\lambda)+Q}$

$$\begin{aligned} [[-\lambda] \oplus \mathbf{B}] &= [-\lambda \otimes \mathbf{I}_n + [1] \otimes \mathbf{B}], \\ &= [\mathbf{B} - \lambda \mathbf{I}_n]. \end{aligned}$$

The original transitions of the state space of \mathbf{B} is preserved, and additional transitions with rate λ from each state to the absorbing state—represented by $-\lambda \mathbf{I}_n$ —are added.

Now, we assume that for $m = k$, the associated APH representation of \mathcal{M}_{P+Q} is $(\vec{e}_1|_{kn}, \mathbf{G})$, where

$$\mathbf{G} = [\mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_k \otimes \mathbf{B}].$$

Induction step $m = k+1$: Take an arbitrary process P' , such that $|Reach(P')| = k+1$, and let the associated PH-generator of the APH representation of $\mathcal{M}_{P'}$ be \mathbf{A} . Then process P must be a process that can reach some processes in $Reach(P')$. Let the sum of the rates of all transition from P to the processes in $Reach(P')$ be λ .

Intuitively, repeated applications of SOS rule (4.b) in Table 6.1 result in processes $P + Q$, $P + Q_1$, \dots , $P + Q_n$. However, for each process $P + Q_i \in \langle P + Q_1, P +$

$Q_2, \dots, P + Q_n) - \{P + Q_n\}$, we can apply SOS rule (4.c), which results in a transition from $P + Q_i$ to $P' + Q$. This is reflected in the following PH-generator of the associated APH representation of \mathcal{M}_{P+Q}

$$\begin{aligned} \left[\begin{array}{cc} -\lambda & \vec{x} \\ \vec{0} & \mathbf{A} \end{array} \right] \oplus \mathbf{B} &= \left[\begin{array}{cc} -\lambda & \vec{x} \\ \vec{0} & \mathbf{A} \end{array} \right] \otimes \mathbf{I}_n + \left[\begin{array}{cc} 1 & \vec{0} \\ \vec{0} & \mathbf{I}_k \end{array} \right] \otimes \mathbf{B}, \\ &= \left[\begin{array}{cc} -\lambda \otimes \mathbf{I}_n & \vec{x} \otimes \mathbf{I}_n \\ \mathbf{0} & \mathbf{A} \otimes \mathbf{I}_n \end{array} \right] + \left[\begin{array}{cc} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_k \otimes \mathbf{A} \end{array} \right], \\ &= \left[\begin{array}{cc} \mathbf{B} - \lambda \mathbf{I}_n & \vec{x} \otimes \mathbf{I}_n \\ \mathbf{0} & \mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_k \otimes \mathbf{B} \end{array} \right]. \end{aligned}$$

Process $P' + Q$ is represented by the sub-matrix $\mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_k \otimes \mathbf{B}$. The original transitions of the state space of \mathbf{B} is preserved, and additional transitions with accumulated rate λ from each state in \mathbf{B} to $\mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_k \otimes \mathbf{B}$ —represented by $-\lambda \mathbf{I}_n$ —are added.

Maximum To show that $\max(\text{PH}(P), \text{PH}(Q)) = \text{PH}(P \parallel Q)$, we just have to show that the associated APH representation of $\mathcal{M}_{P \parallel Q}$ is (from Equation (2.20)) $(\vec{e}_1|_{mn+m+n}, \mathbf{G})$, where

$$\mathbf{G} = \begin{bmatrix} \mathbf{A} \oplus \mathbf{B} & \mathbf{I}_m \otimes \vec{B} & \vec{A} \otimes \mathbf{I}_n \\ \mathbf{0} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B} \end{bmatrix}.$$

Let $|Reach(P)| = m + 1$ and $|Reach(Q)| = n + 1$. We fix the value of n , and we perform induction on m .

Induction basis $m = 1$: Process P must be of the form (λ) . Intuitively, a repeated application of SOS rule (5.b) in Table 6.1 results in processes $(\lambda) \parallel Q$, $(\lambda) \parallel Q_1$, \dots , $(\lambda) \parallel Q_n$. However, for each process $(\lambda) \parallel Q_i \in \langle (\lambda) \parallel Q_1, (\lambda) \parallel Q_2, \dots, (\lambda) \parallel Q_n \rangle - \{(\lambda) \parallel Q_n\}$, we can apply SOS rule (5.a), which results in a transition from $(\lambda) \parallel Q_i$ to $\text{stop} \parallel Q_i$ with rate λ . Again, a repeated application of SOS rule (5.b) from $\text{stop} \parallel Q$ results in processes $\text{stop} \parallel Q$, $\text{stop} \parallel Q_1$, \dots , $\text{stop} \parallel Q_n$. This is reflected in the following PH-generator of the associated APH representation of $\mathcal{M}_{(\lambda) \parallel Q}$

$$\left[\begin{array}{ccc} -\lambda \otimes \mathbf{I}_n + [1] \otimes \mathbf{B} & [1] \otimes \vec{B} & \lambda \otimes \mathbf{I}_n \\ \vec{0} & -\lambda & \vec{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B} \end{array} \right] = \left[\begin{array}{ccc} \mathbf{B} - \lambda \mathbf{I}_n & \vec{B} & \lambda \mathbf{I}_n \\ \vec{0} & -\lambda & \vec{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B} \end{array} \right].$$

The first two sets of repeated applications of the SOS rules correspond to the sub-matrix $[\mathbf{B} - \lambda \mathbf{I}_n \quad \vec{B} \quad \lambda \mathbf{I}_n]$. The third set corresponds to the sub-matrix $[\mathbf{0} \quad \mathbf{0} \quad \mathbf{B}]$. The sub-matrix $[\vec{0} \quad -\lambda \quad \vec{0}]$ corresponds to the direct transition from process $(\lambda) \parallel Q_n$ to process stop , which we have excluded above.

Now, we assume that for $m = k$, the associated APH representation of $\mathcal{M}_{P \parallel Q}$ is $(\vec{e}_1|_{kn+k+n}, \mathbf{G})$, where

$$\mathbf{G} = \begin{bmatrix} \mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_k \otimes \mathbf{B} & \mathbf{I}_k \otimes \vec{B} & \vec{A} \otimes \mathbf{I}_n \\ \mathbf{0} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B} \end{bmatrix}.$$

Induction step $m = k + 1$: Take an arbitrary process P' , such that $|Reach(P')| = k + 1$, and let the associated PH-generator of the APH representation of $\mathcal{M}_{P'}$ be \mathbf{A} . Then

process P must be a process that can reach some processes in $Reach(P')$. Let the sum of the rates of all transition from P to the processes in $Reach(P')$ be λ .

Intuitively, a repeated application of SOS rule (5.b) in Table 6.1 results in processes $P\|Q, P\|Q_1, \dots, P\|Q_n$. However, for each process $P\|Q_i \in \langle P\|Q_1, P\|Q_2, \dots, P\|Q_n \rangle - \{P\|Q_n\}$, we can apply SOS rule (5.a), which results in a transition from $P\|Q_i$ to $P'\|Q_i$. This is reflected in the following PH-generator of the associated APH representation of $\mathcal{M}_{P\|Q}$

$$\begin{aligned}
& \begin{bmatrix} \begin{bmatrix} -\lambda & \vec{x} \\ \vec{0} & \mathbf{A} \end{bmatrix} \otimes \mathbf{I}_n + \begin{bmatrix} 1 & \vec{0} \\ \vec{0} & \mathbf{I}_k \end{bmatrix} \otimes \mathbf{B} & \begin{bmatrix} 1 & \vec{0} \\ \vec{0} & \mathbf{I}_k \end{bmatrix} \otimes \vec{B} & \begin{bmatrix} z \\ \vec{A} \end{bmatrix} \otimes \mathbf{I}_n \\ & \mathbf{0} & \begin{bmatrix} -\lambda & \vec{x} \\ \vec{0} & \mathbf{A} \end{bmatrix} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} & \mathbf{B} \end{bmatrix} \\
= & \begin{bmatrix} \begin{bmatrix} -\lambda \otimes \mathbf{I}_n & \vec{x} \otimes \mathbf{I}_n \\ \mathbf{0} & \mathbf{A} \otimes \mathbf{I}_n \end{bmatrix} + \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \otimes \mathbf{I}_k \end{bmatrix} & \begin{bmatrix} \vec{B} & \mathbf{0} \\ \vec{0} & \mathbf{I}_k \otimes \vec{B} \end{bmatrix} & \begin{bmatrix} z \otimes \mathbf{I}_n \\ \vec{A} \otimes \mathbf{I}_n \end{bmatrix} \\ & \mathbf{0} & \begin{bmatrix} -\lambda & \vec{x} \\ \vec{0} & \mathbf{A} \end{bmatrix} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} & \mathbf{B} \end{bmatrix} \\
= & \begin{bmatrix} \mathbf{B} - \lambda \mathbf{I}_n & \vec{x} \otimes \mathbf{I}_n & \vec{B} & \mathbf{0} & z \otimes \mathbf{I}_n \\ \mathbf{0} & \mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_k \otimes \mathbf{B} & \vec{0} & \mathbf{I}_k \otimes \vec{B} & \vec{A} \otimes \mathbf{I}_n \\ \vec{0} & \vec{0} & -\lambda & \vec{x} & \vec{0} \\ \mathbf{0} & \mathbf{0} & \vec{0} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \vec{0} & \mathbf{0} & \mathbf{B} \end{bmatrix} \\
= & \begin{bmatrix} \mathbf{B} - \lambda \mathbf{I}_n & \vec{B} & \vec{x} \otimes \mathbf{I}_n & \mathbf{0} & z \otimes \mathbf{I}_n \\ \vec{0} & -\lambda & \vec{0} & \vec{x} & \vec{0} \\ \mathbf{0} & \vec{0} & \mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_k \otimes \mathbf{B} & \mathbf{I}_k \otimes \vec{B} & \vec{A} \otimes \mathbf{I}_n \\ \mathbf{0} & \vec{0} & \mathbf{0} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \vec{0} & \mathbf{0} & \mathbf{0} & \mathbf{B} \end{bmatrix}.
\end{aligned}$$

The process $P'\|Q$ is represented by the sub-matrix

$$\begin{bmatrix} \mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_k \otimes \mathbf{B} & \mathbf{I}_k \otimes \vec{B} & \vec{A} \otimes \mathbf{I}_n \\ \mathbf{0} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B} \end{bmatrix}.$$

The first two sets of repeated applications of the SOS rules correspond to the sub-matrix

$$[\mathbf{B} - \lambda \mathbf{I}_n \quad \vec{B} \quad \vec{x} \otimes \mathbf{I}_n \quad \mathbf{0} \quad z \otimes \mathbf{I}_n].$$

On the other hand, the sub-matrix

$$[\vec{0} \quad -\lambda \quad \vec{0} \quad \vec{x} \quad \vec{0}]$$

corresponds to the direct transition from process $P\|Q_n = P\|\text{stop}$ to process $P'\|\text{stop}$, which we have excluded above.

Alternative Proof Sketch for Minimum and Maximum KNUTH in [Knu08] showed that the Kronecker sum of the adjacency matrices of two graphs is the adjacency matrix of the Cartesian product graph of the two graphs. Recall that a Cartesian product of two graphs is formed by taking the cross product of the two vertex sets, and any two vertices (u, u') and (v, v') are adjacent in the Cartesian product graph if and only if either

- $u = v$ and u' is adjacent with v' , or
- $u' = v'$ and u is adjacent with v .

This can be easily extended into CTMCs by replacing the adjacency matrices with rate or infinitesimal generator matrices.

The choice (+) operator accomplishes the minimum operation on the absorbing CTMC semantics: Rules (4.a) and (4.b) carry out the Cartesian product of the transient states of \mathcal{M}_P and \mathcal{M}_Q . Rules (4.c) and (4.d), on the other hand, maintain that the Cartesian product of any state with the absorbing state is the absorbing state. By inspecting Equation (2.19), we see that the minimum of $(\vec{\alpha}, \mathbf{A})$ and $(\vec{\beta}, \mathbf{B})$ is formed by the Kronecker sum of \mathbf{A} and \mathbf{B} .

The parallel (\parallel) operator, on the other hand, matches the maximum operation on the absorbing CTMC semantics: Rules (5.a) and (5.b) perform the Cartesian product of the state spaces of \mathcal{M}_P and \mathcal{M}_Q , including the absorbing states of both; at every step whenever P can proceed, it proceeds, and similarly with Q . By inspecting Equation (2.20), we see that the maximum of $(\vec{\alpha}, \mathbf{A})$ and $(\vec{\beta}, \mathbf{B})$ is formed by the Kronecker sum of

$$\begin{bmatrix} \mathbf{A} & \vec{A} \\ \vec{0} & 0 \end{bmatrix} \text{ and } \begin{bmatrix} \mathbf{B} & \vec{B} \\ \vec{0} & 0 \end{bmatrix}.$$

B.4 Lemma 6.18

We provide a proof sketch of the congruence of strong bisimilarity in the line of [vGSS95, HR94]. We will omit the proof of the congruence of weak bisimilarity; it proceeds in a similar fashion.

In the following, we show that strong bisimilarity is a congruence with respect to each operator in the language \mathcal{L} .

Disabling operator We are going to prove that for all $P_1, P_2 \in \mathcal{L}$ and for all $\mu \in \mathbb{R}_{\geq 0}$ and $\lambda \in \mathbb{R}_+$, $P_1 \sim P_2$ implies $(\mu) \triangleleft (\lambda)P_1 \sim (\mu) \triangleleft (\lambda)P_2$. This means that we have to find a strong bisimulation \mathcal{S} such that $P_1 \sim P_2$ implies $((\mu) \triangleleft (\lambda)P_1, (\mu) \triangleleft (\lambda)P_2) \in \mathcal{S}$.

We define

$$\mathcal{R} = \{((\mu) \triangleleft (\lambda)P_1, (\mu) \triangleleft (\lambda)P_2) \mid P_1 \sim P_2 \in \mathcal{L}\}.$$

We note that \mathcal{R} is a reflexive and symmetric relation. Let \mathcal{S} be the transitive closure of $(\mathcal{R} \cup \sim)$, namely

$$\mathcal{S} = (\mathcal{R} \cup \sim)^* = \bigcup_{i=1}^{\infty} (\mathcal{R} \cup \sim)^i.$$

First, we prove that \mathcal{S} is an equivalence relation. The reflexivity of \mathcal{S} follows immediately from the reflexivity of \mathcal{R} and \sim . The transitivity of \mathcal{S} follows from its definition. For the symmetry, assume that $P \mathcal{S} Q$, then $\exists n \in \mathbb{Z}_{\geq 0} : P(\mathcal{R} \cup \sim)^n Q$, namely

$$P(\mathcal{R} \cup \sim)Q_1(\mathcal{R} \cup \sim)Q_2 \cdots Q_{n-1}(\mathcal{R} \cup \sim)Q,$$

for some Q_1, Q_2, \dots, Q_{n-1} . Since each individual relation $(\mathcal{R} \cup \sim)$ is symmetric, it follows that

$$Q(\mathcal{R} \cup \sim)Q_{n-1}(\mathcal{R} \cup \sim)Q_{n-2} \cdots Q_1(\mathcal{R} \cup \sim)P,$$

and hence $\exists n \in \mathbb{Z}_{\geq 0} : Q(\mathcal{R} \cup \sim)^n P$. Thus, \mathcal{S} is symmetric and an equivalence relation.

For our main goal, if we can show that \mathcal{S} is a strong bisimulation, then we are done. For that purpose, we show that

$$\forall (\hat{P}, \hat{Q}) \in \mathcal{S}, \forall C \in \mathcal{L}/\mathcal{S} : \gamma_c(\hat{P}, C) = \gamma_c(\hat{Q}, C). \quad (\text{B.6})$$

Take an arbitrary $(\hat{P}, \hat{Q}) \in \mathcal{S}$ and an arbitrary $C \in \mathcal{L}/\mathcal{S}$; then $\exists n \in \mathbb{Z}_{\geq 0} : \hat{P}(\mathcal{R} \cup \sim)^n \hat{Q}$.

Lemma B.7. *If $\simeq \subseteq \approx$ are two equivalence relations on \mathcal{L} , then*

$$\forall C \in \mathcal{L}/\approx : \exists D_1, D_2, \dots, D_m \in \mathcal{L}/\simeq : C = \dot{\bigcup}_{i=1}^m D_i.$$

To prove Equation (B.6), we do induction on n . For $n = 0$, $\hat{P} \equiv \hat{Q}$ implies $\gamma_c(\hat{P}, C) = \gamma_c(\hat{Q}, C)$. For $n = 1$, let $\hat{P}(\mathcal{R} \cup \sim)\hat{Q}$; we distinguish two cases:

Case 1: $\hat{P} \sim \hat{Q}$. This implies $\gamma_c(\hat{P}, D) = \gamma_c(\hat{Q}, D)$, for all $D \in \mathcal{L}/\sim$. Since $\sim \subseteq (\mathcal{R} \cup \sim)$, by Lemma B.7, C is a disjoint union of some D 's (equivalence classes of \sim). We conclude that $\gamma_c(\hat{P}, C) = \gamma_c(\hat{Q}, C)$.

Case 2: $\hat{P} \mathcal{R} \hat{Q}$. This implies that there are $P, Q \in \mathcal{L}$, such that (\hat{P}, \hat{Q}) is of the form $((\mu) \triangleleft (\lambda)P, (\mu) \triangleleft (\lambda)Q)$ and $P \sim Q$. We observe from SOS rules (2.a) and (2.b) in Table 6.1 that

$$\begin{aligned} \gamma_c((\mu) \triangleleft (\lambda)P, [\text{stop}]_{\mathcal{S}}) &= \lambda = \gamma_c((\mu) \triangleleft (\lambda)Q, [\text{stop}]_{\mathcal{S}}), \text{ and} \\ \gamma_c((\mu) \triangleleft (\lambda)P, [P]_{\mathcal{S}}) &= \mu = \gamma_c((\mu) \triangleleft (\lambda)Q, [Q]_{\mathcal{S}}). \end{aligned}$$

However, since $\sim \subseteq \mathcal{S}$ and $P \sim Q$, $[P]_{\mathcal{S}} = [Q]_{\mathcal{S}}$ and

$$\gamma_c((\mu) \triangleleft (\lambda)Q, [Q]_{\mathcal{S}}) = \gamma_c((\mu) \triangleleft (\lambda)Q, [P]_{\mathcal{S}}).$$

Therefore, $\forall ((\mu) \triangleleft (\lambda)P, (\mu) \triangleleft (\lambda)Q) \in \mathcal{S}, \forall C \in \mathcal{L}/\mathcal{S} : \gamma_c((\mu) \triangleleft (\lambda)P, C) = \gamma_c((\mu) \triangleleft (\lambda)Q, C)$.

For the induction step: assuming Equation (B.6) is valid for $n = m$, $\forall (P', Q') \in \mathcal{S}, \forall C \in \mathcal{L}/\mathcal{S} : P'(\mathcal{R} \cup \sim)^m Q'$ implies $\gamma_c(P', C) = \gamma_c(Q', C)$.

Assume that $\hat{P}(\mathcal{R} \cup \sim)^{m+1}\hat{Q}$. We know then that there exists $\hat{R} \in \mathcal{L}$ such that $\hat{P}(\mathcal{R} \cup \sim)^m \hat{R}(\mathcal{R} \cup \sim)\hat{Q}$. From the step assumption $\forall C \in \mathcal{L}/\mathcal{S} : \gamma_c(\hat{P}, C) = \gamma_c(\hat{R}, C)$. It remains to show whether $\forall C \in \mathcal{L}/\mathcal{S} : \gamma_c(\hat{R}, C) = \gamma_c(\hat{Q}, C)$.

The proof proceeds in the same way as the proof for the basis of the induction $n = 1$.

Sequential operator We are going to prove that for all $P, P_1, P_2 \in \mathcal{L}$, $P_1 \sim P_2$ implies $P_1.P \sim P_2.P$. This means that we have to find a strong bisimulation \mathcal{S} such that $P_1 \sim P_2$ implies $(P_1.P, P_2.P) \in \mathcal{S}$.

We define

$$\mathcal{R} = \{(P_1.P, P_2.P) \mid P_1 \sim P_2 \in \mathcal{L}\}.$$

We note that \mathcal{R} is a reflexive and symmetric relation. Let \mathcal{S} be the transitive closure of $(\mathcal{R} \cup \sim)$, namely

$$\mathcal{S} = (\mathcal{R} \cup \sim)^* = \bigcup_{i=1}^{\infty} (\mathcal{R} \cup \sim)^i.$$

First, we prove that \mathcal{S} is an equivalence relation. The reflexivity of \mathcal{S} follows immediately from the reflexivity of \mathcal{R} and \sim . The transitivity of \mathcal{S} follows from its definition. For the symmetry, assume that $P \mathcal{S} Q$, then $\exists n \in \mathbb{Z}_{\geq 0} : P(\mathcal{R} \cup \sim)^n Q$, namely

$$P(\mathcal{R} \cup \sim)Q_1(\mathcal{R} \cup \sim)Q_2 \cdots Q_{n-1}(\mathcal{R} \cup \sim)Q,$$

for some Q_1, Q_2, \dots, Q_{n-1} . Since each individual relation $(\mathcal{R} \cup \sim)$ is symmetric, it follows that

$$Q(\mathcal{R} \cup \sim)Q_{n-1}(\mathcal{R} \cup \sim)Q_{n-2} \cdots Q_1(\mathcal{R} \cup \sim)P,$$

and hence $\exists n \in \mathbb{Z}_{\geq 0} : Q(\mathcal{R} \cup \sim)^n P$. Thus, \mathcal{S} is symmetric and an equivalence relation.

For our main goal, if we can show that \mathcal{S} is a strong bisimulation, then we are done. For that purpose, we show that

$$\forall (\hat{P}, \hat{Q}) \in \mathcal{S}, \forall C \in \mathcal{L}/\mathcal{S} : \gamma_c(\hat{P}, C) = \gamma_c(\hat{Q}, C). \quad (\text{B.7})$$

Take an arbitrary $(\hat{P}, \hat{Q}) \in \mathcal{S}$ and an arbitrary $C \in \mathcal{L}/\mathcal{S}$; then $\exists n \in \mathbb{Z}_{\geq 0} : \hat{P}(\mathcal{R} \cup \sim)^n \hat{Q}$.

To prove Equation (B.7), we do induction on n . For $n = 0$, $\hat{P} \equiv \hat{Q}$ implies $\gamma_c(\hat{P}, C) = \gamma_c(\hat{Q}, C)$. For $n = 1$, let $\hat{P}(\mathcal{R} \cup \sim)\hat{Q}$; we distinguish two cases:

Case 1: $\hat{P} \sim \hat{Q}$. This implies $\gamma_c(\hat{P}, D) = \gamma_c(\hat{Q}, D)$, for all $D \in \mathcal{L}/\sim$. Since $\sim \subseteq (\mathcal{R} \cup \sim)$, by Lemma B.7, C is a disjoint union of some D 's (equivalence classes of \sim). We conclude that $\gamma_c(\hat{P}, C) = \gamma_c(\hat{Q}, C)$.

Case 2: $\hat{P}\mathcal{R}\hat{Q}$. This implies that there are $P, Q, R \in \mathcal{L}$, such that (\hat{P}, \hat{Q}) is of the form $(P.R, Q.R)$ and $P \sim Q$.

We take an arbitrary $(P.R, Q.R) \in \mathcal{S}$. According to the SOS rule (3.a) in Table 6.1, $C \in \mathcal{L}/\mathcal{S}$ must contain a process of the form $P'.R$ or $Q'.R$, where $P' \sim Q'$. On the other hand, by SOS rule (3.b), $C \in \mathcal{L}/\mathcal{S}$ must contain a process of the form R . However, since $\sim \subseteq \mathcal{S}$ and $P' \sim Q'$, $[P']_{\mathcal{S}} = [Q']_{\mathcal{S}}$ and

$$\gamma_c(P.R, [P'.R]_{\mathcal{S}}) = \gamma_c(Q.R, [Q'.R]_{\mathcal{S}}).$$

In the same fashion, we obtain

$$\gamma_c(P.R, [R]_{\mathcal{S}}) = \gamma_c(Q.R, [R]_{\mathcal{S}}).$$

Therefore, $\forall (P.R, Q.R) \in \mathcal{S}, \forall C \in \mathcal{L}/\mathcal{S} : \gamma_c(P.R, C) = \gamma_c(Q.R, C)$.

For the induction step: assuming Equation (B.7) is valid for $n = m$, $\forall (P', Q') \in \mathcal{S}, \forall C \in \mathcal{L}/\mathcal{S} : P'(\mathcal{R} \cup \sim)^m Q'$ implies $\gamma_c(P', C) = \gamma_c(Q', C)$.

Assume that $\hat{P}(\mathcal{R} \cup \sim)^{m+1}\hat{Q}$. Hence we know that there exists $\hat{R} \in \mathcal{L}$ such that $\hat{P}(\mathcal{R} \cup \sim)^m \hat{R}(\mathcal{R} \cup \sim)\hat{Q}$. From the step assumption $\forall C \in \mathcal{L}/\mathcal{S} : \gamma_c(\hat{P}, C) = \gamma_c(\hat{R}, C)$. It remains to show whether $\forall C \in \mathcal{L}/\mathcal{S} : \gamma_c(\hat{R}, C) = \gamma_c(\hat{Q}, C)$.

The proof proceeds in the same way as the proof for the basis of the induction $n = 1$.

We can show that for all $P, P_1, P_2 \in \mathcal{L}$, $P_1 \sim P_2$ implies $P.P_1 \sim P.P_2$ in a similar line of proof.

Choice operator We are going to prove that for all $P, P_1, P_2 \in \mathcal{L}$, $P_1 \sim P_2$ implies $P + P_1 \sim P + P_2$. In the same way as for the previous operators, this means that we have to find a strong bisimulation \mathcal{S} such that $P_1 \sim P_2$ implies $(P + P_1, P + P_2) \in \mathcal{S}$.

We define

$$\mathcal{R} = \{(P + P_1, P + P_2) \mid P_1 \sim P_2 \in \mathcal{L}, P \in \mathcal{L}\}.$$

Let \mathcal{S} be the reflexive closure of \mathcal{R} . If we can show that \mathcal{S} is a strong bisimulation, then we are done. For that purpose, we show that for all $Q, R, S \in \mathcal{L}$, $R \sim S$ implies

$$\forall C \in \mathcal{L}/\mathcal{S} : \gamma_c(Q + R, C) = \gamma_c(Q + S, C).$$

We take an arbitrary $(Q + R, Q + S) \in \mathcal{S}$ and an arbitrary $C \in \mathcal{L}/\mathcal{S}$. According to the SOS rules (4.a)–(4.d) in Table 6.1, C must contain a process of the form $(\hat{Q} + \hat{R})$ or $C = \{\text{stop}\}$.

Assume that $(\hat{Q} + \hat{R}) \in C$, from SOS rules (4.a) and (4.b), we obtain

$$\begin{aligned} \gamma_c(Q + R, [\hat{Q} + \hat{R}]_{\mathcal{S}}) &= \sum_{Q'+R \in [\hat{Q} + \hat{R}]_{\mathcal{S}}} \sum_{(\lambda, w) \in \{(\lambda, w) \mid Q+R \xrightarrow{\lambda, w} Q'+R\}} \lambda \\ &+ \sum_{Q+R' \in [\hat{Q} + \hat{R}]_{\mathcal{S}}} \sum_{(\lambda, w) \in \{(\lambda, w) \mid Q+R \xrightarrow{\lambda, w} Q+R'\}} \lambda. \end{aligned}$$

From SOS rules (4.c) and (4.d), on the other hand,

$$\begin{aligned} \gamma_c(Q + R, [\text{stop}]_{\mathcal{S}}) &= \sum_{(\lambda, w) \in \{(\lambda, w) \mid Q \xrightarrow{\lambda, w} \text{stop}\}} \lambda + \sum_{(\lambda, w) \in \{(\lambda, w) \mid R \xrightarrow{\lambda, w} \text{stop}\}} \lambda, \\ &= \gamma_c(Q, \text{stop}) + \gamma_c(R, [\text{stop}]_{\sim}). \end{aligned}$$

Observing that $Q + R' \in [\hat{Q} + \hat{R}]$ if and only if $R' \in [\hat{R}]_{\sim}$, we have

$$\begin{aligned} \gamma_c(Q + R, [\hat{Q} + \hat{R}]_{\mathcal{S}}) &= \sum_{(\lambda, w) \in \{(\lambda, w) \mid Q \xrightarrow{\lambda, w} Q'\}} \lambda + \sum_{R' \in [\hat{R}]_{\sim}} \sum_{(\lambda, w) \in \{(\lambda, w) \mid R \xrightarrow{\lambda, w} R'\}} \lambda, \\ &= \gamma_c(Q, \hat{Q}) + \gamma_c(R, [\hat{R}]_{\sim}). \end{aligned}$$

By analogous argument we have $\gamma_c(Q + S, [\hat{Q} + \hat{R}]_{\mathcal{S}}) = \gamma_c(Q, \hat{Q}) + \gamma_c(S, [\hat{R}]_{\sim})$ and $\gamma_c(Q + S, [\text{stop}]_{\mathcal{S}}) = \gamma_c(Q, \text{stop}) + \gamma_c(S, [\text{stop}]_{\sim})$.

Using the assumption that $R \sim S$, we have $\gamma_c(R, [\hat{R}]_{\sim}) = \gamma_c(S, [\hat{R}]_{\sim})$, and also $\gamma_c(R, [\text{stop}]_{\sim}) = \gamma_c(S, [\text{stop}]_{\sim})$. Therefore, $\gamma_c(Q + R, [\hat{Q} + \hat{R}]_{\mathcal{S}}) = \gamma_c(Q + S, [\hat{Q} + \hat{R}]_{\mathcal{S}})$ and $\gamma_c(Q + R, [\text{stop}]_{\mathcal{S}}) = \gamma_c(Q + S, [\text{stop}]_{\mathcal{S}})$.

We can show that for all $P, P_1, P_2 \in \mathcal{L}$, $P_1 \sim P_2$ implies $P_1 + P \sim P_2 + P$ in a similar line of proof.

Parallel operator We are going to prove that for all $P, P_1, P_2 \in \mathcal{L}$, $P_1 \sim P_2$ implies $P \parallel P_1 \sim P \parallel P_2$. In the same way as for the previous operators, this means that we have to find a strong bisimulation \mathcal{S} such that $P_1 \sim P_2$ implies $(P \parallel P_1, P \parallel P_2) \in \mathcal{S}$.

We define

$$\mathcal{R} = \{(P \parallel P_1, P \parallel P_2) \mid P_1 \sim P_2 \in \mathcal{L}, P \in \mathcal{L}\}.$$

Let \mathcal{S} be the reflexive closure of \mathcal{R} . If we can show that \mathcal{S} is a strong bisimulation, then we are done. For that purpose, we show that for all $Q, R, S \in \mathcal{L}$, $R \sim S$ implies

$$\forall C \in \mathcal{L}/\mathcal{S} : \gamma_c(Q \parallel R, C) = \gamma_c(Q \parallel S, C).$$

We take an arbitrary $(Q \parallel R, Q \parallel S) \in \mathcal{S}$ and an arbitrary $C \in \mathcal{L}/\mathcal{S}$. According to the SOS rules (5.a) and (5.b) in Table 6.1, C must contain a process of the form $(\hat{Q} \parallel \hat{R})$.

Assume that $(\hat{Q} \parallel \hat{R}) \in C$, from SOS rules (5.a) and (5.b), we obtain

$$\begin{aligned} \gamma_c(Q \parallel R, [\hat{Q} \parallel \hat{R}]_{\mathcal{S}}) &= \sum_{Q' \parallel R \in [\hat{Q} \parallel \hat{R}]_{\mathcal{S}}} \sum_{(\lambda, w) \in \{(\lambda, w) \mid Q \parallel R \xrightarrow{\lambda, w} Q' \parallel R\}} \lambda \\ &+ \sum_{Q \parallel R' \in [\hat{Q} \parallel \hat{R}]_{\mathcal{S}}} \sum_{(\lambda, w) \in \{(\lambda, w) \mid Q \parallel R \xrightarrow{\lambda, w} Q \parallel R'\}} \lambda. \end{aligned}$$

Observing that $Q \parallel R' \in [\hat{Q} \parallel \hat{R}]$ if and only if $R' \in [\hat{R}]_{\sim}$, we have

$$\begin{aligned} \gamma_c(Q \parallel R, [\hat{Q} \parallel \hat{R}]_{\mathcal{S}}) &= \sum_{(\lambda, w) \in \{(\lambda, w) \mid Q \xrightarrow{\lambda, w} Q'\}} \lambda + \sum_{R' \in [\hat{R}]_{\sim}} \sum_{(\lambda, w) \in \{(\lambda, w) \mid R \xrightarrow{\lambda, w} R'\}} \lambda, \\ &= \gamma_c(Q, \hat{Q}) + \gamma_c(R, [\hat{R}]_{\sim}). \end{aligned}$$

By analogous argument we have $\gamma_c(Q \parallel S, [\hat{Q} \parallel \hat{R}]_{\mathcal{S}}) = \gamma_c(Q, \hat{Q}) + \gamma_c(S, [\hat{R}]_{\sim})$. Using the assumption that $R \sim S$, we have $\gamma_c(R, [\hat{R}]_{\sim}) = \gamma_c(S, [\hat{R}]_{\sim})$. Therefore,

$$\gamma_c(Q_1 \parallel R, [\hat{Q} \parallel \hat{R}]_{\mathcal{S}}) = \gamma_c(Q_2 \parallel R, [\hat{Q} \parallel \hat{R}]_{\mathcal{S}}).$$

We can show that for all $P, P_1, P_2 \in \mathcal{L}$, $P_1 \sim P_2$ implies $P_1 \parallel P \sim P_2 \parallel P$ in a similar line of proof.

Bibliography

- [AB96] Søren Asmussen and Mogen Bladt. *Matrix-Analytic Methods in Stochastic Models*, chapter Renewal theory and queueing algorithms matrix-exponential distributions, pages 313–341. CRC Press, 1996.
- [ACW92] Joseph Abate, Gagan L. Choudhury, and Ward Whitt. Calculation of the GI/G/1 waiting time distribution and its cumulants from Pollaczek’s formulas. *Archiv für Elektronik und Übertragungstechnik*, 47(5/6):311–321, 1992.
- [AFV01] Luca Aceto, Wan J. Fokkink, and Chris Verhoef. *Handbook of Process Algebra*, chapter Structural operational semantics, pages 197–292. Elsevier, 2001.
- [AL82] David Assaf and Benny Levikson. Closure of phase-type distributions under operations arising in reliability theory. *Annals of Probability*, 10(1):265–269, 1982.
- [ANO96] Søren Asmussen, O. Nerman, and O. Olsson. Fitting phase-type distributions via the EM algorithm. *Scandinavian Journal of Statistics*, 23:419–441, 1996.
- [AO98] Søren Asmussen and Colm A. O’Cinneide. *Encyclopedia of Statistical Science Update*, volume 2, chapter Matrix-exponential distributions, pages 435–440. John Wiley and Sons, 1998.
- [AS87] David Aldous and Larry Shepp. The least variable phase-type distribution is erlang. *Communications in Statistics: Stochastic Models*, 3:467–473, 1987.
- [Asm92] Søren Asmussen. Phase-type representations in random walk and queueing problems. *Annals of Probability*, 20(2):772–789, 1992.
- [ASSB00] Adnan Aziz, Kumud Sanwal, Vigyan Singhal, and Robert Brayton. Model-checking continuous-time Markov chains. *ACM Transactions on Computational Logic*, 1(1):162–170, 2000.
- [BB87] Tommaso Bolognesi and Ed Brinksma. Introduction to the ISO specification language LOTOS. *Computer Networks*, 14:25–59, 1987.
- [BC88] Gérard Boudol and Ilaria Castellani. A non-interleaving semantics for CCS based on proved transitions. *Fundamenta Informaticae*, 11:433–452, 1988.

- [BC92] Andrea Bobbio and Aldo Cumani. ML estimation of the parameters of a PH distribution in triangular canonical form. In *Computer Performance Evaluation*, pages 33–46. Elsevier, Inc., 1992.
- [BCDK00] Peter Buchholz, Gianfranco Ciardo, Susanna Donatelli, and Peter Kemper. Complexity of memory-efficient Kronecker operations with applications to the solution of Markov models. *INFORMS Journal on Computing*, 12(3):203–222, 2000.
- [BCS07a] Hichem Boudali, Pepijn Crouzen, and Mariëlle Stoelinga. A compositional semantics for dynamic fault trees in terms of interactive Markov chains. In *Automated Technology for Verification and Analysis, 5th International Symposium, ATVA 2007, Tokyo, Japan, October 22-25, 2007, Proceedings*, volume 4762 of *Lecture Notes in Computer Science*, pages 441–456. Springer, 2007.
- [BCS07b] Hichem Boudali, Pepijn Crouzen, and Mariëlle Stoelinga. Dynamic fault tree analysis using input/output interactive Markov chains. In *The 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2007, 25-28 June 2007, Edinburgh, UK, Proceedings*, pages 708–717. IEEE Computer Society, 2007.
- [BG96] Marco Bernardo and Roberto Gorrieri. Extended Markovian process algebra. In *CONCUR '96, Concurrency Theory, 7th International Conference, Pisa, Italy, August 26-29, 1996, Proceedings*, volume 1119 of *Lecture Notes in Computer Science*, pages 315–330. Springer, 1996.
- [BG02] Mario Bravetti and Roberto Gorrieri. The theory of interactive generalized semi-markov processes. *Theoretical Computer Science*, 282(1):5–32, 2002.
- [BH02] Henrik C. Bohnenkamp and Boudewijn R. Haverkort. The mean value of the maximum. In *Process Algebra and Probabilistic Methods, Performance Modeling and Verification, Second Joint International Workshop PPM-PROBMIV 2002, Copenhagen, Denmark, July 25-26, 2002, Proceedings*, volume 2399 of *Lecture Notes in Computer Science*, pages 37–56. Springer, 2002.
- [BHHK03] Christel Baier, Boudewijn R. Haverkort, Holger Hermanns, and Joost-Pieter Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(6):524–541, 2003.
- [BHM87] Robert F. Botta, Carl M. Harris, and William G. Marchal. Characterizations of generalized hyperexponential distribution functions. *Stochastic Models*, 3(1):115–148, 1987.
- [BHST03] Andrea Bobbio, András Horváth, Marco Scarpa, and Miklós Telek. Acyclic discrete phase type distributions: properties and a parameter estimation algorithm. *Performance Evaluation*, 54(1):1–32, 2003.

- [BHT04] Andrea Bobbio, András Horváth, and Miklós Telek. The scale factor: a new degree of freedom in phase-type approximation. *Performance Evaluation*, 56(1-4):121–144, 2004.
- [BKHW05] Christel Baier, Joost-Pieter Katoen, Holger Hermanns, and Verena Wolf. Comparative branching-time semantics for Markov chains. *Information and Computation*, 200(2):149–214, 2005.
- [Bra02] Mario Bravetti. Revisiting interactive Markov chains. *Electronic Notes in Theoretical Computer Science*, 68(5), 2002.
- [Bre07] Freimut Brenner. Absorbing joint markov chains: Exploiting compositionality for cumulative measures. In *Proceedings of the ASMTA 2007, 14th International Conference on Analytical and Stochastic Modelling Techniques and Applications, Prague, Czech Republic*, pages 128–143, 2007.
- [BT94] Andrea Bobbio and Miklós Telek. A benchmark for PH estimation algorithm: Results for acyclic PH. *Stochastic Models*, 10:661–667, 1994.
- [Buc94] Peter Buchholz. Exact and ordinary lumpability in finite Markov chains. *Journal of Applied Probability*, 31:59–75, 1994.
- [BW90] Jos C. M. Baeten and W. Peter Weijland. *Process algebra*. Cambridge University Press, 1990.
- [CC93] Christian Commault and Jean-Paul Chemla. On dual and minimal phase-type representations. *Communications in Statistics: Stochastic Models*, 9(3):421–434, 1993.
- [CC96] Christian Commault and Jean-Paul Chemla. An invariant of representations of phase-type distributions and some applications. *Journal of Applied Probability*, 33:368–381, 1996.
- [CHZ08] Pepijn Crouzen, Holger Hermanns, and Lijun Zhang. On the minimisation of acyclic models. In *CONCUR 2008 - Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, August 19-22, 2008. Proceedings*, volume 5201 of *Lecture Notes in Computer Science*, pages 295–309. Springer, 2008.
- [CKU92] Srinivas R. Chakravarthy, A. Krishnamoorthy, and P. V. Ushakumari. A k-out-of-n reliability system with an unreliable server and phase type repairs and services: the (N,T) policy. *Journal of Applied Mathematics and Stochastic Analysis*, 14(4):361–380, 1992.
- [CM02] Christian Commault and Ștefăniță Mocanu. A generic property of phase-type representations. *Journal of Applied Probability*, 39:775–785, 2002.
- [CM03] Christian Commault and Ștefăniță Mocanu. Phase-type distributions and representations: Some results and open problems for system theory. *International Journal of Control*, 76(6):566–580, 2003.

- [Cox55] David R. Cox. A use of complex probabilities in the theory of stochastic processes. *Proceedings of the Cambridge Philosophical Society*, 51(2):313–319, 1955.
- [CR91] Srinivas R. Chakravarthy and K. V. Ravi. *A Stochastic Model for a Computer Communication Network Node with Phase Type Timeout Periods*, chapter 14 in *Numerical Solutions of Markov Chains*, pages 261–286. Marcel Dekker, 1991.
- [Cum82] Aldo Cumani. Canonical representation of homogeneous Markov processes modelling failure time distributions. *Microelectronics and Reliability*, 2(3):583–602, 1982.
- [DBB92] Joanne B. Dugan, Salvatore J. Bavuso, and Mark A. Boyd. Dynamic fault-tree models for fault-tolerant computer systems. *IEEE Transactions on Reliability*, 41(3):363–377, 1992.
- [DBB93] Joanne B. Dugan, Salvatore J. Bavuso, and Mark A. Boyd. Fault trees and Markov models for reliability analysis of fault-tolerant digital systems: Reliability performance, analysis, and evaluation of programmable electronic systems, with emphasis on chemical process applications. *Reliability engineering & systems safety*, 39(3):291–307, 1993.
- [DD96] Joanne B. Dugan and Stacy A. Doyle. New results in fault-tree analysis. In *Proceedings Annual Reliability and Maintainability Symposium Tutorial Notes*, pages 1–23, Las Vegas, NV, 1996.
- [DHS03] Salem Derisavi, Holger Hermanns, and William H. Sanders. Optimal state-space lumping in Markov chains. *Information Processing Letters*, 87(6):309–315, 2003.
- [DK05] Pedro R. D’Argenio and Joost-Pieter Katoen. A theory of stochastic systems. part ii: Process algebra. *Information and Computation*, 203(1):39–74, 2005.
- [DL82] Michel Dehon and Guy Latouche. A geometric interpretation of the relations between the exponential and generalized Erlang distributions. *Advances in Applied Probability*, 14(4):885–897, 1982.
- [Fac03] Mark Fackrell. *Characterization of matrix-exponential distributions*. PhD thesis, University of Adelaide, Australia, 2003.
- [FG88] Bennett L. Fox and Peter W. Glynn. Computing Poisson probabilities. *Communications of the ACM*, 31(4):440–445, 1988.
- [GM84] Donald Gross and Douglas R. Miller. The randomization technique as a modeling tool and solution procedure for transient markov processes. *Operations Research*, 32(2):343–361, 1984.
- [GMP08] GMP Team. *The GNU Multiple Precision Arithmetic Library*, 2008. <http://gmpplib.org/>.

- [Gov07] Rob M.P. Goverde. Railway timetable stability analysis using Max-Plus system theory. *Transportation Research Part B: Methodological*, 41(2):179–201, 2007. Advanced Modelling of Train Operations in Stations and Networks.
- [Gra91] Winfried K. Grassmann. *Numerical Solution of Markov Chains*, chapter Finding Transient Solutions in Markovian Event Systems through Randomization, pages 357–371. Marcel Dekker Inc., 1991.
- [GyK00] R. Goverde and Gerardo Soto y Koelemeijer. *Performance Evaluation of Periodic Railway Timetables: Theory and Algorithms*, volume 2 of *TRAIL Studies in Transportation Science*. Delft University Press, 2000.
- [Hav98] Boudewijn R. Haverkort. *Performance Evaluation of Computer Communication Systems: A Model-Based Approach*. John Wiley & Sons, 1998.
- [Her01] Holger Hermanns. Construction and verification of performance and reliability models. *Bulletin of the EATCS*, 74:135–153, 2001.
- [Her02] Holger Hermanns. *Interactive Markov Chains: The Quest for Quantified Quality*, volume 2428 of *Lecture Notes in Computer Science*. Springer, 2002.
- [Hil96] Jane Hillston. *A compositional approach to performance modelling*. Cambridge University Press, 1996.
- [HK92] Ernest J. Henley and Hiromitsu Kumamoto. *Probabilistic Risk Assessment: Reliability Engineering, Design, and Analysis*. IEEE Press, 1992.
- [HK01] Jane Hillston and Leïla Kloul. An efficient Kronecker representation for PEPA models. In *Process Algebra and Probabilistic Methods, Performance Modeling and Verification: Joint International Workshop, PAPM-PROBMIV 2001, Aachen, Germany, September 12-14, 2001, Proceedings*, volume 2165 of *Lecture Notes in Computer Science*, pages 120–135. Springer, 2001.
- [HKMKS03] Holger Hermanns, Joost-Pieter Katoen, Joachim Meyer-Kayser, and Markus Siegle. A tool for model-checking Markov chains. *STTT*, 4(2):153–172, 2003.
- [HKN⁺03] Holger Hermanns, Marta Z. Kwiatkowska, Gethin Norman, David Parker, and Markus Siegle. On the use of MTBDDs for performability analysis and verification of stochastic systems. *Journal of Logic and Algebraic Programming*, 56(1-2):23–67, 2003.
- [HKNP06] Andrew Hinton, Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM: A tool for automatic verification of probabilistic systems. In Holger Hermanns and Jens Palsberg, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 12th International Conference, TACAS 2006 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2006, Vienna, Austria, March 25 - April*

- 2, 2006, *Proceedings*, volume 3920 of *Lecture Notes in Computer Science*, pages 441–444. Springer, 2006.
- [HLD88] R. E. Harper, J. H. Lala, and J. J. Deyst. Fault tolerant parallel processors architecture overview. In *Proceedings of the 18th Symposium on Fault Tolerant Computing*, pages 252–257, 1988.
- [Hoa78] Charles A. R. Hoare. Communicating sequential processes. *Communication of ACM*, 21(8):666–677, 1978.
- [HR94] Holger Hermanns and Michael Rettelbach. Syntax, semantics, equivalences and axioms for MTIPP. In *Proceedings of the 2nd Workshop on Process Algebras and Performance Modelling (PAPM '94)*, pages 71–87, 1994.
- [HZ06a] Qi-Ming He and Hanqin Zhang. PH-invariant polytopes and Coxian representations of phase type distributions. *Stochastic Models*, 22(3):383–409, 2006.
- [HZ06b] Qi-Ming He and Hanqin Zhang. Spectral polynomial algorithms for computing bi-diagonal representations for phase type distributions and matrix-exponential distributions. *Stochastic Models*, 2(2):289–317, 2006.
- [HZ07a] Qi-Ming He and Hanqin Zhang. An Algorithm for Computing Minimal Coxian Representations. *INFORMS Journal on Computing*, page ijoc.1070.0228, 2007.
- [HZ07b] Qi-Ming He and Hanqin Zhang. On matrix exponential distributions. *Advances in Applied Probability*, 39(1):271–292, 2007.
- [Jen53] A. Jensen. Markoff chains as an aid in the study of Markoff processes. *Skandinavisk Aktuarietidskrift*, 36:87–91, 1953.
- [JKO⁺08] David N. Jansen, Joost-Pieter Katoen, Marcel Oldenkamp, Mariëlle Stoelinga, and Ivan S. Zapreev. How fast and fat is your probabilistic model checker? an experimental performance comparison. In Karen Yorav, editor, *Hardware and Software: Verification and Testing, Third International Haifa Verification Conference, HVC 2007, Haifa, Israel, October 23-25, 2007, Proceedings*, volume 4899 of *Lecture Notes in Computer Science*, pages 69–85. Springer, 2008.
- [JT88] Mary A. Johnson and Michael R. Taaffe. The denseness of phase distributions. Purdue School of Industrial Engineering Research Memoranda 88-20, Purdue University, 1988.
- [Kel79] Frank P. Kelly. *Reversibility and Stochastic Networks*. Wiley, 1979.
- [KKZ05] Joost-Pieter Katoen, Maneesh Khattri, and Ivan S. Zapreev. A Markov reward model checker. In *Second International Conference on the Quantitative Evaluation of Systems (QEST 2005), 19-22 September 2005, Torino, Italy*, pages 243–244. IEEE Computer Society, 2005.

- [KKZJ07] Joost-Pieter Katoen, Tim Kemna, Ivan S. Zapreev, and David N. Jansen. Bisimulation minimisation mostly speeds up probabilistic model checking. In Orna Grumberg and Michael Huth, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 13th International Conference, TACAS 2007, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2007 Braga, Portugal, March 24 - April 1, 2007, Proceedings*, volume 4424 of *Lecture Notes in Computer Science*, pages 87–101. Springer, 2007.
- [Knu08] Donald E. Knuth. *Fascicle 0: Introduction to Combinatorial Algorithms and Boolean Functions*, volume 4A of *The Art of Computer Programming*. Addison-Wesley, 2008.
- [KS76] John G. Kemeny and J. Laurie Snell. *Finite Markov Chains*. Springer, 1976.
- [KSH03] Rachid El-Abdouni Khayari, Ramin Sadre, and Boudewijn R. Haverkort. Fitting world-wide web request traces with the EM-algorithm. *Performance Evaluation*, 52(2-3):175–191, 2003.
- [Lin74] Ching-Tai Lin. Structural controllability. *IEEE Transactions on Automatic Control*, AC-19(3):201–208, 1974.
- [LR89] Lester Lipsky and Vaidyanat Ramaswami. A unique minimal representation of Coxian service centres. Technical report, Department of Computer Science and Engineering, University of Nebraska, 1989.
- [LR99] Guy Latouche and Vaidyanat Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM, 1999.
- [MC99] Ștefăniță Mocanu and Christian Commault. Sparse representation of phase-type distributions. *Communications in Statistics: Stochastic Models*, 15(4):759–778, 1999.
- [MDCS98] Ragavan Manian, Joanne B. Dugan, David Coppit, and Kevin J. Sullivan. Combining various solution techniques for dynamic fault tree analysis of computer systems. In *3rd IEEE International Symposium on High-Assurance Systems Engineering (HASE '98), 13-14 November 1998, Washington, D.C, USA, Proceedings*, pages 21–28. IEEE Computer Society, 1998.
- [Mey04] Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2004.
- [Mil95] Robin Milner. *Communication and concurrency*. Prentice Hall International (UK) Ltd., 1995.
- [Mil99] Robin Milner. *Communicating and mobile systems: the π -calculus*. Cambridge University Press, New York, NY, USA, 1999.
- [ML78] Cleve B. Moler and Charles F. Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, 20(4):801–836, 1978.

- [ML03] Cleve B. Moler and Charles F. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, 2003.
- [MM07] Ludolf E. Meester and Sander Muns. Stochastic delay propagation in railway networks and phase-type distributions. *Transportation Research Part B: Methodological*, 41(2):218–230, 2007. Advanced Modelling of Train Operations in Stations and Networks.
- [MO92] Robert S. Maier and Colm A. O’Cinneide. Closure characterisation of phase-type distributions. *Journal of Applied Probability*, 29(1):92–103, 1992.
- [MPvdL96] Kenneth Mitchell, Jerry Place, and Appie van de Liefvoort. Analytic modeling with matrix exponential distributions. *Simulation Councils Proceedings Series*, 28(1):201–204, 1996.
- [MRT87] Raymond A. Marie, Andrew L. Reibman, and Kishor S. Trivedi. Transient analysis of acyclic Markov chains. *Performance Evaluation*, 7(3):175–194, 1987.
- [Neu81] Marcel F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Dover, 1981.
- [Nic89] Victor F. Nicola. Lumping in markov reward processes. Technical report, Res. Rep. 14719, IBM, 1989.
- [O’C89] Colm A. O’Cinneide. On non-uniqueness of representations of phase-type distributions. *Communications in Statistics: Stochastic Models*, 5(2):247–259, 1989.
- [O’C90] Colm A. O’Cinneide. Characterization of phase-type distributions. *Communications in Statistics: Stochastic Models*, 6(1):1–57, 1990.
- [O’C91] Colm A. O’Cinneide. Phase-type distributions and invariant polytopes. *Advances in Applied Probability*, 23(43):515–535, 1991.
- [O’C93] Colm A. O’Cinneide. Triangular order of triangular phase-type distributions. *Communications in Statistics: Stochastic Models*, 9(4):507–529, 1993.
- [PA91] Brigitte Plateau and Karim Atif. Stochastic automata network for modeling parallel systems. *IEEE Transactions on Software Engineering*, 17(10):1093–1108, 1991.
- [Plo04] Gordon D. Plotkin. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60-61:17–140, 2004.
- [PT87] Robert Paige and Robert E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–989, 1987.
- [Roc70] R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

- [Ros07] Sheldon M. Ross. *Introduction to Probability Models*. Elsevier, Inc., 9th edition, 2007.
- [RSS08] Martin Riedl, Johann Schuster, and Markus Siegle. Recent extensions to the stochastic process algebra tool caspa. In *Fifth International Conference on the Quantitative Evaluation of Systems (QEST 2008), 14-17 September 2008, Saint-Malo, France*, pages 113–114. IEEE Computer Society, 2008.
- [Rud87] Walter Rudin. *Real and Convex Analysis*. McGraw-Hill, 1987.
- [Ste94] William J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.
- [TBT06] Axel Thümmler, Peter Buchholz, and Miklós Telek. A novel approach for phase-type fitting with the EM algorithm. *IEEE Transactions on Dependable and Secure Computing*, 3(3):245–258, 2006.
- [Tij07] Henk Tijms. *Understanding Probability: Chance Rules in Everyday Life*. Cambridge University Press, 2007.
- [Tri02] Kishor S. Trivedi. *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. John Wiley & Sons, Inc., 2nd edition, 2002.
- [vGSS95] Rob J. van Glabbeek, Scott A. Smolka, and Bernhard Steffen. Reactive, generative and stratified models of probabilistic processes. *Information and Computation*, 121(1):59–80, 1995.
- [Wei51] Waloddi Weibull. A statistical distribution function of wide applicability. *Journal of Applied Mechanics*, 18:293–297, 1951.
- [Wol08] Verena Wolf. *Equivalences on Phase-Type Processes*. PhD thesis, University of Mannheim, Germany, 2008.