Formalization of Continuous Time Markov Chains with Applications in Queueing Theory

Donia Chaouch

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Electrical & Computer Engineering)

Concordia University

Montréal, Québec, Canada

March 2015

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Donia Chaouch**

Entitled: **Formalization of Continuous Time Markov Chains with Applications in Queueing Theory**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical & Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

_____ Dr. Zahangir Kabir (Chair)

_____ Dr. Nizar Bouguila (Examiner)

_____ Dr. Walaa Hamouda (Examiner)

_____ Dr. Sofiène Tahar (Supervisor)

Approved by _____

Chair of the ECE Department

_____ 2015 _____

Dean of Engineering

Abstract

Formalization of Continuous Time Markov Chains with Applications in Queueing Theory

Donia Chaouch

The performance analysis of engineering systems have become very critical due to their usage in safety and mission critical domains such as military and biomedical devices. Such an analysis is often carried out based on the Markovian (or Markov Chains based) models of underlying software and hardware components. Furthermore, some important properties can only be captured by queueing theory which involves Markov Chains with continuous time behavior. Classically, the analysis of such models has been performed using paper-and-pencil based proofs and computer simulation, both of which cannot provide perfectly accurate results due to the error-prone nature of manual proofs and the non-exhaustive nature of simulation. Recently, model checking based formal methods have also been used to analyze Markovian and queuing systems. However, such an approach is only applicable for small systems and cannot certify generic properties due to the sate-space explosion problem.

In this thesis, we propose to use higher-order-logic theorem proving as a complementary approach to conduct the formal analysis of queueing systems. To this aim, we present the higher-order-logic formalization of the Poisson process which is the foremost step to model queueing systems. We also verify some of its classical properties such as exponentially distributed inter-arrival time, memoryless property and independent and stationary increments. Moreover, we used the formalization of the Poisson process to model and verify the error probability of a generic optical communication system. Then we present the formalization of Continuous-Time Markov

Chains along with the Birth-Death process. Lastly, we demonstrate the utilization of our developed infrastructure by presenting the formalization of an M/M/1 queue which is widely used to model telecommunication systems. We also formally verified the generic result about the average waiting time for any given queue.

To My Parents, My Sister, My Brother and My beloved ones.

Acknowledgments

First and above all, I praise ALLAH, the almighty for providing me this opportunity and granting me the capability to proceed successfully. I attribute the level of my Master's degree to the assistance and guidance of several people. Therefore, I would like to offer my sincere thanks to all of them.

I offer my sincerest gratitude to my supervisor, Prof. Sofiène Tahar, who has supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. I deeply appreciate the enormous amount of help and time he offered me. Special thanks goes to Dr. Osman Hasan for his excellent guidance and care. He always provided me with much valuable advices and insightful discussions. I greatly appreciate their assistance and their spiritual supports during this journey.

The members of the Hardware Verification Group have contributed immensely to my personal and professional time at Concordia University. The group has been a great source of friendship as well as good advice. Though I do not list all their names, I have had the pleasure to work with or alongside them. Exceptionally, I would like to acknowledge honorary group members Muhammad Umair Siddique, Sanaz Khan Afshar, Ghassen Helali, Dr. Li Ya Liu and Maissa Elleuch, who as good friends, were always willing to help and give their best suggestions. It would have been a lonely lab without them around.

I owe my deepest gratitude towards my family for their eternal support and understanding of my goals and aspirations. My hard-working parents have sacrificed their lives for my sister, my brother and myself. Without their help, I would not have been

able to complete much of what I have done and become who I am. Thus I would like to ensure them that their infallible love has always been my strength. A special and warm thanks to my brother and sister for their love and affection and to my lovely sweet angel Ramy for all the happiness he has been giving me. I am also very much grateful to all my family members for their constant inspiration and encouragement. There are no words to convey how much I love and respect all of them.

As always it is impossible to mention everybody who supported me during my studies however there are those whose I cannot imagine my life without. I feel a deep sense of gratitude for my friend Arwa, for her unconditional love and faith in me and my intellect even when I didn't have faith in myself. I feel privileged to be associated with a person like her therefore may ALLAH give her all the best in return.

Contents

List of Figures

List of Tables

List of Acronyms

ACL2	A Computational Logic for applicative common Lisp
CAS	Computer Algebra System
CDF	Cumulative Distribution Function
CSL	Continuous Time Stochastic Logic
CTMC	Continuous-Time Markov Chain
DNA	Deoxyribonucleic acid
DTMC	Discrete-Time Markov Chain
FCFS	First Come First Served
FIFO	First In First Out
HMM	Hidden Markov Model
HOL	Higher-Order Logic
IDD	Independent and Identically Distributed
KLAIM	A Kernel Language for Agents, Interaction and Mobility
LED	Light Emitting Diode
MAP	Maximum a Posteriori
MCMC	Markov Chain Monte Carlo
MDP	Markov Decision Process
ML	Meta Language
OTP	One-Time Pad
PCTL	Probabilistic Computer Temporal Logic
PDF	Probability Density Function
PMF	Probability Mass Function
PRISM	PRobabilistIc Symbolic Model checker
SMC	Statistical Model Checking
SMDP	Semi-Markov Decision Process
SML	Standard Meta-Language

Chapter 1

Introduction

1.1 Motivation

Most of the important engineering systems encountered in our everyday life have random nature, i.e., their actual or future behavior is unpredictable due to various environmental conditions. The analysis of such systems involves *probabilistic analysis*, where we use probability theory principles to mathematically model elements of randomness and uncertainty in order to measure the likeliness of occurrence of a particular event. More specifically, *probability theory* has grown to be one of the most important branch of mathematics which is used for the probabilistic analysis of random experiments, providing the basis to model the complex behavior of numerous engineering systems.

A *Markov process* is a random process which exhibits the *Memoryless property* (also known as the Markov property) [15], which states that the future behavior of the process only depends on its current state regardless of its past behavior. In the probability literature, *Markovian systems* are usually divided into four types which are essentially based on their time and state parameters [15]: *discrete-time and discrete state*, *continuous-time and continuous state*, *continuous-time and discrete state*, and *discrete-time and continuous state*. The continuous-time and discrete state Markov Process is usually called *Continuous-Time Markov Chain* (CTMC) [15] which

describes the collection of *random variables* that takes values in a countable set or countably infinite set where elements of this set represent possible states and the chain transits from one state to the other. Moreover, *the sojourn time* (total time spent in one state) is random (particularly, exponentially distributed). CTMC is considered as the basic concept of many mathematical theories, such as the embedded Markov chain theory, hidden Markov models and Queueing theory.

CTMC is widely used to model and analyze complex software and hardware systems in a variety of areas such as engineering, basic sciences, health care, finance, etc., as shown in Figure 1.1.

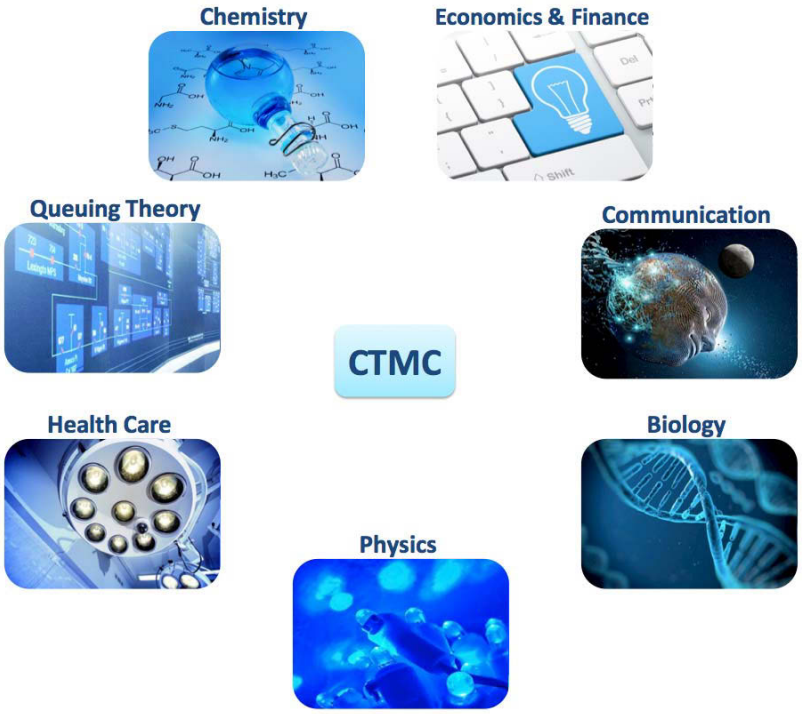

Figure 1.1: Markov Chain Application Fields

For instance, the CTMC theory can be applied in constructing the reliability models and analyzing system performance, e.g., software-based control systems and their dependability properties can be modeled as CTMC since they exhibit a stochastic behavior. A potential case study can illustrate the dependability analysis in an embedded control system where delays can occur due to failure of a component, a

transient fault or processor reboot. All these delays are distributed exponentially; hence the system can be modeled as a CTMC and its analysis can give a clear idea about the performance and reliability of the system [56].

CTMC can also be used to model the progression of some diseases such as breast cancer [20]. This study started when researchers noted that women above forty have less chances of fighting this disease when using mammographic screening compared to younger women. Consequently, a series of Markov-chain models have been developed to estimate the tumor progression rates and sensitivity. The main parameter to estimate in this study is the mean sojourn time (the average duration of the preclinical screen-detectable period) which is set to be less than two years to achieve a reduction in cancer mortality. Further, this study can be useful in the design and analysis of future studies of breast cancer screening [20].

Combined with related probabilistic models, CTMCs are nowadays the basis of many algorithms for the analysis of biological sequences, like comparative sequence analysis, in particular the annotation of simultaneous alignment and multiple alignments [77]. This combination has been mainly used to predict genes and conserved regions in DNA sequences, secondary structures and transmembrane topologies in protein sequences and base pairing structures in DNA sequences [77]. In addition to this, CTMC has been applied to model reaction networks, which are chemical systems involving multiple reactions and chemical elements. The system is treated as a CTMC where its states correspond to the number of molecules of each species and the reactions are modeled as possible transitions between the molecules [58]. Chemical reactions are mainly used for the experiments validation with an ultimate goal of new drugs discovery for the treatment of different diseases.

A variety of network's security problems are today described as Markov chains, e.g., the case of a virus infecting a network and multiplying through the connected nodes. Once a node is infected, the virus remains at that node and repeatedly tries to infect any of the neighboring nodes while they remain uninfected. This model is based on a probabilistic extension of KLAIM (Kernel Language for Agents Interaction

and Mobility) [81] and CTMC where the behavior of the network and the individual nodes is determined by a probabilistic allocation environments, which describe the logical neighborhood of each node [57].

Numerous queueing models are built on the CTMC concepts. For example, the *M/M/1 queue* is a CTMC over the non-negative integers where upward transitions from n to $n+1$ occur at rate λ according to a Poisson process which describe arrivals to the queue, while backward transitions from n to $n-1$ occur at rate μ which describe completed services. When analyzing queueing systems, it is very important

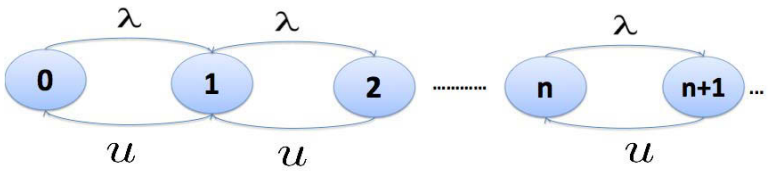

Figure 1.2: Flow Diagram of the M/M/1 Queue

to assess important characteristics, like the measure of the typical waiting time of a customer or the manner in which customers accumulate needs to be verified [24]. Moreover probability distributions or their expected values are also parameters of great interests [24].

In order to formally model and analyze queueing systems, the *birth-death process* [98] is deployed as a simple yet important form of CTMCs. In a Markov model of a queue, the state represents the number of costumers in the queuing system while transitions occur only between adjacent states. Using CTMC, a variety of queues with memoryless arrival processes and service time can be modeled. For example, the *M/M/1 queue* [98] is a very common model where arrivals follow a *Poisson process* and service times are *exponentially distributed*.

With the increasing usage of engineering systems in safety-critical domains, such as medicine, transportation and communication systems, accurate, precise and scalable analysis techniques have become a dire need of the present era. Traditionally,

paper-and-pencil proofs have been used for conducting probabilistic analysis. However, when it comes to complex computations and scalability, this method fails to maintain the correctness of large proofs due to the risk of human error. The second commonly used analysis method is computer based techniques, which can be divided into two main streams, i.e., the simulations based methods and the computer algebra systems (CAS). The main idea of simulation based methods is to construct a system model that can be simulated with unlimited variations producing different scenarios. In the second alternative, i.e., computer algebra systems, the mathematical computations are done using symbolic algorithms, and hence they are better than simulation based analysis. But analysis based on both computer simulations and computer algebra systems cannot provide 100% percent accurate results. In computer simulation, the analysis is based on different approximations which may lead to an erroneous analysis. On the other hand computer algebra systems, which are very efficient for mathematical computation, are not sound because the computed results are not always mathematically correct. For example, in Maple [65], if we give the following input:

$$\frac{x^2-1}{x-1} \tag{1}$$

the result will be $x+1$, which is an over simplification, as the case $x=1$ is ignored since it gives an indeterminate value $\frac{0}{0}$.

The accuracy is the main concern of system analysis because any minor error can lead to disastrous consequences, which may result in the loss of human lives. Some consequences of erroneous simulations based analysis include, the Ariane 5 crash in 1996 due to data conversion error that resulted in the loss of more than 370 million US$ [30] and the Air France flight 447 crash due to inaccurate air speed measurement by the sensors, which resulted in the loss of 228 human lives [25]. Due to above mentioned limitations, the traditional analysis techniques cannot be relied upon for the analysis of Markov chains systems.

Formal methods [37] allow accurate and precise analysis and provide means to overcome the limitations of traditional approaches. Formal methods tend to develop

a mathematical model for a given system, this model is analyzed using mathematical reasoning which help in catching critical design errors that are often ignored by traditional techniques like numerical methods. The two most commonly used formal methods techniques are *model checking* [10] and *theorem proving* [40]. Model checking is an automatic verification technique for systems that can be expressed as finite-state machines. On the other hand, theorem proving is an interactive verification technique, which is more powerful in terms of expressiveness (e.g., higher-order-logic) and mathematical analysis.

Given the sophistication of the present age Markov chain systems and their extensive usage in safety-critical applications, there is a dire need of using formal techniques in this domain. In fact, the applicability of formal methods for queueing system analysis is limited. This thesis presents some mathematical foundations that provide a novel platform for the formal analysis of queueing systems using higher-order-logic theorem proving. The ability to accurately conduct these analysis may prove to be a very useful feature for the systems used in safety-critical domains.

1.2 Related Work

There exists a significant amount of research going on in the area of Markov chains and Queueing systems. In this section, we present existing state-of-the-art techniques for these system analysis. Traditionally, the analysis of Markov chains based models has been done using paper-and-pencil proof methods. However, considering the complexity of present age engineering and scientific systems, such kind of analysis can hardly guarantee accurate analysis due to the risk of human errors. Thus, computer-based techniques have been proposed as an alternative to the traditional approaches.

1.2.1 Simulation

Simulation is one of the widely used computer based probabilistic analysis technique for Markov chain models. Markov chain Monte Carlo (MCMC) methods [64] have

emerged as the main simulation algorithm for sampling from a probability distribution, which are based on constructing a Markov chain that has an approximate distribution in terms of the residual effect of the initial position. MCMC methods sample successively from a target distribution and each sample depends only on the previous one, hence the notion of the Markov chain. Approximately, the constructed Markov chain has the desired distribution in terms of the residual effect of the initial position. Although some more sophisticated MCMC-based algorithms are capable of producing exact samples matching the given probability distribution, they often introduce additional computation overhead and unbounded running time [100].

Additionally, reliability evaluation tools and Markov analyzers uses numerical methods in order to model and analyze the reliability, maintainability or safety of systems based on Markovian models. These tools offer simplistic modeling approaches and are more flexible compared to traditional techniques. Some prevalent tool examples include Möbius [73] and SHARPE [89]. They mainly provide the services on analyzing the failure or repair of a model, which may occur in the lifetime of any product. Some other software tools used for evaluating performance, e.g., MACOM [87] and HYDRA [53], use a popular Markovian process algebra [12], i.e., PEPA [80], to model systems and efficiently compute passage time densities and quantities in large-scale Markov chains. Since most of the models are complex, they are analyzed using expanded iterative methods, which often lead to approximations because the computations stop at some convergence point. Hence, the results might become untrusted since numerical computations are certainly affected by roundoff and truncation errors.

Although simulation techniques are widespread and able to provide a practical feedback when it comes to analyzing real-world systems but like most of the other analysis methods, they have their drawbacks. Generally, the used algorithms are based on numerical methods which lead to inaccurate results. In addition, many rounding errors also creep into the analysis due to the involvement of computer arithmetics.

These limitations are considered as serious problems especially while analyzing highly sensitive and safety-critical applications, such as nuclear reactor controllers or aerospace computing systems.

1.2.2 Computer Algebra Systems

In order to overcome the inaccurate results generated by applying numerical methods, computer algebra systems (CAS) offer a fully automated and friendly human-machine interface which supports Markovian models analysis in symbolic form. Recently, the CAS based tool, Mathematica [66], has introduced a Markov chain analysis toolbox which provides a completely automatic analysis. Moreover, Mathematica has been long used to derive symbolic Maximum Likelihood estimator [85]. One of its important task is constructing the log-likelihood for a random sample consisting of Poisson Random Variable [86]. Another well-known CAS, Maple [65], also utilizes Markov chains for solving many problems like financial problems, by automatically constructing transition matrices in Markovian models.

Symbolic computation has its limitations and thus cannot always supersede numerical solutions. In fact the results often include approximations due to the utilization of some numerical methods, such as the Jacobi Over-Relaxation, Gauss-Seidel and Successive Over-Relaxation algorithms [11]. However, the simplifications performed in the CASs are not strictly logical as they are not able to deal with side conditions. For example, Mathematica returns 1 as the answer when given x/x as the input. It is clear that $x/x = 1$ holds only when $x \neq 0$ [60]. Another serious analysis issue is caused by the use complex symbolic manipulation algorithms, which have not been verified.

1.2.3 Probabilistic Model Checking

Probabilistic Model Checking is a formal verification technique that allows the analysis of systems exhibiting Markovian behavior. It supports various tools that combine

a range of techniques for calculating the likelihood of the occurrence of certain events during the execution of the system and can establish properties to be considered. However, some important probabilistic questions are hard to be answered directly for the reason that the logics used to express the properties are limited in expressiveness. Some of the most widely used probabilistic model checkers include PRISM [83], VESTA [1] and Ymer [99]. In this context, we can mention the work of Bortolussi et. al. [2] where they used the Statistical Model Checking (SMC) [52] approach based on Bayesian statistics and advanced machine learning (Gaussian processes, the GPU-CB algorithm [94]), in order to learn about the parameters of stochastic processes from observations of the state of the system. They specified an analytical expression for the log-likelihood for both CTMC and Poisson process [2].

Also, we can mention the ErlangenTwente Markov Chain Checker [9] which supports the model checking of CTMCs using temporal logic specifications called continuous time stochastic logic (CSL) [93]. This tool was proven to be inefficient when it comes to accuracy of numerical and verification results [46].

Several other formal CTMC analysis tools are available, for example MARCA [95] and TIPPtool [45]. MARCA is designed to facilitate the generation of large Markov chain models, to determine mathematical properties of the chain, to compute its stationary probability and to compute transient distributions. However, TIPPtool provides performance evaluation where Markov chain models are specified by means of a process algebra [45]. These tools do not allow logic specifications and instead support steady-state and transient analysis.

Although the above tools offer exhaustive solutions, they suffer from the state-explosion problem [8]. Moreover, the algorithms integrated in these tools for analysis are based on iterative methods, simulation and statistical means which leads to inaccurate results.

1.2.4 Theorem Proving

Theorem proving based probabilistic analysis tends to overcome the limitations of the above mentioned approaches. Over the past decade, many foundational mathematical theories have been formalized. Nedzusiak [74] and Bialas [16] were among the first to formalize some measure and probability theories in higher-order-logic. Then, Hurd [50] developed a probability theory and formalized the measure space as a pair (Σ, μ) in the HOL theorem prover [35]. However in this formalization the space is implicitly the universal set of appropriate data-type. The probability space was also defined in HOL as a pair $(\xi, \mathbb{P})$ where ξ is a σ-algebra closed under complements and countable unions, and the domain of $\mathbb{P}$ is the set ξ which is the set of subsets of finite Boolean Sequences $\mathbb{P}^{\infty}$. Hasan [42] built upon Hurd's work and formalized statistical properties of both discrete and continuous random variables and their *Cumulative Distribution Function* (CDF) in the HOL4 theorem prover [91]. However Hasan's work inherits the same limitations as of Hurd's work. As a consequence, when the space is not the universal set, the definition of the arbitrary space becomes very complex. Later, Coble [23] defined probability space and random variables based on an enhanced formalization of measure space which is the triple (X, Σ, μ). This measure space overcomes the disadvantage of Hurd's work since it contains an arbitrary space. Coble's probability theory is built upon finite-valued (standard real numbers) measures and function. Specifically, the Borel Sigma spaces cannot be defined on open intervals which constrains the verification of some applications. More recently, Mhamdi [67] used the axiomatic definition of probability proposed by Kolmogorov [54] to provide a significant formalization of both measure and probability theory for formally analyzing information theory in HOL4. His work overcomes the limitations of the above mentioned works by allowing the definition of Sigma-finite and other infinite measures as well as the signed measures. Hölzl [47] has also formalized three chapters of measure theory in Isabelle/HOL [75]. Affeldt [4] simplified the formalization of probability theory in Coq [27].

Based on Mhamdi's formalization, Liu [63] provided an alternative approach to

verify Markovian models, which is capable of offering accurate, scalable and generic results. To meet this objective, she constructed a foundational framework for conducting Markov chain based analysis in HOL4. Mainly, she provided the formalization of Discrete-Time Markov Chain (DTMC) and the verification of the most important properties, in which the concepts of reversibility and stationary properties accommodate the formal reasoning about Markov chain mixing time [59] and the formalizations of stationary process. In addition to this, Liu developed the formal definitions of classified states and classified DTMCs, as well as the verified properties of the aperiodic and irreducible DTMCs [55]. She also investigated the formalization of discrete-time HMMs and the verification of their associated properties, such as joint probability and the probability of observation path [60]. Hölzl [48] formally defined a time-homogeneous Markov chain based on the finite state space and the transition matrix in Isabelle/HOL. The mainly goal of this work was to verify Probabilistic Computation Tree Logic (PCTL) in probabilistic model checkers, hence, a generalized formalization of DTMC theory has not been provided. Furthermore, this work does not support time-inhomogeneous Markov chains.

From the above discussions of related work, computer-aided techniques such as simulation, CAS and model checking, clearly provide a number of advantages over traditional paper-and-pencil based proofs. However, their usage is limited due to their inherent nature. For instance, due to the inaccurate nature of the underlying algorithms, which are based on numerical methods, they may generate inaccurate results. The theorem proving approach, on the other hand, tends to overcome these limitations as the analysis carried out will be free from any approximation and precision issues. Similarly, the high expressibility of higher-order logic allows us to analyze a wider range of systems without any modeling limitations. Particularly, the HOL4 theorem prover provides rich libraries and theories for the formal probabilistic analysis of a variety of systems.

In this thesis, we are providing a framework that can be used to formalize CTMCs in the HOL4 theorem prover. Our work uses and extends the work done by Liu

[60]. The main reasons behind this choice include the availability of basic building blocks of probability theory and real analysis related theories in HOL4. Moreover, the availability of CTMCs in HOL4 theorem prover facilitates the formal reasoning about queueing systems.

1.3 Proposed Framework

The objective of this thesis is mainly targeted towards the development of a Poisson process and CTMC based system analysis framework using higher-order-logic theorem proving, which can handle the analysis of real-world systems. In particular, we want to develop a framework characterizing:

1. The ability to formally express transition probabilities in higher-order logic.
2. The ability to formally verify the properties of the Poisson process, Poisson distribution and M/M/1 queue in higher-order logic theorem prover.
3. The ability to utilize the above mentioned capabilities to formally model and reason about real-world queueing and Markovian systems.

A general overview of the proposed framework is illustrated in Figure 1.3. The framework outlines the main idea behind the theorem proving based Markovian and queueing system analysis. Like any system analysis tool, the input to this framework is the description of the system that needs to be analyzed and a set of properties that are required to be checked for the given system.

To conduct the queueing system analysis, the first step is to construct the formal model as a function in higher-order logic based on the given system description. For this purpose, the foremost requirement is the ability to provide the formal mathematical definitions of continuous-time Markov chain, Poisson process, Poisson distribution, Birth-Death process and the $M/M/1$ queue. We used Liu's work on the formalization of conditional probability [60] to fulfill the requirements. The second step is to utilize the formal model of Markovian and queueing systems, developed in the first step,

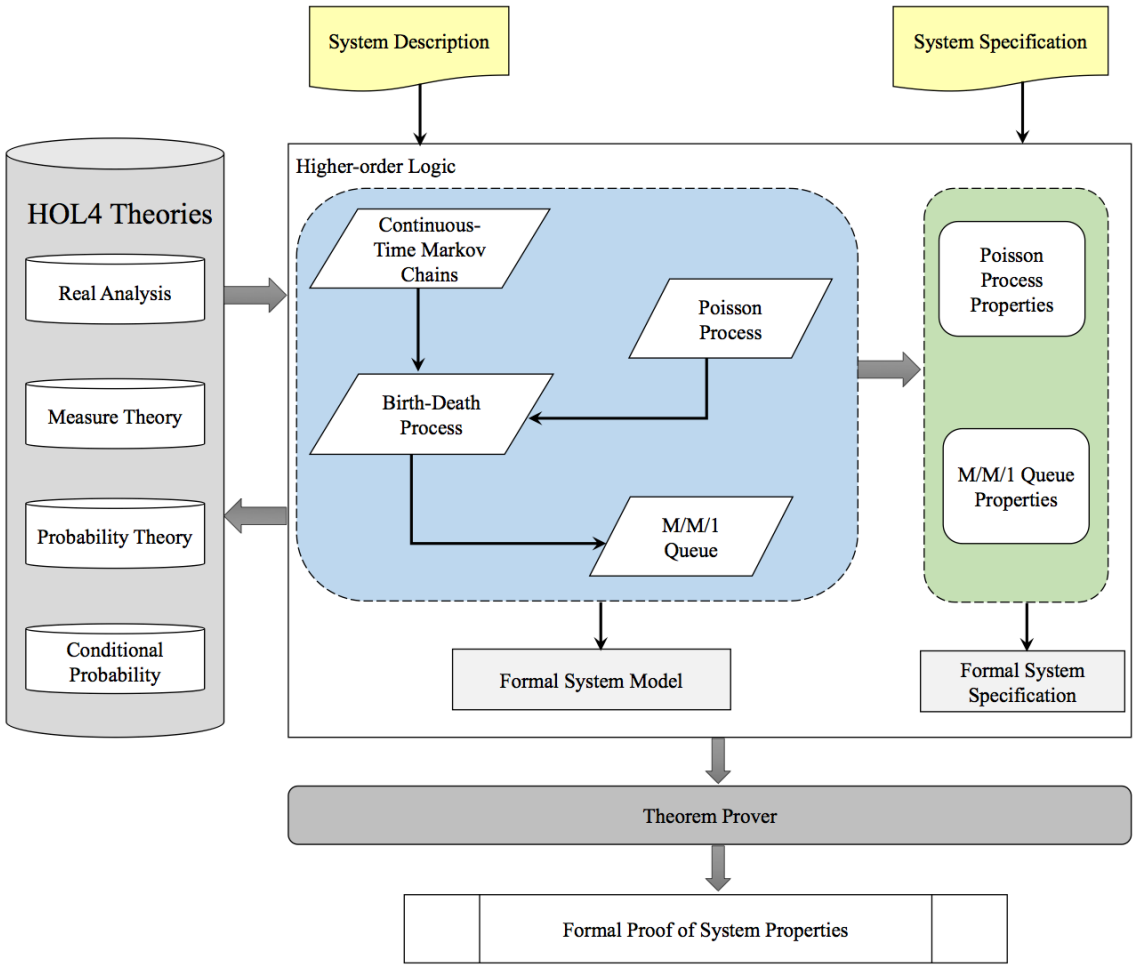

Figure 1.3: Overview of the Proposed Framework

to express system properties as higher-order logic theorems. In order to conduct the formal analysis of the system properties, we used a library containing some already verified theorems, such as the probability independency theorem, binomial expansion and coefficient, L'Hopital rule, etc.

The third and last step is the formal verification of system properties in higher-order logic. For this verification, it would be quite handy to establish this library for the purpose of facilitating the formal reasoning about Markovian and queueing systems. To fulfil this requirement, this thesis presents the formal verification of the classical properties of Poisson process, such as, independent and stationary increments property, the exponential inter-arrival time and the memoryless property,

using the HOL4 theorem prover. Building on such a library of theorems minimizes the interactive verification efforts and thus speeds up the verification process. Finally, the output of the theorem prover in this framework are the formal proofs of the system properties certifying that the given system properties are valid for the given Markovian and queueing system.

1.4 Thesis Organization

The rest of the thesis is organized as follows: In Chapter 2, we provide a brief introduction to the HOL theorem prover and an overview of the formalization of the probability theory and the conditional probability to equip the reader with some notation and concepts that are going to be used in the rest of this thesis. Chapter 3 describes the formalization of the Poisson process along with the formal verification of its corresponding properties in HOL, e.g., exponential interarrival times and the memoryless property. To illustrate the utilization of these mathematical formalizations, we use them for a formal probabilistic analysis of an optical communication system. Chapter 4 presents a case study for the formalization of the M/M/1 queue based on the time-homogenous CTMC and the birth-death process. Based on their definitions, the major interesting properties of the M/M/1 queue are formally verified as theorems. Then, a single runway model and a network of queues are formally validated as an application. Finally, Chapter 5 concludes the thesis and outlines some future research directions.

Chapter 2

Preliminaries

In this chapter, we provide a brief introduction to the HOL theorem prover and present an overview of Mhamdi's [68] and Liu's [61] formalization of Probability Theory and Conditional probability, respectively. The intent is to introduce the basic theories along with some notations that are going to be used in the rest of the thesis.

2.1 Theorem Proving

Theorem proving is one of the most developed research area in automated reasoning. It is concerned with the construction of mathematical theorems using a computer program. These mathematical theorems can be built on different types of logic, such as, propositional logic [18], first-order logic [33] or higher-order logic [17], depending upon the expressibility requirement. For example, the use of higher-order logic is advantageous over first-order logic in terms of the availability of additional quantifiers and highly expressive nature of higher-order logic. The main idea behind the theorem proving based formal analysis is to mathematically model the given system in an appropriate logic, later the properties of interest are verified using computer based formal reasoning. Using higher-order logic theorem proving for modeling the system behaviors makes the analysis very flexible as it allows the formal verification of any system that can be expressed mathematically. The core of theorem provers

usually consists of some well-known axioms and primitive inference rules. The theorem proving based verification assures the soundness since every new theorem must be created from these basic axioms and primitive inference rules or any other already proved theorems.

There are two types of provers, i.e., automatic and interactive. In an interactive theorem prover, significant user-computer interaction is required while automatic theorem provers can perform different proof tasks automatically. The degree of automation is dependent on the used logic, for example first-order logic can be significantly automated whereas it is difficult to automate high-order logic theorem proving due to its undecidable nature. Some commonly used automated provers include SATURATE, LeanTAP, Gandalf, METEOR, SETHEO, Otter and MetiTarski [5]. The family of interactive higher-order logic based theorem provers includes Isabelle, Coq, HOL, HOL4, HOL Light and ProofPower [41].

This thesis uses the HOL4 theorem prover [91] to conduct all the analysis. The main reasons behind this choice include the richness of Mhamdi's probability and measure theories [67], which are fundamental to our work, and the ability to use a part of Liu's formalization [60] to formalize the Poisson process, CTMC and the M/M/1 queue and to formally verify their properties. Moreover, some earlier work related to the formal analysis of Markov chains, such as, Elleuch's formal probabilistic analysis of wireless sensor networks [31] and Liu's formal analysis of discrete time Markov chains [62], inspired this thesis to be done in the HOL4 theorem prover.

2.2 HOL Theorem Prover

HOL is an interactive theorem proving environment for the construction of mathematical proofs in higher-order logic. The first version of HOL was developed by Mike Gordon at Cambridge University, in the 1980′s. The core of HOL is interfaced to the functional programming language ML-Meta Language [79]. HOL utilizes the simple

type theory of Church [21] along with Hindley-Milner polymorphism [71] to implement higher-order logic. The first version of HOL is called HOL88 and other versions of HOL are HOL90, HOL98 and HOL4. HOL4, the most recent version of HOL family, uses Moscow ML which is an implementation of Standard ML (SML) [72]. The HOL core consists of only 5 basic axioms and 8 primitive inference rules, which are implemented as ML functions. HOL has been widely used for the formal verification of software and hardware systems along with the formalization of mathematical theories.

2.2.1 Terms

HOL has four types of terms: constants, variables, function applications, and lambda-terms. Variables are sequences of digits or letters beginning with a letter, e.g., y, b. The syntax of the constants is similar to the variables, but they cannot be bounded by quantifiers. The type of an identifier, i.e., variable or a constant, is determined by a theory; e.g., F, T. Applications in HOL represent the evaluation of a function g at an argument y, different terms can be used instead of g and y, e.g., f and x. In HOL, we can use λ-terms, also called lambda abstractions for denoting functions. A λ-term has the form $\lambda x.f(x)$ and represents a function which takes x and returns $f(x)$.

2.2.2 Types

According to the lambda calculus implemented in HOL, every HOL term has a unique type which is either one of the basic types or the result of applying a type constructor to other types. In HOL, each variable and constant must be assigned a type and variables with the same name but different types are considered as different. When a term is entered into HOL, the type is inferred using the type checking algorithm implemented in HOL, e.g., when $(\sim y)$ is entered into HOL, the HOL type checker deduces that the variable y must have type bool because negation $(\sim)$ has a type $bool \rightarrow bool$. If the type of a term cannot be deduced automatically then it is possible

to explicitly mention the type of that term, e.g., $(x : real)$ or $(x : bool)$.

2.2.3 Inference Rules

Inference rules are procedures for deriving new theorems and they are represented as ML functions. There are eight primitive inference rules in HOL and all other rules are derived from these inference rules and axioms. The rules are *Assumption introduction*, *Reflexivity*, *Beta-conversion*, *Substitution*, *Abstraction*, *Type instantiation*, *Discharging an assumption* and *Modus Ponens* [29].

2.2.4 Theorems

A theorem is a formalized statement that may be an axiom or follows from theorems by an inference rule. A theorem consists of a finite set of Boolean terms Ω called the assumptions and a Boolean term S called the conclusion. For example, if (Ω, S) is a theorem in HOL then it is written as $\Omega \vdash S$.

2.2.5 Theories

A HOL theory consists of a set of types, type operators, constants, definitions, axioms and theorems. It contains a list of theorems that have already been proved from the axioms and definitions. The user can load HOL theories to utilize the available definitions and theorems. These theories allow the utilization and extension of existing results without duplicating the efforts made in building them. HOL theories are organized in a hierarchical fashion and theories can have other theories as parents and all of the types, constants, definitions, axioms and theorems of a parent theory can be used in the child theory. For example, one of the basic theory in HOL is `bool` which is also parent theory of individuals `ind`. We utilized the HOL theories of Booleans, positive integers, real numbers, sequences, limits and transcendental functions in our work. In fact, one of the primary motivations of selecting the HOL theorem prover for our work was to benefit from these built-in mathematical theories.

2.2.6 Proofs in HOL

There are two types of interactive proof methods when using HOL: forward and backward. In a forward proof, the user starts from the primitive inference rules and tries to prove the goals using these rules and already proved theorems. The forward proof method is not an easy approach as it requires all the low level details of the proof in advance. A backward or a goal directed proof method is the reverse of the forward proof method. It is based on the concept of a *tactics*; which is an ML function that breaks goals into simple subgoals. There are many automatic proof procedures and proof assistants [38] available in HOL which helps the user in directing the proof to the end. In interactive theorem proving, the user interacts with HOL proof editor and guides the prover using the necessary tactics until the last step of the proof. Some of the proof steps are solved automatically while others require signification user interaction.

2.2.7 HOL Notations

Table 1 provides the mathematical interpretations of some frequent HOL symbols and functions used in this thesis. The use of HOL4 has emerged over the past few decades, for instance, the early formalization of main concepts in higher-order logic, such as real numbers, topology, limits, sequences and series as well as differentiation and integration, were done by Harrison [39]. Mhamdi [67] presented the higher-order logic formalization of Probability theory in the HOL theorem prover, which is a fundamental concept in many mathematical theories.

Table 1: HOL Symbols and Functions

HOL Symbol	**Meaning**
/\	Logical *and*
\/	Logical *or*
~	Logical *negation*
==>	Logical implication
<==>	Logical equality
!x.f	for all x : f
?x.f	for some x : f
(&n : num)	type casting (&n:extended real)
x pow n	x^n
λx.f	Function that maps x to $f(x)$
Univ	Universal Set
a IN S	a in S
A INTER B	A intersection B
disjoint A B	Sets A and B are disjoint
IMAGE f A	Set with elements $f(x)$ for all $x \in A$
PREIMAGE f B	Set with elements $x \in X$ for all $f(x) \in B$ and $f : X \to Y$
∅	Empty Set
FINITE S	S is a finite set
suc n	Successor of natural number
ln x	Natural logarithm function
exp x	Exponential function
BIGUNION P	Union of all sets in the set P
BIGINTER P	Intersection of all sets in the set P
λn.f(n) --→ p	$\lim_{n\to\infty} f(n) = p$
suminf(λn.f(n))	$\lim_{k\to\infty} \sum_{n=0}^{k} f(n)$
SIGMA(λn.f(n))S	$\sum_{n\in S} f(n)$
summable(λn.f(n))	$\exists x. \lim_{k\to\infty} \sum_{n=0}^{k} f(n) = x$

2.3 Probability Theory

The purpose of probability theory is to model random phenomena and experiments so that we can describe and predict relative frequencies (averages) of these experiments in terms of probabilities of events. The fundamental mathematical object is a triple (Ω, Σ, μ) called the *measure space*, where Ω is a set, called the *sample space*, Σ represents a σ-algebra of subsets of Ω, where the subsets are usually referred to as *measurable sets*, and μ is a measure with domain Σ. Mhamdi [69] defined a *probability space* as a measure space (Ω, Σ, Pr) where the measure of the sample space, denoted by Pr and referred to as the probability, is equal to 1. A probability space is needed for each experiment or collection of experiments that we wish to describe mathematically. Therefore, using measure theory to formalize probability has the advantage of providing a mathematically rigorous treatment of probabilities and a unified framework for discrete and continuous probability measures. A probability theory is developed based on three axioms [67]:

1. $\forall A.\ 0 \leq Pr(A)$
2. $Pr(\Omega) = 1$
3. For any countable collection $A0$, $A1$,... of mutually exclusive events, $Pr(\bigcap_{i\in\Omega} A_i) \ = \ \sum_{i\in\Omega} Pr(A_i)$

The above approach has been successfully used to formally verify most basic probability theorems, such as [67]:

$$0 \leq Pr(A) \leq 1$$
$$\sum_{A_i\in\Omega} Pr(A_i) = 1$$

Two events are independent if the occurrence of one does not change the probability of the other occurring. Thus, if events are independent, then the probability of them both occurring is the product of the probabilities of each occurring.

Definition 2.1. *Two events A and B are independent iff $p(A \cap B) = p(A)p(B)$.*

Here $A \cap B$ is the intersection of A and B, that is, it is the event that both events A and B occur.

```
⊢ independent p a b ⇔
      a ∈ events p ∧ b ∈ events p ∧
      prob p (a ∩ b) = prob p a * prob p b
```

A *random variable* is considered to be one of the core concepts in probabilistic analysis. It can be defined as a measurable function from a probability space (Ω, Σ, Pr) into a *measurable space* (S, Σ) also known as the state space, where S denotes a set and Σ represents a nonempty collection of subsets of S.

Definition 2.2. *$X : \Omega \to \bar{\mathbb{R}}$ is a random variable iff X is $(F, \mathcal{B}(\bar{\mathbb{R}}))$ measurable*

where F denotes the set of events. Here we focus on real-valued random variables but the definition can be adapted for random variables having values on any topological space thanks to the general definition of the Borel sigma algebra.

```
⊢ random_variable X p s ⇔
      prob_space p ∧ X ∈ measurable (p_space p,events p) s
```

Definition 2.3. *Two random variables X and Y are independent iff $\forall A, B \in \mathcal{B}(\bar{\mathbb{R}})$, the events $\{X \in A\}$ and $\{Y \in B\}$ are independent.*

The set $\{X \in A\}$ denotes the set of outcomes ω for which $X(\omega) \in A$. In other words $\{X \in A\} = X^{-1}(A)$.

```
⊢ independent_rv p X Y s t ⇔
  ∀ A B. A ∈ subsets s ∧ B ∈ subsets t ⇒
  independent p (PREIMAGE X A ∩ p_space p) (PREIMAGE Y B ∩ p_space p)
```

Once random variables are formalized, as mentioned above, we can utilize the formalized probability theory infrastructure to reason about their probabilistic properties. For example, the probability that a random variable X is exactly equal to some value

i is defined as the *Probability Mass Function* (PMF) and it is mathematically expressed as $Pr(X = i)$. The event $\{X \in A\}$ is used to define the PMF of a random variable.

Definition 2.4. *The Probability Mass Function p_x of a random variable X is defined as the function assigning to A the probability of the event $\{X \in A\}$.*

$$\forall A \in \mathcal{B}(\bar{\mathbb{R}}),\ p_X(A) = p(\{X \in A\}) = p(X^{-1}(A))$$

```
⊢ distribution p X = (λA. prob p (PREIMAGE X A ∩ p_space p))
```

Also utilizing the same infrastructure, we can denote a *random process* as a collection of random variables X_t ($t \in T$). If the indices (t) of random variables X_t are discrete, then this random process is a *discrete-time random process* otherwise it is known as a *continuous-time random process*.

```
⊢ (∀ t. random variable (X t) p s)
```

Mhamdi [67] generalized the formalizations of the probability and information theory by introducing the notion of extended real numbers, the Borel sigma algebra which covers larger classes of functions in terms of integrability and convergence. Hölzl [47] has also formalized a generic version of the measure, probability and information theory in Isabelle/HOL. Affeldt [4] simplified the formalization of probability theory in Coq. Among these works, the probability theory formalized by Mhamdi provides the most generic formal reasoning support and thus can be used to analyze wider range of applications.

2.4 Conditional Probability

One of the crucial concepts in the random process study is the conditional probability, which is used to calculate the occurrence probability of an event when another event is known to occur. Conditional probability basically reflects the dependency between the events which happen at different times in a process. The formal definition of

conditional probability in HOL can be found in [44], which is based on Hurd's work [50].

In order to make use of the most advanced probability theory of [67], Liu [60] defined an improved version of the formalization of conditional probability by:

Definition 2.5. *The conditional probability of the event A given the occurrence of the event B is*

$$Pr(A|B) = Pr(A \cap B)/Pr(B)$$

```
⊢ ∀ A B. cond_prob p A B = prob p (A ∩ B) / prob p B
```

where `cond_prob` represents the conditional probability, and `prob` denotes the probability. Liu [60] has verified various classical properties of conditional probability in order to facilitate the formalization of Markov chains . Some of the prominent ones are listed below:

$$Pr(A \cap B) = Pr(A|B)Pr(B)$$

$$Pr(A) = \sum_{i \in \Omega} Pr(B_i)Pr(A \cap B_i)$$

$$Pr(A) = \sum_{i \in \Omega} Pr(A)Pr(B_i \cap A)$$

$$\sum_{i \in \Omega} Pr(B_i \cap A) = 1$$

where A, B and C are events in the event space, and the finite events set $(B_i)_\Omega$ contains mutually exclusive and exhaustive events. The first theorem is obviously based on the conditional probability definition. The second one is the Total Probability Theorem [101] and the third one is a lemma of the Total Probability Theorem [78]. The last theorem is the Additivity Theorem [101].

Mathematically, the *conditional independence* [51] is an important concept, which is the foundation of graphical models and mainly used in Bayesian Network. The mathematical definition of conditional independence is:

Definition 2.6. *The events A and B are conditionally independent given the event C if*

$$Pr(A|B \cap C) = Pr(A|C)$$

2.5 Summary

In this chapter, we started with a brief introduction of theorem proving and discussed different state-of-the-art theorem provers. Then we provided an overview of the HOL theorem prover that we have used for our formalization related to fMarkovian systems. We later summarized Mhamdi's work on the formalization of probability theory. We also presented some formalization details related to conditional probability developed by Liu. The next chapter presents the formalization of the Poisson process and some of its important properties.

Chapter 3

Formalization of the Poisson Process

In this chapter, we describe the formalization of the Poisson process and the formal verification of some of its most important properties using the probability theory in HOL4. In order to illustrate the usefulness of this work, an optical communication system is formally analyzed in HOL.

3.1 Higher-Order-Logic Formalization

Given a probability space, a *stochastic process* $X_t : \Omega \to S$ represents a sequence of random variables X, where t represents the time that can be discrete (represented by non-negative integers) or continuous (represented by real numbers) [15]. The set of values taken by each X_t, commonly called states, is referred to as the state space. The sample space Ω of the process consists of all the possible state sequences based on a given state space S. Now, based on these definitions, a Poisson process can be defined as a stochastic process $\{X_t; t > 0\}$ with a specific *transition probability function*. A Poisson process denotes the number of events that occur after time 0 up through and including time $t > 0$. For example, these events might be the number of insurance claims filed by a particular driver, or the number of callers phoning in to a

help line, or the number of people retiring from a particular employer [28].

Therefore, in the time interval $(t, t+h)$, where $h > 0$, there may or may not be some events that take place. If h is small, then the likelihood of an event is roughly proportional to h, i.e., it is not very likely that two or more events will occur in a small interval. Thus, the increment is simply the random number of events occurring strictly after time t and up through and including time $t+h$. More formally we make the following definition of the *Poisson Process Transition Probability*:

$$\mathbb{P}(X(t+h) = n+m | X(t) = n) = \begin{cases} \lambda h + o(h), & \text{if } m = 1; \\ 1 - \lambda h + o(h), & \text{if } m = 0; \\ o(h), & \text{if } m > 1. \end{cases}$$

We speak of n and m as the number of events of the process by a certain time t. λ is the rate at which events are, on average, occurring. The rate (or intensity) function λ gives the rate as $\lambda(t)$ at time t. Note that the rate can vary with time, at this case it is possible to integrate the rate function over the interval. In our case we are dealing with a *homogenous Poisson process* where the rate is constant. Little-o notation [76] means that the function $f(h)$ approaches 0 faster than h itself approaches 0, $\lim_{h \to \infty} f(h)/h = 0$. Now, the *Poisson process transition probability function* can be formalized as follows:

Definition 3.1. *(Poisson Process Transition Probability Function)*

```
⊢ ∀ X p t h n m λ. Poisson_Trans_Fun X p t h n m λ ⇔ ∃ o1 o2 o3.
  if n ∈ space s ∧ m ∈ space s then
  cond_prob p (PREIMAGE (λ t. X (t + h)) {&n + &m} INTER p_space p)
  (PREIMAGE (λ t. X t) &n INTER p_space p) =
  if (m = 0) then 1- λ * h + o1 h
  else if (m = 1) then λ * o2 h
  else o3
  else 0
```

This definition states that the probability of an event occurring after time t and up through and including time $t + h$ can be expressed in terms of the probability of events which occurred up to time t. It is easy to understand that the probability of an event is zero, when this event is not in the event space. For instance, n is not in the state space implies that the event $\{t \mid X_t = n\} = \emptyset$. In this case, the conditional probability related to an empty set is zero. Therefore a *Poisson process* can be formalized as follows:

Definition 3.2. *(Poisson Process)*

$\vdash \forall$ `X p t h n m` λ`. is_poisson_process X p t h n m` $\lambda \Leftrightarrow$
`real_random_variable X p` $\wedge$ `Poisson_Trans_Fun X p t h n m` λ

The first conjunct indicates that the Poisson process is based on a random process $X_t : \Omega \to S$. The quantified variable X represents a function of the random variables associated with time t which has the type real. This ensures the process is a *continous time* random process. The random variables in this process are the functions built on the probability space p and a measurable space s.

Each realization of the process is a non-negative, non-decreasing and integer-valued step function. It is described with the *Poisson distribution*, which is a discrete probability distribution for a countably infinite sample space that expresses the probability of a number of events occurring during a fixed period of time, where these events occur with a known average rate and independently of the time since the last event. Thus, a discrete random variable X is said to have a Poisson distribution with parameter $\lambda > 0$, if the probability mass function of X is given by:

$$\mathbb{P}(X(t) = n) = \frac{\lambda t^n e^{-\lambda t}}{n!}$$

In HOL, our definition will be as follows

Definition 3.3. *(Poisson distribution)*

```
⊢ ∀ X p t n λ. Poisson_distr_rv X p t n λ ⇔
 real_random_variable X p ∧ &n ∈ IMAGE X (p_space p) ∧
 distribution p X {&n} = (λ * t) pow n * exp (-λ * t) / &FACT n
 IMAGE f s = f x | x IN s
```

Here the first two variables are inherited from the random variable definition, while t and n refer to the time and the number of events, respectively. `IMAGE` fs returns the image of a given set s by a function f, where f is the random variable X and s is the state space of the Ω of the probability space p. Thus, the second condition ensures that n is in the image sample space of the random variable function in the considered probability space.

It is important to note that t in our case has the type *real* which describes the continuous feature of the process. Moreover, the rate λ is once again a constant with a *real* type.

3.2 Formal Verification of the Poisson Process Properties

Using the formal definition of the Poisson process and its distribution, we proved some of the most important properties of the Poisson process, which are frequently used in the analysis of many systems modeled as CTMCs.

3.2.1 Independent and Stationary Increments Property

Given a Poisson probability distribution, this property states that in any small interval the probability of occurrence of one event is linearly proportional to the rate and interval length and the probability of occurrence of more than one event in a small interval is negligible.

Theorem 3.1.

```
⊢ ∀ X p t h n m λ. Poisson_Process X p t h n m λ ⇔
 (∃ i. {&n + &i} ∈ subsets s) ∧
  (∃ n m. indep_rv p (λt. X (t + h)) (λt. X t) s s {&m} {&n}) ∧
   (λ > 0) ∧ (t > 0) ∧ (h > 0) ∧
     (∃ k n. Poisson_distr_rv (λt. X (t + k)) p (t + k) λ (n)) ⇒
is_poisson_proces X p t h n m λ
```

The proof of Theorem 3.1 is based on probability theoretic reasoning along with transcendental functions properties. Following, we describe the concept of the binomial expansion and the Taylor series expansion for the exponential function, both are used in order to find the expression of the function $o(h)$.

The Binomial Expansion

The *binomial expansion* [7] is one of the most well-known mathematical objects, it is the algebraic expansion of powers of a binomial. Thus, the binomial theorem allows to find the expansion binomials rased to varying degrees based on the theorem

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k$$

where $\binom{n}{k}$ is called the *binomial coefficient*. Combinatorially, the binomial coefficient counts the number of subsets of size k from a size n set. Elleuch [31] formalized the binomial coefficient based on the *Pascal Relationship* and a lot of mathematical reasoning related to the real summation. Based on Elleuch's formalization and various summation properties that we proved, we were able to decompose the binomial expansion as follow:

$$(a+b)^n = a^n + \sum_{k=1}^{n} \binom{n}{k} a^{n-k} b^k$$

$$(a+b)^n = a^n + \binom{n}{1} a^{n-1} b + \sum_{k=1}^{n} \binom{n}{k} x^{n-k} y^k$$

Lemma 1.

```
⊢ ∀ a b n. (a + b) pow n =
a pow n + sum (1,n) (λn. &binomial n x * a pow (n - x) * b pow x)
```

Lemma 2.

```
⊢ ∀ a b n. (a + b) pow n = a pow n + &binomial n 1 * a pow (n - 1) * b
 sum (2,n-1) (λn. &binomial n x * a pow (n - x) * b pow x)
```

The Taylor Series Expansion for the Exponential Function

A Taylor series [82] is commonly used in engineering analysis to approximate functions that do not have closed form solution. It is an expansion that can be helpful in approximating many commonly used functions such as exponential functions. The exponential function is defined as follows:

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

Based on this definition, we were able to prove that

$$e^x = 1 + \sum_{k=1}^{\infty} \frac{x^k}{k!}$$

$$e^x = 1 + x + \sum_{k=2}^{\infty} \frac{x^k}{k!}$$

Lemma 3.

```
⊢ ∀ x. exp x = 1 + lim (λn. sum (1,n) (λn. inv (&FACT n) * x pow n))
```

Lemma 4.

```
⊢ ∀ x. exp x = 1 + x + lim (λn. sum (2,n) (λn. inv (&FACT n) * x pow n))
```

This was verified based on proving some properties of the infinite summation using Siddique's [90] formalization regarding the limitation properties of a sequence.

Lemma 5.

```
⊢ ∀ f n m. (λn. f n) --→ p ⇔ (λn. f (n + m)) --→ p
```

Lemma 6.

```
⊢ ∀ f. lim (λn. sum (0,n) f) = lim (λn. sum (0,n+1) f)
```

Lemma 7.

```
⊢ ∀ f. lim (λn. sum (0,n) f) = lim (λn. sum (0,n+2) f)
```

L'Hopital's Rule

In order to verify that $\lim_{h\to\infty} f(h)/h = 0$, we used the L'Hopitale [22] Rule, which was already verified in HOL4 [43].

3.2.2 The Exponential Interarrival Times

The interarrival time refers to the time between successive events. Taking into consideration that the Poisson process is itself a form of CTMC, the interarrival time is actually the sojourn time in one state. These interarrival times are typically exponentially distributed with mean $\frac{1}{\lambda}$.

Let T_k be the time of the kth event in a Poisson process. The number of events occurring before some fixed time t is less than k if and only if the waiting time until the kth event is more than t. Formally, this means that the probability of the event $(X(t) < k)$ occurring is equal to the probability of the event $(T_k > t)$ taking place:

$$\mathbb{P}(T_k > t) = \mathbb{P}(X(t) < k)$$

As a consequence, if we consider T_1 the time of the first arrival, then clearly the waiting until T_1 is greater than t if and only if the number of events occurring before time t is 0. Applying this property to the probability distribution of homogeneous Poisson process gives us the following expression:

$$\mathbb{P}(T_1 > t) = \mathbb{P}(X(t) = 0) = e^{-\lambda t}$$

We managed to verify this property, theorem 3.3, in HOL based on the definition of the increasing sequence [60] and the counting process definitions.

Definition 3.4. *(Counting Process)*

```
⊢ ∀ X. counting_process X ⇔ (X 0 = 0) ∧ increasing_seq X
```

Theorem 3.2.

```
⊢ ∀ X p t T₁. (X(T₁) = 1) ∧
counting_process X ∧ (t < T₁) ⇒
     (prob p (PREIMAGE X {&X t} ∩ p_space p) = distribution p X {0})
```

In this theorem, T_1 refers to the time of the first arrival. Thus, the probability of an event occurring (a customer's arrival) at any time t less than T_1 is equal to the probability of having 0 customers in the system.

Theorem 3.3.

```
⊢ ∀ X p t T₁ λ. (X(T₁) = 1) ∧
counting_process X ∧ (t < T₁) ∧ Poisson_distr_rv X p t λ {n}
⇒ (prob p (PREIMAGE X {&X t} ∩ p_space p) = exp(-λt) )
```

The probability distribution of a random variable X can be uniquely described by its cumulative distribution function (CDF), which is defined as

$$(F_X(x)) = \mathbb{P}(X_x)$$

We then utilize this definition with Theorem 3.3 along with the additivity property of probabilities to prove that the probability of the event $(T_1 \leq t)$ is equal to the CDF of the exponential distribution.

$$\mathbb{P}(T_1 \leq t) = 1 - \mathbb{P}(T_1 > t) = 1 - e^{-\lambda t}$$

3.2.3 The Memoryless Property

The memoryless property of the exponential distribution gives the Poisson process its uniqueness among random processes. The memoryless property is meant to describe the conditional behavior of exponential random variables, which is one of the key results to derive the solution of queueing systems. If we consider that we have already waited for a time t and no decay has been observed, then the event $(T_1 > t)$ has occurred. Our main concern is the probability that the event $(T_1 > t + s)$ will occur. In fact, this type of problem shows up frequently in queueing systems where the time between events provides very useful information about arrival or service times.
To address this situation, we use the definition of conditional probability as follows:

$$\mathbb{P}(T_1 > t + s | T_1 > t) = \frac{\mathbb{P}(T_1 > t + s \cap T_1 > t)}{\mathbb{P}(T_1 > t)}$$

If we already know that $\mathbb{P}(T_1 > t + s)$, then $(T_1 > t)$ is redundant, therefore we can simplify the numerator.

$$\mathbb{P}(T_1 > t + s | T_1 > t) = \frac{\mathbb{P}(T_1 > t + s)}{\mathbb{P}(T_1 > t)}$$

Based on the fact that $(T_1 > t)$ is redundant in our case, we verified the above theorem in HOL using probabilistic and set theoretic reasoning

Theorem 3.4.

```
⊢ ∀A B X p. (PREIMAGE X A ∩ p_space p) SUBSET
(PREIMAGE X B ∩ p_space p) ⇒
(prob p ((PREIMAGE X A ∩ p_space p) ∩(PREIMAGE X B ∩ p_space p))
= prob p (PREIMAGE X A INTER p_space p) )
```

Now applying the CDF of the exponential distribution, we get:

$$\mathbb{P}(T_1 > t + s | T_1 > t) = \frac{e^{-\lambda(t+s)}}{e^{-\lambda t}} = e^{-\lambda s}$$

This shows that the conditional probability does not depend on t. It means that if you have waited for time t, the probability of waiting for an additional time s is

the same as the probability that you will wait for time s. In fact, the exponential distribution is the only memoryless continuous distribution, because the past has no bearing on its future behavior. Every moment is considered to be the beginning of a new random period, which has the same distribution regardless of how much time has already elapsed. Thus, we proved this property as follows

Theorem 3.5.

```
⊢ ∀X p t T1 λ s. (X (T1) = 1) ∧
counting_process X ∧ (t < T1) ∧
(PREIMAGE X A ∩ p_space p) SUBSET (PREIMAGE X B ∩ p_space p) ∧
(λn. Poisson_distr_rv X p t λ {n})
⇒ (cond_prob p (PREIMAGE (λt. X (t + s)) {&X (t + s)} ∩ p_space p)
      (PREIMAGE (λt. X t) {&X t} ∩ p_space p) =
   exp(-λ s))
```

We verify this property by directly applying Theorem 3.4 and the definitions of the Poisson distribution (Definition 3.3) and the counting process (Definition 3.4). The rest of the proof is primarily based on the conditional probability (Definition 2.5) along with some arithmetic reasoning.

This concludes our formalization of the Poisson process along with the verification of its important properties such as memoryless property and exponential interarrival time. The formal verification of these properties reassures the correctness of our formal definitions related to Poisson process. Another interesting feature of our work is the availability of the verified properties for arbitrary parameters (e.g., λ, mean arrival time). This reduces the efforts to analyze practical applications by instantiating particular values to these parameters which correspond to the give system specification. The main challenges of our formalization were to choose one general definition of the Poisson process which is applicable in different practical scenarios. This is because of the fact that different researchers and textbooks present different notions of Poisson process. Moreover, our HOL formalization intensively involves the

real analysis (e.g., limits, derivatives and transcendental functions) and probability theory (e.g., random variables and conditional probability). Note that our formalization and proof outlines can be used to formalize similar concepts from probabilistic and queueing systems analysis.

3.3 Application: Formal Probabilistic Analysis of Optical Communication Systems

Fiber optic communication systems are widely used in the domain of telecommunications, data networking and biomedicine. Its applications are widespread, ranging from basic data transmission to communication and control in very high-risk environments (chemical, nuclear, etc.). A simple fiber optic system consists of a transmitting device (laser or a light emitting diode LED) coupled to an optical fiber. The string of data to be transmitted along the fiber is in fact a series of pulses where a binary 1 is transmitted by turning on the light source for T seconds, while the transmission of binary 0 is represented by turning the source off for the same time period. At the receiver side, a photodetector is used to convert the optical signal back into a string of binary numbers. Figure 3.1 shows a simplified block diagram of this system [70]. When the received light strikes a photoemissive surface, electrons are ejected

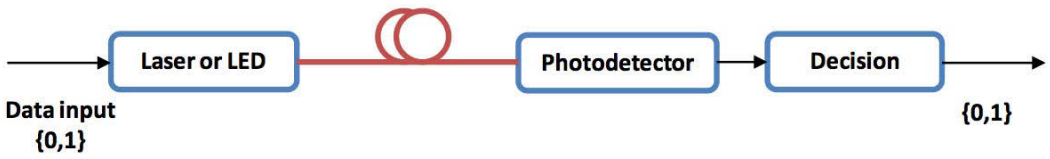

Figure 3.1: Block Diagram of an Optical Communication System [70]

randomly escaping into the space around the surface. The more intense the light that strikes the photoemissive surface, the more photoelectrons are ejected per T second interval. Therefore, we represent the number of electrons counted during a T second

interval using a Poisson random variable X changes its PMF according to the intensity of the light. Figure 3.2 depicts the random emission of electrons. When a binary

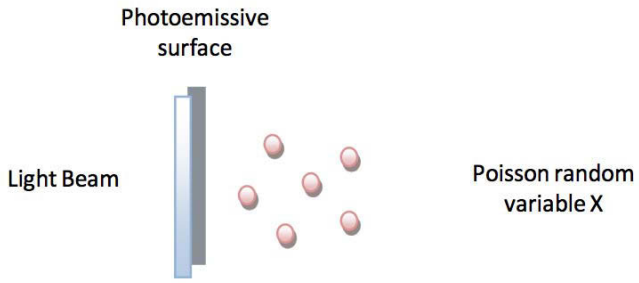

Figure 3.2: Random Emission of Electrons

0 is sent, a relatively low number of electrons is typically observed; whereas, when a 1 is sent, a higher number of electrons is typically counted. In particular, suppose the two probability mass functions are given by [70]:

$$\mathbb{P}(T = k|0\ sent) = \frac{\lambda_a^k}{k!}e^{-\lambda_a} \quad k = 0, 1, 2, ...,$$

$$\mathbb{P}(T = k|1\ sent) = \frac{\lambda_b^k}{k!}e^{-\lambda_b} \quad k = 0, 1, 2, ...,$$

These are formalized in HOL as follows:

Definition 3.5. *(Probability When Zero Is Sent)*

$\vdash \forall$ `X p A n` λ_a`.` `prob_zero_sent X p A n` $\lambda_a \Leftrightarrow$
`cond_prob p X A = Poisson_distr 1` λ_a `n`

Definition 3.6. *(Probability When One Is Sent)*

$\vdash \forall$ `X p B n` λ_b`.` `prob_zero_sent X p B n` $\lambda_b \Leftrightarrow$
`cond_prob p X B = Poisson_distr 1` λ_b `n`

where X is a Poisson random variable and p is the probability space, the parameters A and B refer to the event of sending a binary 1 and the event of sending a binary 0, respectively, while n is the number of electrons emitted during a unit of time. The

parameters λ_a and λ_b represent the average number of electrons observed when a 1 is sent and when a 0 is sent, respectively.

To decide whether a 0 or 1 was sent, we use the a maximum a posteriori probability (MAP) decision rule. Thus we calculate the a posteriori probabilities of each bit being sent given the observation of the number of electrons emitted and choose the data bit that maximizes the a posteriori probability. We decide that a binary 1 was sent if $\mathbb{P}(1\ sent|X = k) > \mathbb{P}(0\ sent|X = k)$ otherwise we decide a 0 was sent.

Definition 3.7. *(Decision Rule)*

```
⊢ ∀ A B D. Decison A B D ⇔ (D ZERO = B) ∧ (D ONE = A)
```

Definition 3.8. *(MAP Rule)*

```
⊢ ∀ A B D. MAP p X A B ⇔ cond_prob p A X > cond_prob p B X
```

From the MAP decision rule we are supposed to find the exact threshold $\frac{\lambda_a - \lambda_b}{ln\frac{\lambda_a}{\lambda_b}}$ [70]. The receiver for our optical communication system counts the number of electrons emitted and compares that number with the threshold. If the number of electrons emitted is above the threshold, we decide that a 1 was sent; otherwise, we decide that a 0 was sent. The MAP decision rule can be verified in HOL as follows:

Theorem 3.6.

```
⊢ ∀ p X A B n λa λb. prob_space p ∧ A ∈ events p ∧
B ∈ events p ∧ (prob p A = 1 / 2) ∧ (prob p B = 1 / 2) ∧ 0 < λb ∧
prob_zero_sent p (PREIMAGE X {&n} INTER p_space p) A n λa ∧
 DISJOINT ZERO ONE ∧ (λa - λb) / ln (λa / λb) < n ∧
prob_one_sent p (PREIMAGE X {&n} INTER p_space p) B n λb ∧ λb < λa
⇒ MAP p (PREIMAGE X {&n} INTER p_space p) A B
```

It is assumed that $\lambda_b < \lambda_a$, so when a 0 is sent, we tend to observe fewer electrons than when a 1 is sent and the a priori probabilities $\mathbb{P}(0\ sent)$ and $\mathbb{P}(1\ sent)$ are considered

to be equal to 1/2. The proof of this theorem is mainly done by applying some real arithmetic reasoning and by using the Bayes theorem [78] and the total probability theorem theorem which are formalized in HOL as follows:

Theorem 3.7.

```
⊢ ∀ A B p. prob_space p ∧ A ∈ events p ∧ B ∈ events p
⇒ (cond_prob p B A = cond_prob p A B * prob p B / prob p A)
```

The Bayes law relates the conditional and marginal probabilities of two random events A and B. Mathematically, it is expressed as $\mathbb{P}(A \mid B) = \frac{\mathbb{P}(B|A)*\mathbb{P}(A)}{\mathbb{P}(B)}$

Theorem 3.8.

```
⊢ ∀ p. prob_space p ∧ PREIMAGE X {&n} ∩ p_space p ∈ events p ∧
(∀x. x ∈ {ZERO;ONE} ⇒ D x ∈ events p) ∧ FINITE {ZERO;ONE} ∧
(∀a b. a ∈ {ZERO;ONE} ∧ b ∈ {ZERO;ONE} ∧ a ≠ b
⇒ DISJOINT (D a) (D b)) ∧ (BIGUNION (IMAGE D {ZERO;ONE}) = p_space p)
⇒ (prob p (PREIMAGE X {&n} ∩ p_space p) = SIGMA (λi. (prob p (D i)) *
(cond_prob p (PREIMAGE X {&n} ∩ p_space p) (D i))) {ZERO;ONE})
```

In the above theorem, (*PREIMAGE X* {&n} ∩ *p_space p*) represents an event, whereas D represents a sequence of sets. The second and the third assumptions ensure that all events are in the event space. With the fourth assumption, we guarantee that the two sets are disjoint, i.e., their intersection is an empty set. The last assumption ensures that the union of the elements of the sequence D gives the sample space *p_space p*. In this case, the law of total probability helps us to find the probability of a particular event based on the conditional probability of that same event given that some events form a partition of the sample space.

Similarly, we can find the probability of error invoking the total probability theorem and the concepts of conditional probability:

Definition 3.9. *(Probability Error When One Is Sent)*

$\vdash$ $\forall$p X A n λ_a ERROR n_0.

prob_error_one_sent p X A n λ_a ERROR n_0

$\Leftrightarrow$ (cond_prob p ERROR A =

SIGMA (λ i. λ_a pow i * exp (-λ_a) / &FACT i) (count_mn 0 n_0))

Definition 3.10. *(Probability Error When Zero Is Sent)*

$\vdash$ $\forall$p X B n λ_b ERROR n_0.

prob_error_zero_sent p X B n λ_b ERROR n_0

$\Leftrightarrow$ (cond_prob p ERROR B =

SIGMA (λ i. λ_b pow i * exp (-λ_b) / &FACT i) (count_mn 0 n_0))

We note that errors can occur in two ways. First a 0 could be sent and the number of electrons observed could fall above the threshold, causing us to decide that a 1 was sent. Likewise, if a 1 is actually sent and the number of electrons observed is low, we would mistakenly decide that a 0 was sent.

Recalling the concepts of conditional probability, we know that

$$\mathbb{P}(error) = \mathbb{P}(error|0sent) * \mathbb{P}(0sent) + \mathbb{P}(error|1sent) * \mathbb{P}(1sent)$$

Hence, if we consider n_0 to be the threshold with which we compare X to decide which data bit was sent, then with some mathematical reasoning, we can calculate the probability of error for our optical communication system

$$\mathbb{P}(error) = \frac{1}{2} - \frac{1}{2} * \sum_{k=0}^{n_0} \frac{\lambda_a^k * e^{-\lambda_a} - \lambda_b^k * e^{-\lambda_b}}{k!}$$

This expression can be verified using HOL as the following theorem:

Theorem 3.9.

$\vdash \forall$ p X A B ERROR n λ_a λ_b. prob_space p $\wedge$ A $\in$ events p

$\wedge$ ERROR $\in$ events p $\wedge$ Decison A B D $\wedge$

($\bigcup$(IMAGE D {ZERO;ONE}) = p_space p) $\wedge$

B $\in$ events p $\wedge$ (prob p A = 1 / 2) $\wedge$ (prob p B = 1 / 2) $\wedge$ 0 < λ_b

$\wedge$ prob_error_zero_sent p X A n λ_a ERROR $\wedge$ DISJOINT ZERO ONE $\wedge$

prob_error_one_sent p X B n λ_b ERROR n_0 $\wedge$ λ_b < λ_a

$\Rightarrow$ prob p ERROR = 1 / 2 - 1 / 2 *

SIGMA(λx. λ_a pow x * exp (-λ_a) -

λ_b pow x * exp(-λ_b)/&FACT x) (count_mn 0 n_0)

We proceed with the verification of this theorem by first rewriting the goal using Definitions 3.9 and 3.10 then we used some real analysis and properties of transcendental functions. Later, we used the constraints of our goal, the definitions and the probability theorems previously proved in order to reach the final result.

This application illustrates how our formalization of Poisson process and Poisson distribution can be used to reason about real world applications. Conducting the analysis within the sound core of a theorem prover helped to add more trust to the proved results. This is because of the logical foundations of higher-order logic theorem proving systems. Indeed all the steps performed in the proof of the application can be traced back to the logical axioms and inference rules of HOL4. On the other hand, simulation and paper-and-pencil based proof are not capable of providing such soundness. For example, simulation based results are only valid for the particular values of inputs and it is very difficult of certify the correctness of paper-and-pencil based proofs due the human-error proneness.

This is obviously not a large application but it serves as an example to illustrate the usefulness of the framework presented in this thesis. We were able to verify the desired probabilistic characteristics as *generic* theorems that are universally quantified for all values of variables (e.g. n, λ_a, λ_b). These variables can also be specialized to specific

values to obtain corresponding precise conditional probabilistic and probability errors values. In fact knowing the exact probability error for a given scenario makes it possible to change the rates by adjusting the intensity of the laser or LED. These proofs required approximately 300 lines of HOL code. The upside is that these results can be reused in several other engineering applications.

3.4 Summary

In this chapter, we presented a higher-order-logic formalization of the Poisson Process with an infinite state space. Both homogeneous and inhomogeneous Poisson processes can be modeled based on this formalization. We also presented a higher-order-logic formalization of the Poisson distribution which can be regarded as the first step towards a successful theorem-proving based analysis of discrete distributions. This formalization is flexible and more realistic since it is time dependent.

Some of the most interesting properties such as the Markov property and the exponential interarrival time were formally verified in HOL. Then, we used our formalization to analyse an optical communication system. This channel is formalized as a Poisson process model using higher-order logic. Then, two interesting properties of this channel were proved based on this model. This example mainly illustrates a flow of a verification process of a Poisson process model using theorem proving and it shows the usefulness of our formalization of the Poisson distribution.

Chapter 4

Case Study: Formalization of the the M/M/1 queue based on CTMC

In this chapter, we make use of the formalizations of the Poisson process in HOL, CTMC and birth-death process to provide a higher-order-logic formalization of M/M/1 queueing system.

4.1 Queueing Theory

Nowadays, concrete numbers provided by quantitative analysis play an important role in the development of a wide range of applications in many spheres of life such as computer networks, telecommunications, manufacturing systems, transportation, logistics, etc. Queueing theory is quite common in all of these fields, consequently, numerous properties such as performance metrics, e.g., throughput, service times and waiting times require accurate and reliable modeling of these systems. In fact, Erlang [32] was the first to ever raise the congestion problems of queueing theory in telephone exchanges. Later researchers were inspired by his work and started working on queueing problems using probabilistic methods. Queueing networks and Markov chains have now become a field of applied probability and statistical methods and both of them provide effective and practical models to analyze a wide range of

applications.

Queueing systems consist of service centres that provide any kind of service to arriving customers. It is quite evident from the above mentioned systems that arrivals may demand a service from a finite-capacity resource. The general assumption is that one station cannot at the same time serve two or more arrival entities. In this case, conflicts for the use of the resource will arise and arrivals are more likely to wait on queues in front of the servers, hence the name *queuing systems*. When one of the resources is free, a waiting customer is taken over from the queue according to the pre-defined discipline and it gets served. In the case of a finite queue, an arrival can be rejected. It is necessary to take into consideration that the term customer does not necessarily imply a human customer; any entity which needs a service of some sort is considered a customer.

Kendall [26] was the first to introduce the $A/B/C$ queueing notation in 1953. The first letter specifies the interarrival time distribution or the arrival pattern and the second one the service time distribution. A and B are described by symbols that represent probability distributions. For example, M stands for markovian or memoryless distribution, D for deterministic, and GD for general distribution. In the case of general distribution, results can be applicable to all probability distributions. Finally, the third letter specifies the number of servers. The notation can be extended with two extra letters Y and Z to put a restriction on the system capacity and the queue discipline, respectively. If the queue discipline is first come, first served (FCFS) then the standard is to omit the symbol. Hence M/M/1 denotes a system with Poisson arrivals with parameter λ, exponentially distributed service times with parameter μ which are assumed to be independent and identically distributed, and a single server that serves the entity which is the first to reach it (FIFO principle).

$M/M/1$ queue is the simplest model in the queueing theory. It has an infinite number of states since the buffer may contain any number of customers and allow transitions in continuous time. CTMC is one of the most efficient and powerful technique for the investigation of the $M/M/1$ queue or any other queueing system

[92].

Birth-death process plays a very important role in modeling elementary queueing systems such as the $M/M/1$ queue where the only difference between the two models is a variant and a constant rate, respectively. In addition to this, a birth-death process is indeed a special case of CTMC where the states represent the current size of a population and where the transitions, across an infinitesimal time interval h are limited to births and deaths.

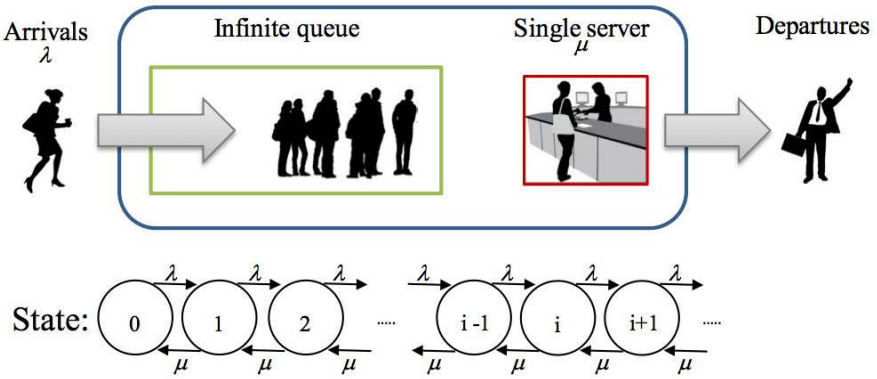

Figure 4.1: M/M/1 Queueing System

Based on Figure 4.1, each system state has adjoined a certain number denoting the number of units in the system. The arrows point to the direction of possible transitions from the state system with transition rates λ and μ. When an entity joins the system, then the graph state changes from $(nton + 1)$ or when the entity is served then the state changes from $(nton - 1)$. Thus, the state space is typically the set of all integers or a subset of the integers.

4.2 Formalization in Higher-Order-Logic

4.2.1 Formalization of Continuous-time Markov Chain

Given a probability space, a stochastic process $X_t : \Omega \to S$ represents a sequence of random variables X, where t is a real number. The set of values taken by each $X_1(t)$

is a discrete state space (finite or countably infinite). This allows the process to make its transitions at any moment along the positive real axis and not necessarily at predefined epochs. Based on these definitions, a Markov process can be defined as a stochastic process with the Markov property. A Markov process is a CTMC if [88]:

$$\mathbb{P}(X(t+h) = j \mid X(t) = i, X(u) : 0 \leq u < t) = \mathbb{P}(X(t+h) = j \mid X(t) = i)$$

The main objective is to place conditions on the holding times to ensure that the continuous time process satisfies the Markov property where the future, given the present state, $X(t)$ is independent of the past, $X(u) : 0 = u < h$. The formal definition is given by:

Definition 4.1. *(Markov Property)*

```
⊢ ∀ X p i j. Marcov_Property X p i j ⇔
(real_random_variable X p) ∧
(BIGINTER (IMAGE (PREIMAGE (λ k. X u) {&k} INTER p_space p)
(count i)) <> 0)
cond_prob p (PREIMAGE (λ t. X (t + h)) {&j} INTER p_space p)
((PREIMAGE (λ t. X t) {&i} ∧ INTER p_space p) INTER
 BIGINTER (IMAGE (PREIMAGE (λ k. X u) {&k} INTER p_space p)(count i)))
= cond_prob p (PREIMAGE (λ t. X (t + h)) {&j} INTER p_space p)
(PREIMAGE (λ t. X t) {&i} INTER p_space p)
```

for all states i and j and for all times $h > 0$ and $t > 0$, we are conditioning on the values of $X(u)$ for all times u in a subset of past times in addition to the value at the current time t. In this case, we assume that any arbitrary subset of $[0, t) \equiv u : 0 \leq u < t$ is a finite subset.

In the case of a CTMC, the Markov property can be defined in the same way as for a DTMC; assuming that the distribution of the future, given the present state $X(t)$, does not depend on the present time t or the amount of time h that has elapsed since time t, but only depends on the present state $X(t) = i$. This definition makes

the analysis of CTMC more difficult technically because there is no longer a fixed and small epoch of time until the next transition, in fact there is a continuum of such possible times t. Thus the reasoning is related to the holding or sojourn time where the remaining holding time must only depend (in distribution) on the current state i and be independent of its time age.

To define a CTMC we have to specify both the initial distribution which gives the probability of initial occurrence for every state; and the transition probability function which is a continuous function of t that gives the probability of going from state i to state j. In general, it is hard to determine the transition probability function in a efficient closed form [19]. In simple cases we can define it as:

$$\mathbb{P}(X(t+h) = j | X(t) = i) = \begin{cases} \lambda_{i,j}\ h + o_1(h), & \text{if } i \neq j; \\ 1 - \lambda_i\ h + o_2(h), & \text{if j = i;} \end{cases}$$

where $\lambda_{i,j}$ is the local rate interpreted as the transition rate out of state i to a state j given that $X(t) = i$ and $\lambda_i = \sum_{i \neq j} \lambda_{i,j}$. The function $o(h)$ is understood to be a quantity which is asymptotically negligible as $h \to 0$ after dividing by h, formally $f(h) = o(h)$ as $h \to 0$ if $\frac{f(h)}{h} \to 0$ as $h \to 0$. The exponential holding time will end, independent of the past, in the next h units of time with probability $\lambda_{i,j}$. In this case, the chain cannot go anywhere in zero time and the probability of an event is zero, when this event is not in the event space.

Definition 4.2. *(CTMC Transition Probability Function)*

```
⊢ ∀ X p t h i j λ_{i,j} λ_i. CTMC_Trans_Fun X p t h i j λ_{i,j} λ_i
 ⇔ ∃ o_1 o_2.
       if i ∈ space s ∧ j ∈ space s then
       cond_prob p (PREIMAGE (λ t. X (t + h)) &j INTER p_space p)
       (PREIMAGE (λ t. X t) &i INTER p_space p)=
       if (&i ≠ &j) then λ_{i,j} * h + o_1 h
       else 1 - λ_i * h + o_2 h
```

From this definition, we easily extract the transition probability of the Poisson process. In fact, the Poisson process is a CTMC with one-step transition probabilities and exponential sojourn rates $\lambda_{i,j} = \lambda$ which are regular on the state space of nonnegative integers. This is an easy consequence of the independent-increments property of the Poisson process that we already proved.

The main difference between the poisson process and CTMC is that when a jump occurs in the case of the latter, the location where the chain jumps is not deterministic. However in the case of the Poisson process, the jump occurs exclusively from state i to state $i + 1$.

Now, the continuous-time Markov chain (CTMC) can be formalized as follows:

Definition 4.3. *(Continuous-Time Markov Chain)*

```
⊢ ∀ X p I F i j. CTMC X p I F i j λ_{i,j} λ_i ⇔
Markov_Property X p i j ∧
(∃ x. x ∈ space s ⇒ {x} ∈ subsets s) ∧
( L x = distribution p (↷ X 0) i) ∧
(F t h i j = CTMC_Trans_Fun X p t h i j λ_{i,j} λ_i)
```

The first condition in this definition makes use of the Markov property while the second one ensures that any observable events in the state space s are discrete in the event space subsets of s. The third and forth conditions designate the initial distributions and the transition probabilities of the chain, respectively .

A CTMC does not need to be time homogeneous but homogeneous CTMCs play an important role in different application areas. By time homogeneity we mean that, whenever the process enters a state i, its probability distribution from that point is the same as if the chain started in state i at time 0. Thus the holding time distribution is the same every time the chain enters state i. We will consider the special case of homogeneous transition probabilities (sometimes referred to as stationary transition probabilities) while defining time homogeneous CTMC [97].

$$\mathbb{P}(X(t+h) = j \mid X(t) = i) = \mathbb{P}(X(h) = j \mid X(0) = i)$$

We formalized the time homogeneous CTMC as follows:

Definition 4.4. *(Time-Homogeneous Continuous-Time Markov Chain)*

$\vdash \forall$ `X p I F i j` $\lambda_{i,j}$ λ_i `. TH_CTMC X p I F i j` $\lambda_{i,j}$ λ_i $\Leftrightarrow$
`CTMC X p I F i j` $\wedge$
`(CTMC_Trans_Fun X p t h i j` $\lambda_{i,j}$ λ_i `= ctmc_fun X p 0 h i j` $\lambda_{i,j}$ λ_i`)`

This definition holds for all states i and j and for all times $t > 0$ and $h > 0$. The independence of h characterizes the homogeneity.

4.2.2 Formalization of the Birth-Death Process

The Birth-Death process is a special case of CTMC, where the states represent a current number of customers in the system and the transitions are limited to births and deaths. The process goes from state i to state $i+1$ when a birth occurs. Similarly, it goes from state i to state $i-1$ when a death occurs. It is assumed that the birth and death events are independent of each other. The birth-death process is characterized by the birth rate $\lambda_{i,i+1}$ and death rate $\lambda_{i,i-1}$, which vary according to the state i of the system.

In order to define the transition probability function, we will follow the same approach already used to define the transition probability function of both the Poisson process and CTMC. Let $h > 0$ be a small interval of time, during which there exist observable changes in a chain. Our main goal is to calculate the conditional probability of changes occurring at time $t + h$ compared to t. While formalizing the poisson process and CTMC we mentioned that it is impossible to observe more than one event over such a short interval of time h [96]. The transition probability function of the Birth-Death process is defined as follows :

$$\mathbb{P}(X(t+h) = j | X(t) = i) = \begin{cases} \lambda_{i,i+1}h + o_1(h), & \text{if j = i + 1;} \\ \lambda_{i,i-1}h + o_2(h), & \text{if j = i - 1;} \\ 1 - \lambda_i h + o_3(h), & \text{if j = i.} \end{cases}$$

Over the time interval h and given that there are currently i costumers in the system, the probability that there will be $i+1$ costumers is represented by the probability of one birth and no death which is the main probabilitic component plus other combinations with very small occurring chances represented by $o(h)$. In this case, the $o(h)$ term represents the fact that there are two births and one death, three births and two deaths, etc. The same explanation applies to decreasing the number of customers to $i-1$.

Now, The transition probability function of the Birth-Death process and the Birth-Death process itself can be formalized as follows:

Definition 4.5. *(Birth-Death Transition Probability Function)*

$\vdash \forall$ `X p t h i j` $\lambda_{i,j}\ \lambda_i$`.` `BD_Trans_Fun X p t h i j` $\lambda_{i,j}\ \lambda_i \Leftrightarrow$
`if (j = i + 1) then CTMC_Trans_Fun X p t h (i + 1) i` $\lambda_{i,i+1}\ \lambda_i$
`else if (j = i - 1) then CTMC_Trans_Fun X p t h (i - 1) i` $\lambda_{i,i-1}\ \lambda_i$
`else CTMC_Trans_Fun X p t h (i - 1) i` $\lambda_{i,i}\ \lambda_i$

In this definition we are using the CTMC transition probability function given in Definition 4.2. $\lambda_{i,j}$ is the transition rate out of state i to the state j given that $X(t) = i$ and $\lambda_i = \sum_{i \neq j} \lambda_{i,j}$ where in this case it will be equal to $\lambda_{i,i+1} + \lambda_{i,i-1}$.

Definition 4.6. *(Birth-Death Process)*

$\vdash \forall$ `X p I F i j` $\lambda_{i,j}\ \lambda_i$ `. BD_CTMC X p t h I F i j` $\lambda_{i,j}\ \lambda_i \Leftrightarrow$
`TH_CTMC X p I F i j` $\lambda_{i,j}\ \lambda_i \wedge$ `BD_Trans_Fun X p t h i j` $\lambda_{i,j}\ \lambda_i$

It is clear that the probabilities of customers increasing or decreasing by 1 are proportional to the length of the interval. These definitions support the notion that the events are rare and almost exclude the possibility of simultaneous occurrence of two or more events. Basically, only one event can occur in a very small interval of time h. And even though the probability for more than one event is non-zero, it is negligible [84]. We now turn to the M/M/1 queue, which combines CTMC with the Poisson

process and the Birth-Death process. Most properties of the M/M/1 queue follow directly from results about the latter formalizations.

In an M/M/1 queueing system, the requests arrive according to a Poisson process with rate λ, justifying that the interarrival times are independent and represent exponentially distributed random variables. The service times are also assumed to be independent and exponentially distributed with parameter μ. Thus, all involved random variables are supposed to be independent of each other. While investigating the transition probabilities during a very short period of time h, we can see that by using the independence assumption, the probability of having $i+1$ customers in the system at time $t+h$ considering that the number was i at a time t, will be equal to $\lambda h + o(h)$. The first term is equivalent to the probability that during the time h one customer has arrived and no service has been finished. While the second term is equivalent to all other possible scenarios. Basically we got this second term due to the property of the Poisson process. Similarly, the transition probability from state i into state $i-1$ during h can be expressed by $\mu h + o(h)$. Therefore, we have:

$$\mathbb{P}(X(t+h) = j | X(t) = i) = \begin{cases} \lambda h + o(h), & \text{if j = i + 1;} \\ \mu h + o(h), & \text{if j = i - 1;} \\ 1 - (\lambda + \mu) + o(h), & \text{if j = i;} \end{cases}$$

The M/M/1 queue is indeed a simple birth-death process with constant rates $\lambda_{i,j} = \lambda$.

Definition 4.7. *(M/M/1 Queue Transition Probability Function)*

```
⊢ ∀ X p t h i j λ μ. MM1_Trans_Fun X p t h i j λ μ ⇔
if (j = i + 1) then Poisson_Trans_Fun X p t h (i + 1) i λ
else if (j = i - 1) then BD_Trans_Fun X p t h 1 i μ λ
else BD_Trans_Fun X p t h j i λ μ
```

In this definition we make use of all the transition probability functions formalized before, i.e., the transition probability function of the Poisson process and the Birth-Death process, to highlight the different probabilities in the M/M/1 queue case. Thus the M/M/1 queue can now be defined as follows:

Definition 4.8. *(M/M/1 Queue)*

$\vdash \forall$ `X p I F i j` $\lambda_{i,j}$ λ_i `. MM1_Queue X p t h I F i j` λ $\mu \Leftrightarrow$
`BD_CTMC X p I F i j` λ $\mu \wedge$ `MM1_Trans_Fun X p t h i j` λ μ

Thus, the M/M/1 queue will inherit all the specifications of the CTMC and Poisson process. The queue is assumed to have infinite capacity which means that requests for service will never be discarded or affect the likelihood of other requests joining the queue. In addition to this, the Poisson process is able to generate an infinite number of requests which means that the arrival of a request to the system does not influence upcoming arrivals.

4.3 Formal Verification of the M/M/1 Queue Properties

Using the formal definition of the M/M/1 queue, we proved its most important properties which are frequently used in the analysis of many systems modeled as an M/M/1 queue. These properties include the mean number of costumers, the mean response time and the mean waiting time in the queue.

4.3.1 The Mean Number of Costumers

We first start by defining ρ which is the traffic intensity. It is defined as the average arrival rate λ divided by the average service rate μ. The average arrival rate should always be less than the average service rate in order to maintain a stable system, which means ρ should always be less than one. Hence, we can introduce Little's Formulae (or Little's Law) [6] which states that over a period of time T, the mean number of arrivals in the system is λ multiplied by the average time a customer spends in the system. This formulae holds true only in the case of a steady state queuing system. Thus we formally defined the steady-state limiting probability of the system being in state n for the M/M/1 queue:

Definition 4.9. *(Steady State Probability for the M/M/1 Queue)*

$$P^n = (1 - \frac{\lambda}{\mu}) * (\frac{\lambda}{\mu})^n$$

```
⊢ ∀ X n p λ μ. Steady_state_eq X n p λ μ ⇔
  real_random_variable X p ∧
  (prob p (PREIMAGE (λt. X t) {&n} INTER p_space p) =
  (1 - λ/μ) * (λ/μ) pow n)
```

Consequently, we formally verified that the mean number of customers in a steady-state system is equal to $\frac{\rho}{1-\rho}$.

$$E(N) = \frac{\rho}{1 - \rho}$$

Theorem 4.1.

```
⊢ ∀ p X A B. prob_space p ∧ MM1_Queue X p t h I F i j λ μ ∧
0 < λ  ∧ 1 > ρ ∧ Steady_state_eq X n p ⇒
SIGMA (λn. n * Normal (distribution p X {&n})) (IMAGE X (p_space p)) =
NORMAL (ρ / (1 - ρ))
```

In order to prove this theorem we used some summation and derivative properties. The variable ρ represents the fraction of time that the server is in use and is therefore a measure of efficiency. It is important to note that it is also the steady state probability that the transmission line is in use. For a lossless system (one that does not drop arriving jobs) this probability is given by $1 - \mathbb{P}(0)$.

4.3.2 Mean Response Time

The mean response time or sojourn time is the total time a customer spends in the system. We can deduce the mean response time in the queue using Little's Formulae which relates the average number of customers in the system to the average time spent in the system through that arrival rate λ

$$E(T) = \frac{1}{\mu - \lambda}$$

Theorem 4.2.

⊢ ∀ p X A B n. prob_space p ∧ MM1_Queue X p t h I F i j λ μ ∧
1 > ρ ∧ T = N / λ ∧ 0 < λ ⇒ T = NORMAL (1 / (μ - λ))

Note that the mean response time is a very important factor in heavy traffic queues. In our formalization, all parameters do not depend on a scheduling discipline since we modelled this as a first come first serve (FCFS) system. However, whenever a customer arrives at an FCFS queue, it will find an other customer already being served. This latter customer has already completed some service time before and has a pending residual service time left to complete. In our formalization, we did not model the residual service time and we blindly considered it to be a part of the response time. This is only valid for an exponentially distributed service time but not otherwise.

4.3.3 Mean Waiting Time in the queue

The mean waiting time in the M/M/1 queue consists of the residual service time of the customer currently under service and the time needed to serve the customer waiting. In other words we can express it based on the average time spent in the system and the average time a customer is being served.

$$E(w) = E(T) - \frac{1}{\mu}$$

Theorem 4.3.

⊢ ∀ p X A B n. prob_space p ∧ MM1_Queue X p t h I F i j λ μ ∧
1 > ρ ∧ W = T - 1 / μ ∧ 0 < λ ⇒ w = NORMAL (ρ / (μ - λ))

In addition to the above three properties, it is possible to compute the average number in the queue and the average time spent queueing (without being served). Littles

formula is very important as it gives the relationship between the steady-state average number of customers in the system, the steady-state arrival rate, and the steady-state customer delay.

4.4 Applications

In this section, we present the formal analysis of two real-world applications, i.e., the modeling and analysis of a single airport runway and a network of queues.

4.4.1 Airport Runway Modeling and Analysis

Generally, modeling and analysis of runways are considered as a critical element in the design life cycle of airports and corresponding control softwares which are deployed in control towers. In fact, it is very important to evaluate the performance of a single runway due to different factors associated with safety and cost. One of the most traditional ways is to model a single runway (as shown in Figure 4.2) as a single server queue, i.e., M/M/1. Consequently, it is possible to determine some important runway performance metrics such as runway utilization, expected number of airplanes waiting to land and expected waiting time [14].

Figure 4.2: A Schematic Model of Single Airport Runway

In the following, we present the formal modeling and verification of these properties in HOL4 using our formalization.

Definition 4.10. *(Single Runway Model)*

```
⊢ ∀ X p I F i j λ μ.
runway_model X p I F i j λ μ = MM1_Queue X p I F i j λ μ
```

based on our definitions, an airport runway can be defined as an M/M/1 queue. This means that this model will inherit all the properties of an M/M/1 queue. Thus it is assumed that the arrival is a Poisson process and the distribution for landing times is exponential. In our application, airplanes are supposed to arrive at the rate of 15 per hour and it is estimated that each landing takes 3 minutes which means that the service rate is 20 per hour. The mean number of customers in the system can be found as follows

$$E(N) = \frac{\rho}{1-\rho} = 3$$

Theorem 4.4.

```
⊢ ∀ p X. prob_space p ∧ MM1_Queue X p t h I F i j 15 20 ∧
Steady_state_eq X n p ⇒
SIGMA (λn. n * Normal (distribution p X {&n})) (IMAGE X (p_space p)) =
NORMAL (3)
```

Similarly we can find the expected waiting time

$$E(w) = \frac{\rho}{\mu-\lambda} = 9\ minutes$$

Theorem 4.5.

```
⊢ ∀ p X. prob_space p ∧
MM1_Queue X p t h I F i j 15 20 ⇒  w = NORMAL (9)
```

We verified above mentioned properties using Theorems 4.1 and 4.3, respectively. The above theorems are very important performance metrics for any single runway model. Moreover, their verification in the HOL theorm prover increases the trust as well as provide a complete certification proof. Finally, these properties can be used for different values of λ and μ, which shows the reusability of our formalization.

4.4.2 Network of Queues

Generally, queues can interact in the sense that a traffic stream departing from one queue enters one or more other queues. The network of queues model [13] focuses on nodes with finite capacities and studies buffering and waiting behaviors from a stochastic perspective. It is based on relatively simple topologies and lacks the multi-hop flow routing dimension. The nature of this network has the unfortunate effect of complicating the arrival processes at downstream queues. The difficulty is that the customers interarrival times become strongly correlated with the service time once customers have traveled beyond their entry queue. As a result it is very hard to carry out a precise and effective analysis using numerous analysis techniques. Thus we chose to model this network based on our formalization of the M/M/1 queue.

A Network of Queues can be seen as a set of two nodes tandem networks as shown in Figure 4.3. The figure shows a two-stage tandem network composed of two nodes

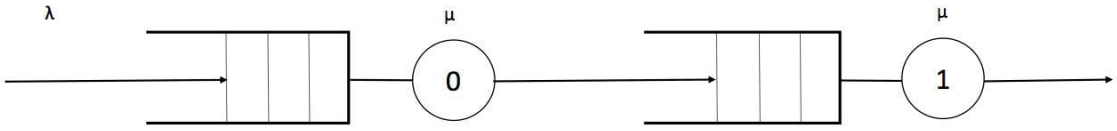

Figure 4.3: Two nodes Tandem Network [13]

with service rates μ_0 and μ_1, respectively. The external arrival rate for node 0 is λ and the arrival process is Poisson. Both nodes are M/M/1 queues.

As an indication of the difficulty of analyzing queueing network problems involving dependent interarrival and service times, no analytical solution is known for even the simple tandem queueing problem of Figure 4.3 involving Poisson arrivals and exponentially distributed service times. In real situations where service times or number of customers in different queues are correlated, the average delay per customer can be smaller than in the idealized situation where there is no such correlation. The reverse is true under light traffic conditions. In this case, we consider a network of queues with independent service times and light traffic. As a result of this assumption, the occupancy distribution in the two queues is the same as if they were independent

M/M/1 queues in isolation.

Consequently, a node in the tandem network can be modeled as follows

Definition 4.11. *(Network Node)*

$\vdash \forall$ `X p i j I F` $\lambda_i\ \mu_i\ N_i$ `.`

`node_model X p i j I F` $\lambda_i\ \mu_i\ N_i$ `= MM1_Queue X p i j I F` $\lambda_i\ \mu_i$

In this example, we assume that the arrival process at node 1 is also a Poisson process with rate λ. Our M/M/1 results can be applied to a group of random variables as well, which gives rise to joint probability distributions of the numbers in both nodes.

$$\mathbb{P}(N_0, N_1) = (1 - \rho_0)\rho_0^{N_0}(1 - \rho_1)\rho_1^{N_1}$$

Theorem 4.6.

$\vdash \forall$ `p X. prob_space p` $\wedge$ `node_model X p i j I F` $\lambda_0\ \mu_0\ N_0\ \wedge$
`node_model X p i j I F` $\lambda_1\ \mu_1\ N_1\ \wedge$ `Steady_state_eq X n p` $\wedge\ \rho_i$ `=` $\lambda_i/\mu_i\ \wedge$
ρ_i `<` $1\ \wedge$ `(events p = POW (p_space p))` $\Rightarrow$
`joint_distribution p X X` $\{N_0\}\ \{N_1\}$ `=`

`(1 -` ρ_0`) * ` ρ_0`^`N_0 `* (1 -` ρ_1`) * ` ρ_1`^`N_1

Let $\rho_i = \frac{\lambda_i}{\mu_i}$ be the corresponding utilization factor to a node i. We assume that $\rho_i < 1$ for stability reasons. The most important assumptions in the theorem above is the fact that we assume N to be the average number of customers in one queue. This means that we are assuming that the arriving process takes a random look, which is true only for the Poisson process that has memoryless arrivals applied for the Markov Chain solution.

It is possible to find the average delay per customer when we know the average number of customers in each queue. Assuming that γ is the total arrival rate in the system, then we can formalize this property as follows

$$D = \frac{1}{\gamma}\sum_i \frac{\rho}{1 - \rho}$$

Theorem 4.7.

$\vdash \forall$ p X. prob_space p $\wedge$ node_model X p i j I F λ_i μ_i N_i $\wedge$
ρ_i = λ_i/μ_i Steady_state_eq X n p $\wedge$
γ = SIGMA (λi. NORMAL (λ_i)) (count i) $\wedge$
N = SIGMA (λi. NORMAL (ρ / (1 - ρ))) (IMAGE X (p_space p)) $\Rightarrow$
D = 1/γ * SIGMA (λi. NORMAL (ρ / (1 - ρ)) (IMAGE X (p_space p))

In this theorem we are assuming that the transition times of all customers from one node to another are exponentially distributed. Also the transition times of all customers are independent including the transition times of the same customer at two different links from one node to another. The proof of this theorem is mainly based on the M/M/1 queue definition along with some arithmetic and probabilistic reasoning.

The single server queuing analysis can be used to estimate the average waiting time, the number of customers and even the joint distribution in a network of queues. Queueing theory is also the primary methodological framework for analyzing network delay. Its use often requires simplifying assumptions since, unfortunately, more realistic assumptions make meaningful analysis extremely difficult. For this reason, it is sometimes impossible to obtain accurate quantitative delay predictions on the basis of queueing models while using simulation techniques. Nevertheless, using our approach, these models often provide a basis for adequate delay approximations, as well as valuable qualitative results and worthwhile insights. The ability to express and verify generic properties, quantified for all values of the variables, is the main strength of theorem proving as can be seen from the above definitions and theorems. All properties, once verified, can hold for any number of nodes and customers and can be further specialized to obtain expressions and values for particular scenarios. Moreover, some important underlying assumptions, e.g., the fact that a queue should always be stable, are always found thanks to the fact that every single step of the proof needs to be derived from axioms or previous theorems using inference rules. To

the best of our knowledge, this is the first time the properties of an M/M/1 queue or any kind of queue, have been analyzed using theorem proving.

4.5 Summary

In this chapter, we used the formalization of Poisson process and its transition probability function to provide a higher-order-logic formalization of the main concepts of M/M/1 queue. We also formalized the definitions of CTMC, time-homogenous CTMC and birth death process. Based on the latter formalization, we introduced the most commonly used definition of a one server queue as well as its transition probability function. To facilitate the probabilistic analysis of queueing model, we verified the most important properties of an M/M/1 queue, which can be found in most textbooks and are frequently used in real-world applications. These properties represent the foundation of many queueing systems, they can also be used to derive more interesting properties such as the average time spent queueing or being served. The airport runway modeling and the network of queues are a typical M/M/1 queue models and many more complicated systems can be based on such a simple structure. For this reason, we analyzed some properties of these two basic applications. In a first step, they were both formalized as an M/M/1 queue model using higher-order logic. Then, their most interesting properties were proved based on this model. These two examples mainly illustrate a flow of the complete verification process of an M/M/1 queue using theorem proving and it shows the usefulness of our formalization.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Numerous important properties of engineering systems such as reliability, availability, and performance metrics mandate the need for accurate modeling and analysis. Continuous time Markov chains offer an effective and practical modeling solutions for a variety of safety-critical domains, such as medicine, transportation and communication systems, etc. However, existing computer-based techniques for conducting these analysis, i.e., simulation, Computer Algebra Systems (CAS) and model checking, can hardly guarantee accurate results. Their results often include approximations due to the utilization of numerical methods or suffer from the state-explosion problem as in the case of model checking. To overcome the limitations of the above mentioned techniques, we propose to use higher-order-logic theorem proving to facilitate the formal analysis of systems modeled as CTMCs and to deliver more accurate and trusted results.

In this thesis, we have presented an application of formal methods in the area of analyzing Markovian systems. In particular, we have developed a framework for accurate and reliable analysis of systems which can be modeled using CTMC. This formalizations also offers the capability of formally evaluating the performance of diverse systems which are described as queues. The higher-order logic theorem proving

approach guarantees generic, accurate and reliable results compared to traditional paper-and-pencil analysis, simulation techniques or computer algebra systems. We believe that a formal analysis based on our development will be free of approximation and precision problems due to the soundness nature of the higher-order logic environment. Thus, our proposed approach can be used in formal performance analysis of safety critical engineering and scientific applications.

The main purpose of this thesis was to develop an infrastructure that can be used to perform formal analysis of queueing systems based on CTMCs. Towards this goal, we built upon the available probability theory of HOL4 to formalize the Poisson process along with the verification of some of its important properties. We have been able to use the formalization of Poisson process to formally verify the error probabilities of optical communication systems as an illustrative case study as well. Furthermore, we formally defined the continuous-time Markov chains which further allowed us to represent birth-death process in higher-order logic. Finally, we used these foundations to formalize a generic model of an M/M/1 queue. Based on this formalization, we have been able to formally verify and model a single airport runway as well as a network of queues, which are expressed as M/M/1 queues. These applications highlight the benefits of the formalization of M/M/1 queues and the formal verification of their properties using a higher-order-logic theorem prover.

This work was conducted using the HOL4 kananaskis 9 version of the theorem prover and the main reason behind this choice was to be able to utilize available higher-order-logic formalizations of the measure and probability theories along with the conditional probability which we ported from HOL4 kananaskis 7 to HOL4 kananaskis 9.

The main challenge of our work was to describe the poisson process, CTMC and the M/M/1 in proper and flexible way in the higher-order logic. The proof script of the formalization and verification of the notions presented in this thesis require around 3000 lines of HOL4 code available at `http://hvg.ece.concordia.ca/projects/prob-it/CTMC.html`.

To the best of our knowledge, no queueing system has been formalized in any of the existing theorem prover. Due to the formal nature of our models, the analysis conducted by this framework will be accurate and reliable even in short intervals. This approach can thus be of great benefit for the analysis of Queueing systems used in safety-critical domains, such as medicine and transportation.

5.2 Future Work

Some of the worth mentioning extensions of our formalization are outlined as follows:

- The formalization of Poisson process along with the formalization of some continuous random variables (such as Normal random variable) can be used to extend the reliability analysis framework available in HOL4 theorem prover [3].

- The extensions of the formalization of Continuous-Time Markov Chains and the M/M/1 queue can be used to formalize a variety of queueing models such as M/M/c (or also called ErlangC model) [34]. The M/M/c queue is a multi-server queueing model thus it is a generalisation of the M/M/1 queue which considers only a single server. Further other queues with infinite number of servers can be also formalized.

- Another interesting direction is to formalize the Semi-Markov Decision Process (SMDP) [49], which is widely used for the performance analysis of software and distributed systems. It is also used in modeling stochastic control problems in Markovian dynamic systems where the sojourn time in exponentially distributed. Our formalization of Continuous-Time Markov Chains can also be further extended to formalize continuous-time Markov Decision Processes (MDP) [36], which can be applied in the analysis of Queueing systems and epidemic processes.

Bibliography

[1] *VESTA: A Statistical Model-Checker and Analyzer for Probabilistic Systems*, 2005.

[2] Learning and Designing Stochastic Processes from Logical Constraints. In *International Conference on Quantitative Evaluation of Systems*, volume 8054 of *LNCS*, pages 89–105, 2013.

[3] N. Abbasi. *Formal Reliability Analysis using Higher-Order Logic Theorem Proving*. PhD thesis, Concordia University, Montreal, QC, Canada, 2012.

[4] R. Affeldt and M. Hagiwara. Formalization of Shannon's Theorems in Ssreflect-coq. In *Interactive Theorem Proving*, volume 7406 of LNCS, pages 233–249. Springer, 2012.

[5] B. Akbarpour and L. C. Paulson. Metitarski: An Automatic Prover for the Elementary Functions. In *AISC/MKM/Calculemus*, volume 5144 of *LNCS*. Springer, 2008.

[6] A.O. Allen. *Probability, Statistics, and Queueing Theory: With Computer Science Applications*. Computer science and scientific computing. Academic Press, 1990.

[7] M. E. AMSI, P. Brown, D. Hunt, and D. Mathews. *The Binomial Theorem*. Education Services Australia, 2013.

[8] C. Baier and J. Katoen. *Principles of Model Checking*. MIT Press, 2008.

[9] C. Baier, J. Katoen, and H. Hermanns. Approximate Symbolic Model Checking of Continuous-Time Markov Chains. In *CONCUR*, volume 1664 of *LNCS*, pages 146–161. Springer, 1999.

[10] C. Baier and J. P. Katoen. *Principles of Model Checking*. The MIT Press, 2008.

[11] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. M. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van Der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics, 1994.

[12] M. Bernardo and R. Gorrieri. Extended Markovian Process Algebra. In *Concurrency Theory*, volume 1119 of *LNCS*, pages 315–330. Springer, 1996.

[13] D. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall, Upper Saddle River, NJ, USA, 2nd edition, 1992.

[14] U. N. Bhat. *An Introduction to Queueing Theory: Modeling and Analysis in Applications*. Statistics for Industry and Technology. Springer, 2008.

[15] R. N. Bhattacharya and E. C. Waymire. *Stochastic Processes with Applications*. John Wiley & Sons, 1990.

[16] J. Bialas. The σ-Additive Measure Theory. *Journal of Formalized Mathematics*, 2(1), 1990.

[17] C. E. Brown. *Automated Reasoning in Higher-order Logic*. College Publications, 2007.

[18] A. P. Bruce. *An Introduction to Mathematical Logic and Type Theory: to Truth Through Proof*. Computer Science and Applied Mathematics. Academic Press, 1986.

[19] E. Çinlar. *Introduction to Stochastic Processes*. Prentice-Hall, 1975.

[20] H. H. Chen, E. Thurfjell, S. W. Duffy, and L. Tabar. Evaluation by Markov Chain Models of a Non-randomised Breast Cancer Screening Programme. *Journal of Epidemiol Community Health*, 52(5):329–335, 1998.

[21] A. Church. A Formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[22] C. Clapham and J. Nicholson. *The Concise Oxford Dictionary of Mathematics*. Oxford University Press, 2009.

[23] A. Coble. *Anonymity, Information, and Machine-Assisted Proof*. PhD thesis, University of Cambridge, UK, 2009.

[24] H. Constantine. Markov Chains and Queuing Theory. REU Papers, The University of Chicago, 2011.

[25] S. Conversy, S. Chatty, S. Haspard-Boulinc, and J. L. Vinot. The Accident of Flight AF447 Rio-Paris: A Case Study for HCI Research. In *Conference on Human-Machine Interaction*, pages 60–69. ACM, 2014.

[26] R. B. Cooper. *Introduction to Queuing Theory*. Elsevier North-Holland, 1981.

[27] Coq. `http://coq.inria.fr/`, 2015.

[28] J. W. Daniel. Poisson Processes. `http://www.actuarialseminars.com`, 2008.

[29] The HOL System Description. `http://hol.sourceforge.net`, 2011.

[30] M. Dowson. The Ariane 5 Software Failure. *SIGSOFT Software Engineering Notes*, 22(2):84, 1997.

[31] M. Elleuch, O. Hasan, S. Tahar, and M. Abid. Formal Probabilistic Analysis of Detection Properties in Wireless Sensor Networks. *Formal Aspects of Computing*, 27(1):79–102, 2014.

[32] A. Erlang. The Theory of Probabilities and Telephone Conversations. *Nyt Tidsskrift for Matematik*, 20:33–39, 1909.

[33] M. Fitting. *First-order Logic and Automated Theorem Proving*. Springer-Verlag, 1996.

[34] N. Gautam. *Analysis of Queues: Methods and Applications*. CRC Press, 2012.

[35] M. J. C. Gordon and T. F. Melham, editors. *Introduction to HOL: A Theorem Proving Environment for Higher-Order Logic*. Cambridge University Press, 1993.

[36] X. Guo and O. Hernández-Lerma. Continuous-time markov decision processes: Theory and applications. In *Stochastic Modelling and Applied Probability*. Springer, 2009.

[37] A. Gupta. Formal Hardware Verification Methods: A Survey. *Formal Methods in System Design*, 1(2-3):151–238, 1992.

[38] J. Harrison. Formalized Mathematics. Technical Report 36, Turku Centre for Computer Science, Finland, 1996.

[39] J. Harrison. *Theorem Proving with the Real Numbers*. PhD thesis, Cambridge University, UK, 1998.

[40] J. Harrison. *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press, 2009.

[41] J. Harrison. A List of Theorem Provers. `http://www.cl.cam.ac.uk/users/jrh/ar.html`, 2011.

[42] O. Hasan. *Formal Probabilistic Analysis using Theorem Proving*. PhD thesis, Concordia University, Montreal, QC, Canada, 2008.

[43] O. Hasan, N. Abbasi, B. Akbarpour, S. Tahar, and R. Akbarpour. Formal Reasoning about Expectation Properties for Continuous Random Variables. In *Formal Methods*, volume 5850, pages 435–450. Springer, 2009.

[44] O. Hasan and S. Tahar. Reasoning about Conditional Probabilities in a Higher-Order-Logic Theorem Prover. *Journal of Applied Logic*, 9(1):23–40, 2011.

[45] H. Hermanns, U. Herzog, U. Klehmet, V. Mertsiotakis, and M. Siegle. Compositional Performance Modelling with the TIPPtool. *Performance Evaluation*, 39(1-4):5–35, 2000.

[46] H. Hermanns, J. Katoen, J. Pieter, J. Meyer, and M. Siegle. A Markov Chain Model Checker. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 1785 of *LNCS*, pages 347–362. Springer, 2000.

[47] J. Holzl and A. Heller. Three Chapters of Measure Theory in Isabelle/HOL. In *Interactive Theorem Proving*, volume 6898 of LNCS, pages 135–151. Springer, 2011.

[48] J. Hölzl and T. Nipkow. Verifying pCTL Model Checking. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 7214 of LNCS, pages 347–361. Springer, 2012.

[49] Q. Hu and W. Yue. Semi-Markov Decision Processes. In *Markov Decision Processes With Their Applications*, volume 14 of *Advances in Mechanics and Mathematics*, pages 105–120. Springer, 2008.

[50] J. Hurd. *Formal Verification of Probabilistic Algorithms*. PhD thesis, University of Cambridge, UK, 2002.

[51] F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer, 2007.

[52] S. K. Jha, E. M. Clarke, C. J. Langmead, A. Legay, A. Platzer, and P. Zuliani. A Bayesian Approach to Model Checking Biological Systems. In *Computational*

Methods in Systems Biology, volume 5688 of *LNCS*, pages 218–234. Springer, 2009.

[53] W. J. Knottenbelt, N. J. Dingle, and P. G. Harrison. HYDRA Hypergraph based distributed Response-time Analyser. In *International Conference on Parallel and Distributed Processing Technique and Applications*, pages 215–219, 2003.

[54] A. N. Kolmogorov. *Foundations of Probability*. 1933. Second English Edition, *Foundations of Probability* 1950, Chelsea Publishing Co.

[55] V. G. Kulkarni. *Modeling, Analysis, Design, and Control of Stochastic Systems*. Springer Text in Statistics. Springer, 1999.

[56] M. Kwiatkowska, G. Norman, and D. Parker. Controller Dependability Analysis by Probabilistic Model Checking. *Control Engineering Practice*, 15(11):1427–1434, 2006.

[57] M. Kwiatkowska, G. Norman, D. Parker, and M.G. Vigliotti. Probabilistic Mobile Ambients. *Theoretical Computer Science*, 410(12–13):1272–1303, 2009.

[58] M. Kwiatkowska and C. Thachuk. Probabilistic Model Checking for Biology. In *Software Safety and Security*, NATO Science for Peace and Security Series - D: Information and Communication Security. IOS Press, 2014.

[59] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. Providence, R.I. American Mathematical Society, 2006.

[60] L. Liu. *Formalization of Discrete-time Markov Chains in HOL*. PhD thesis, Concordia University, Montreal, QC, Canada, 2013.

[61] L. Liu, O. Hasan, V. Aravantinos, and S. Tahar. Formal Reasoning about Classified Markov Chains in HOL. In *Interactive Theorem Proving(ITP)*, volume 7998, pages 295–310. Springer, 2013.

[62] L. Liu, O. Hasan, and S. Tahar. Formalization of Finite-State Discrete-Time Markov Chains in HOL. In *Automated Technology for Verification and Analysis*, volume 6996 of *LNCS*, pages 90–104. Springer, 2011.

[63] L. Liu, O. Hasan, and S. Tahar. Formal Reasoning About Finite-State Discrete-Time Markov Chains in HOL. *Journal of Computer Science and Technology*, 28(2):217–231, 2013.

[64] D. J. C. MacKay. Introduction to Monte Carlo Methods. In *Learning in Graphical Models*, NATO Science Series, pages 175–204. Kluwer Academic Press, 1998.

[65] Maplesoft. `http://www.maplesoft.com/`, 2015.

[66] Mathematica. `http://www.wolfram.com`, 2015.

[67] T. Mhamdi. *Information-Theoretic Analysis using Theorem Proving*. PhD thesis, Concordia University, Montreal, QC, Canada, 2012.

[68] T. Mhamdi, O. Hasan, and S. Tahar. Formalization of Entropy Measures in HOL. In *Interactive Theorem Proving*, volume 6898 of *LNCS*, pages 233–248. Springer, 2011.

[69] T. Mhamdi, O. Hasan, and S. Tahar. Formalization of Measure Theory and Lebesgue Integration for Probabilistic Analysis in HOL. *ACM Transactions on Transactions on Embedded Computing Systems*, 12(1):13, 2013.

[70] S. Miller and D. Childers. *Probability and Random Processes: With Applications to Signal Processing and Communications*. Academic Press, 2012.

[71] R. Milner. A Theory of Type Polymorphism in Programming. *Journal of Computer and System Sciences*, 17:348–375, 1978.

[72] R. Milner, M. Tofte, and D. Macqueen. *The Definition of Standard ML*. MIT Press, Cambridge, MA, USA, 1997.

[73] Mobius. http://www.mobius.illinois.edu/, 2015.

[74] A. Nedzusiak. σ-fields and Probability. *Journal of Formalized Mathematics*, 1:1–6, 1989.

[75] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

[76] I. Niven, H.S. Zuckerman, and H.L. Montgomery. *An Introduction to the Theory of Numbers*. John Wiley & Sons, 1991.

[77] G. Padma and C. Vijayalakshmi. An Analysis of Continuous Time Markov Chains using Generator Matrices. *International Journal of Computer Applications*, 35(10):20–24, 2011.

[78] A. Papoulis. *Probability, (R)andom Variables, and Stochastic Processes*. McGraw Hill, 1984.

[79] L. C. Paulson. *ML for the Working Programmer*. Cambridge University Press, 1996.

[80] PEPA. http://www.dcs.ed.ac.uk/pepa/, 2015.

[81] A. Di Pierro, C. Hankin, and H. Wiklicky. Continuous-Time Probabilistic KLAIM. *Electronic Notes in Theoretical Computer Science*, 128(5):27–38, 2005.

[82] J. H. Pollard. *A Handbook of Numerical and Statistical Techniques*. Cambridge University Press, 1977.

[83] PRISM. http://www.prismmodelchecker.org, 2015.

[84] S. I. Resnick. *Adventures in Stochastic Processes*. Birkhuser, 1992.

[85] C. Rose and M. D. Smith. *Mathematical Statistics with Mathematica*. Mathematical Statistics with Mathematica. Springer, 2002.

[86] C. Rose and M.D. Smith. Symbolic Maximum Likelihood Estimation with Mathematica. *Journal of the Royal Statistical Society, Series D: The Statistician*, 49:229–240, 2000.

[87] M. Sczittnick. Macom - a Tool for Evaluating Communication Systems. In *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation)*, pages 7–10, 1994.

[88] R. Serfozo. *Basics of Applied Stochastic Processes*. Springer, 2009.

[89] SHARPE. `http://sharpe.pratt.duke.edu/`, 2015.

[90] U. Siddique. Formal Analysis of Fractional Order Systems in Higher-order Logic. Master's thesis, National University of Sciences and Technology, Pakistan, 2011.

[91] K. Slind and M. Norrish. A brief overview of HOL4. In *Theorem Proving in Higher Order Logics*, volume 5170 of *LNCS*, pages 28–32. Springer, 2008.

[92] M. Sommereder. *Modelling of Queueing Systems with Markov Chains: An Introduction to Basic and Advanced Modelling Techniques*. Books on Demand, 2011.

[93] L. Song, L. Zhang, and J. C. Godskesen. Bisimulations and Logical Characterizations on Continuous-Time Markov Decision Processes. In *Verification, Model Checking, and Abstract Interpretation*, volume 8318 of *LNCS*, pages 98–117. Springer, 2014.

[94] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Information-theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[95] W. J. Stewart. Marca: Markov Chain Analyzer - A Software Package for Markov Modelling. IRISA Publication Interne No. 45, Universite de Rennes, France, 1996.

[96] H. E. Taylor and S. Karlin. *An Introduction to Stochastic Modeling*. Academic Press, 1998.

[97] W. Ward. Continuous-Time Markov Chain. *Annals OR*, 211:357–379, 2012.

[98] Roy D. Yates and David J. Goodman. *Probability and Stochastic Processes: a Friendly Introduction for Electrical and Computer Engineers*. John Wiley & Sons, 2005.

[99] YMER. `http://www.tempastic.org/ymer/`, 2015.

[100] Y. Zhang. Markov Chain Monte Carlo (MCMC) Simulations. In *Encyclopedia of Systems Biology*, pages 1176–1176. Springer, 2013.

[101] D. Zwillinger. *CRC Standard Mathematical Tables and Formulae, 32nd Edition*. Discrete Mathematics and Its Applications. CRC Press, Boca Raton, FL, USA, 2011.