Probabilistic Analysis using Theorem Proving

Osman Hasan

Dept. of Electrical & Computer Engineering, Concordia University
1455 de Maisonneuve W., Montreal, Quebec, H3G 1M8, Canada
o_hasan@ece.concordia.ca

Abstract. Traditionally, computer simulation techniques are used to perform probabilistic analysis. However, they provide less accurate results and cannot handle large-scale problems due to their enormous CPU time requirements. Recently, a significant amount of formalization has been done in the HOL theorem prover that allows us to conduct precise probabilistic analysis using theorem proving and thus overcome the limitations of the simulation based probabilistic analysis approach. Some major contributions include the formalization of both discrete and continuous random variables and the verification of some of their corresponding probabilistic and statistical properties. This paper presents a concise description of the infrastructures behind these capabilities and the utilization of these features to conduct the probabilistic analysis of real-world systems. For illustration purposes, the paper describes the theorem proving based probabilistic analysis of three examples, i.e., the roundoff error of a digital processor, the Coupon Collector's problem and the Stop-and-Wait protocol.

1 Introduction

Probabilistic analysis is a tool of fundamental importance for the analysis of hardware and software systems. These systems usually exhibit some random or unpredictable elements. Examples include, failures due to environmental conditions or aging phenomena in hardware components and the execution of certain actions based on a probabilistic choice in randomized algorithms. Moreover, these systems act upon and within complex environments that themselves have certain elements of unpredictability, such as noise effects in hardware components and the unpredictable traffic pattern in the case of telecommunication protocols. Due to these random components, establishing the correctness of a system under all circumstances usually becomes impractically expensive. The engineering approach to analyze a system with these kind of unavoidable elements of randomness and uncertainty is to use probabilistic analysis. The main idea behind probabilistic analysis is to mathematically model the random and unpredictable elements of the given system and its environment by appropriate random variables. The probabilistic properties of these random variables are then used to judge system's behavior regarding parameters of interest, such as, downtime, availability, number of failures, capacity, and cost. Thus, instead of guaranteeing that the system meets some given specification under all circumstances, the probability that the system meets this specification is reported.

Even for hardware and software systems for which correctness may be unconditionally guaranteed, the study of system performance primarily relies on probabilistic analysis. In fact, the term *system performance* commonly refers to the average time required by a system to perform a given task, such as the average runtime of a computational algorithm or the average message delay of a telecommunication protocol. These averages can be computed, based on the probabilistic analysis approach, by using appropriate random variables to model inputs for the system model.

Today, simulation is the most commonly used computer based probabilistic analysis technique. Most simulation softwares provide a programming environment for defining functions that approximate random variables for probability distributions. The random elements in a given system are modeled by these functions and the system is analyzed using computer simulation techniques [6], such as the Monte Carlo Method [20], where the main idea is to approximately answer a query on a probability distribution by analyzing a large number of samples. Statistical quantities, such as average and variance, may then be calculated, based on the data collected during the sampling process, using their mathematical relations in a computer. Due to the inherent nature of simulation, the probabilistic analysis results attained by this technique can never be

termed as 100% accurate. The precision and accuracy of the hardware and software system analysis results has become imperative these days because of the extensive usage of these systems in safety and financial critical areas, such as, medicine, transportation and stock exchange markets. Therefore, simulation cannot be relied upon for the analysis of such systems.

Formal methods [9] are capable of conducting precise system analysis and thus allow us to overcome the above mentioned limitations of the simulation approach. Probabilistic model checking [1, 24] is a rapidly emerging formal probabilistic analysis technique. Like traditional model checking [9], probabilistic model checking involves the construction of a precise state-based mathematical model of the given probabilistic system, which is then subjected to exhaustive analysis to verify if it satisfies a set of formally represented probabilistic properties. Numerous probabilistic model checking algorithms and methodologies have been proposed in the open literature, e.g., [5, 22], and based on these algorithms, a number of tools have been developed, e.g., PRISM [17] and VESTA [25]. Besides the accuracy of the results, the most promising feature of probabilistic model checking is the ability to perform the analysis automatically. On the other hand, it is limited to systems that can only be expressed as probabilistic finite state machines. Another major limitation of the probabilistic model checking approach is state space explosion [4]. Similarly, to the best of our knowledge, it has not been possible to precisely reason about statistical quantities, such as expectation and variance, using probabilistic model checking so far.

The second formal method that can be utilized to conduct probabilistic analysis is higher-order-logic theorem proving. Probabilistic analysis can be conducted within a higher-order-logic theorem prover by first modeling the behavior of the system that needs to be analyzed in higher-order logic, while expressing its random or unpredictable elements in terms of formalized random variables. The second step is to use this formal model to express the probabilistic and statistical properties, regarding the system, as higher-order logic theorems. For this purpose, we require higher-order-logic definitions of probabilistic and statistical properties of random variables, such as, *Probability Mass Function* (PMF), *Cumulative Distribution Function* (CDF), expectation and variance, etc. Finally, theorems corresponding to the probabilistic and statistical properties of the system model can be mechanically checked for correctness in a theorem prover.

The above mentioned theorem proving based probabilistic analysis approach tends to overcome the limitations of the simulation and model checking based probabilistic analysis approaches. Due to the formal nature of the models and properties and the inherent soundness of the theorem proving approach, probabilistic analysis carried out in this way 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, such as the state-space explosion problem in the case of probabilistic model checking, and formally verify analytically complex properties like expectation and variance.

The foremost criteria for implementing a theorem proving based probabilistic analysis framework is to be able to formalize and verify random variables in higher-order logic. Hurd's PhD thesis [16] can be considered a pioneering work in this regard as it presents a methodology for the formalization and verification of probabilistic algorithms in the higher-order-logic (HOL) theorem prover [8]. Random variables are basically probabilistic algorithms and thus can be formalized and verified, based on their probability distribution properties, using the methodology proposed in [16]. In fact, [16] presents the formalization of some discrete random variables along with their verification, based on the corresponding PMF properties. Building upon Hurd's formalization framework [16], we have been able to successfully verify the sampling algorithms of a few continuous random variables [13] based on their CDF properties as well. For comparison purposes, it is frequently desirable to summarize the characteristic of the distribution of a random variable by a single number, such as its expectation or variance, rather than an entire function. For example, it is more interesting to find out the expected value of the runtime of an algorithm for an NP-hard problem, rather than the PMF or CDF of the runtime. In [15, 12], we tackled the formalization of expectation and variance in HOL for the first time. We extended Hurd's formalization framework with a formal definition of expectation, which can be utilized to formalize and verify the expectation and variance characteristics associated with discrete random variables that attain values in positive integers only. The current paper provides a brief overview of the formalization of the above mentioned mathematical foundations. It also illustrates the usage of this available formalization for conducting probabilistic analysis in a theorem prover.

The rest of the paper is organized as follows: Section 2 describes a hypothetical theorem proving based probabilistic analysis framework. This description illustrates how the already formalized mathematical concepts of probability theory fit into the global picture of probabilistic analysis while highlighting some of the interesting future research directions in the area of theorem proving based probabilistic analysis. Then in Sections 3 to 5, we briefly describe the already developed HOL infrastructures for the formalization of discrete random variables [16], the formalization of continuous random variables [13] and the verification of statistical properties [15, 12], respectively. In order to illustrate the practical effectiveness of the proposed approach, we then present the probabilistic analysis of three examples using the HOL theorem prover in Section 6. The examples include the roundoff analysis of a digital processor, the probabilistic analysis of the Coupon Collector's problem, which is a commercially used algorithm inspired by "*Collect all n Coupons and win*" contests, and the performance analysis of the Stop-and-Wait protocol, which is a commonly used protocol that ensures reliable communication between computers. Finally, Section 7 concludes the paper.

2 Proposed Framework

A hypothetical model of a theorem proving based probabilistic analysis framework is given in Fig. 1, with some of its most fundamental components depicted with shaded boxes. Like all traditional analysis problems, the starting point of probabilistic analysis is also a system description and some intended system properties and the goal is to check if the given system satisfies these given properties. For simplicity, we have divided system properties into two categories, i.e., system properties related to discrete random variables and system properties related to continuous random variables.

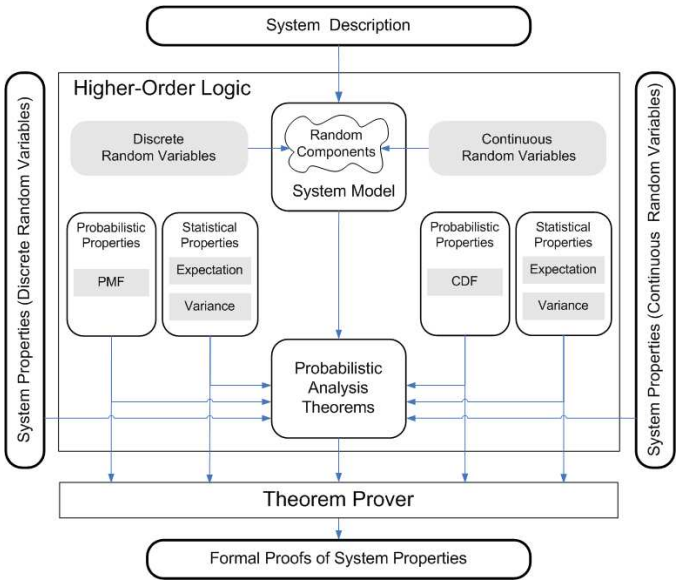

Fig. 1. Theorem Proving based Probabilistic Analysis Framework

The first step in conducting probabilistic analysis in a theorem prover is to construct a model of the given system in higher-order-logic. For this purpose, the foremost requirement is the availability of infrastructures that allow us to formalize all kinds of discrete and continuous random variables as higher-order-logic functions, which in turn can be used to represent the random components of the given system in its higher-order-logic model. The second step in theorem proving based probabilistic analysis is to utilize the formal model of the system to express system properties as higher-order-logic theorems. The prerequisite for this

step is the ability to express probabilistic and statistical properties related to both discrete and continuous random variables in higher-order-logic. All probabilistic properties of discrete and continuous random variables can be expressed in terms of their PMF and CDF functions, respectively. Similarly, most of the commonly used statistical properties can be expressed in terms of the expectation and variance characteristics of the corresponding random variable. Thus, we require the formalization of mathematical definitions of PMF, CDF, expectation and variance for both discrete and continuous random variables in order to be able to express the given system's probabilistic and statistical properties as higher-order-logic theorems. The third and the final step for conducting probabilistic analysis in a theorem prover is to formally verify the higher-order-logic theorems developed in the previous step using a theorem prover. For this verification, it would be quite handy to have access to a library of some pre-verified theorems corresponding to some commonly used properties regarding probability distribution functions, expectation and variance. Since, we can build upon such a library of theorems and thus speed up the verification process.

Most of the above mentioned formalization prerequisites of a theorem proving based probabilistic analysis framework have already been fulfilled in the HOL theorem prover, as has been outlined in the last section. These infrastructures and methodologies are briefly described in the next three sections of this paper. On the other hand, to the best of our knowledge, the formalization and verification of statistical properties, like expectation and variance, for continuous random variables is an open research issue as of now. This step requires a higher-order-logic formalization of an integration function that can also handle functions with domains other than real numbers. Lebesgue integration provides this feature and thus the higher-order-logic formalization of some portions of the Lebesgue integration theory [23] can be built upon for formalizing the mathematical concepts of expectation and variance for continuous random variables.

3 Formalization of Discrete Random Variables

A random variable is called discrete if its range, i.e., the set of values that it can attain, is finite or at most countably infinite [26]. Examples of discrete random variables include the outcome of rolling a dice and the number of children in a family. Discrete random variables can be completely characterized by their PMF that returns the probability that a random variable X is exactly equal to some value x, i.e., $Pr(X = x)$.

Random variables can be formalized in higher-order-logic as deterministic functions with access to an infinite Boolean sequence $\mathbb{B}^\infty$; source of an infinite random bits with data type $(num \rightarrow bool)$ [16]. These deterministic functions make random choices based on the result of popping the top most bit in the infinite Boolean sequence and may pop as many random bits as they need for their computation. When the functions terminate, they return the result along with the remaining portion of the infinite Boolean sequence to be used by other functions. Thus, a random variable that takes a parameter of type α and ranges over values of type β can be represented in HOL by the function

$$\mathcal{F} : \alpha \rightarrow B^\infty \rightarrow \beta \times B^\infty$$

For example, a $Bernoulli(\frac{1}{2})$ random variable that returns 1 or 0 with equal probability $\frac{1}{2}$ can be modeled as follows

```
⊢ bit = λs. (if shd s then 1 else 0, stl s)
```

where the variable s, in the above definition, represents the infinite Boolean sequence and the functions `shd` and `stl` are the sequence equivalents of the list operation *'head'* and *'tail'*. The function `bit` accepts the infinite Boolean sequence and returns a pair with the first element equal to either 0 or 1 and the second element equal to the unused portion of the infinite Boolean sequence, which in this case is the tail of the sequence.

The work in [16] also presents the formalization of some mathematical measure theory in HOL, which can be used to define a probability function $\mathbb{P}$ from sets of infinite Boolean sequences to *real* numbers between 0 and 1. The domain of $\mathbb{P}$ is the set $\mathcal{E}$ of events of the probability. Both $\mathbb{P}$ and $\mathcal{E}$ are defined using the Carathéodory's Extension theorem, which ensures that $\mathcal{E}$ is a σ-algebra: closed under complements and

countable unions. The formalized $\mathbb{P}$ and $\mathcal{E}$ can be used to prove probabilistic properties for random variables such as

$$\vdash \mathbb{P}\ \{\texttt{s | fst (bit s) = 1}\} = \tfrac{1}{2}$$

where the function `fst` selects the first component of a pair and $\{x|C(x)\}$ represents a set of all elements x that satisfy the condition C in HOL.

The above mentioned infrastructure can be utilized to formalize most of the commonly used discrete random variables and verify their corresponding PMF relations. For example, HOL formalization and verification of Bernoulli and Uniform random variables can be found in [16] and of Binomial and Geometric random variables can be found in [11].

4 Formalization of Continuous Random Variables

A random variable is called continuous if it ranges over a continuous set of numbers [26]. A continuous set of numbers, sometimes referred to as an interval, contains all real numbers between two limits. Many experiments lead to random variables with a range that is a continuous interval. Examples include measuring T, the arrival time of a data packet at a web server ($S_T = \{t|0 \leq t < \infty\}$) and measuring V, the voltage across a resistor ($S_V = \{v| -\infty < v < \infty\}$), where T and V are both continuous random variables.

The sampling algorithms for discrete random variables are either guaranteed to terminate or satisfy probabilistic termination, meaning that the probability that the algorithm terminates is 1. On the other hand, the formalization of continuous random variables involves non-terminating algorithms and hence require a different approach than discrete random variables.

The work in [13] presents a methodology for the formalization of continuous random variables using the HOL theorem prover. The methodology builds upon Hurd's verification framework [16], described in the last section, and is primarily based on the concept of the nonuniform random number generation [6], which is the process of obtaining random variates of arbitrary distributions using a Standard Uniform random number generator. The main advantage of this approach is that we only need to formalize one continuous random variable from scratch, i.e., the Standard Uniform random variable, which can be used to model other continuous random variables by formalizing the corresponding nonuniform random number generation method.

Based on the above methodology, [13] presents a framework, illustrated in Fig. 2, for the formalization of all continuous random variables for which the inverse of the CDF can be represented in a closed mathematical form.

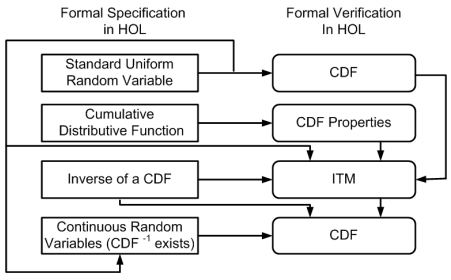

Fig. 2. Framework for the Formalization of Continuous Random Variables

The first step in this framework is the formal specification of the Standard Uniform random variable and the formal verification of this definition by proving the corresponding CDF property in the HOL theorem

prover. Standard Uniform random variable is a continuous random variable for which the probability that it will belong to a subinterval of [0,1] is proportional to the length of that subinterval. It is a well known mathematical fact, see [7] for example, that a Standard Uniform random variate can be modeled by an infinite sequence of random bits (informally coin flips) as follows

$$\sum_{k=0}^{\infty}(\frac{1}{2})^{k+1}X_k \tag{1}$$

where X_k denotes the outcome of the k^{th} random bit; $True$ or $False$ represented as 1 or 0 respectively. The mathematical relation of Equation (1) presents a sampling algorithm for the Standard Uniform random variable which is quite consistent with formalization methodology, described in the last section, i.e, it allows us to model the Standard Uniform random variable by a deterministic function with access to the infinite Boolean sequence. The specification of this sampling algorithm in higher-order logic is not very straight forward though. Due to the infinite sampling, it cannot be modeled by either of the approaches proposed in [16], i.e., a recursive function or the *probabilistic while loop*. A formalization approach for the Standard Uniform random variable has been presented in [14]. The main idea is to split the mathematical expression of (1) into two steps. The first step is to mathematically represent a discrete version of the Standard Uniform random variable.

$$(\lambda n.\sum_{k=0}^{n-1}(\frac{1}{2})^{k+1}X_k) \tag{2}$$

This lambda abstraction function accepts a positive integer n and generates an n-bit Standard Uniform random variable using the computation principle of Equation (1). The continuous Standard Uniform random variable is then represented as a special case of Equation (2) when n tends to infinity

$$\lim_{n\to\infty}(\lambda n.\sum_{k=0}^{n-1}(\frac{1}{2})^{k+1}X_k) \tag{3}$$

The advantage of expressing the sampling algorithm of Equation (1) in these two steps is that now it can be specified in HOL. The mathematical relationship of Equation (2) can be specified in HOL by a recursive function using the methodology for the formalization of discrete random variables, described in the last section, as it consumes a finite number of random bits, i.e., n. Then, the formalization of the mathematical concept of limit of a *real* sequence [10] in HOL can be used to specify the mathematical relation of Equation (3). The work in [14] also presents the correctness verification of this definition of the Standard Uniform random variable by proving its corresponding CDF relation in HOL.

The second step in the framework for the formalization of continuous probability distributions, given in Fig. 2, is the formalization of the CDF and the verification of its classical properties in HOL. CDF is defined as

$$F_X(x) = Pr(X \leq x) \tag{4}$$

for any number x, where Pr represents the probability function. A unique characteristic of CDF is that it can be used to describe the probability distribution of both discrete and continuous random variables. CDF and its properties have been an integral part of the classical probability theory since its early development in the 1930s and play a vital role in characterizing probabilistic properties of random variables. The work in [13] presents a higher-order-logic definition of CDF, based on Equation (4), and utilizes this definition to formally verify the CDF properties, given in Table 1, in the HOL theorem prover.

The next step in the framework for the formalization of continuous probability distributions, given in Fig. 2, is the formal verification of the Inverse Transform Method (ITM) [6], which is a well known nonuniform random generation technique for generating nonuniform random variates for continuous probability distributions for which the inverse of the CDF can be represented in a closed mathematical form. According to the ITM, for any continuous CDF F, the random variable X defined by $X = F^{-1}(U)$ has CDF F, where

No.	Property	Mathematical Representation
1	CDF Bounds	$0 \leq F_X(x) \leq 1$
2	Monotonically Increasing	$a < b \Rightarrow F_X(a) \leq F_X(b)$
3	Interval Probability	$a < b \Rightarrow Pr(a < X \leq b) = F_X(b) - F_X(a)$
4	CDF at Negative Infinity	$F_X(-\infty) = 0$
5	CDF at Positive Infinity	$F_X(\infty) = 1$
6	CDF is Continuous from the Right	$\lim_{x \to a^+} F_X(x) = F_X(a)$
7	CDF Limit from the Left	$\lim_{x \to a^-} F_X(x) = Pr(X < a)$

Table 1. CDF Properties

$F^{-1}(U)$ is defined to be the value of x such that $F(x) = U$ and U represents the Standard Uniform random variable. Mathematically, the ITM can be expressed as

$$Pr(F^{-1}(U) \leq x) = F(x) \tag{5}$$

In order to verify the above proposition in HOL, the work in [13] presents a formalization of the inverse function of a CDF as a higher-order-logic predicate that accepts two functions, f and g, of type $(real \rightarrow real)$ and returns $True$ if and only if the function f is the inverse of the CDF g according to the ITM proposition. Using this predicate, along with the formal definition of the Standard Uniform random variable, the ITM proposition, given in Equation (5), is verified in HOL. The proof is based on the CDF characteristic of the Standard Uniform random variable and some of the CDF properties, given in Table 1.

At this point, the formalized Standard Uniform random variable can be used to formally specify any continuous random variable for which the inverse of the CDF can be expressed in a closed mathematical form as $X = F^{-1}(U)$. Whereas, the CDF of this formally specified continuous random variable, X, can be verified, based on simple arithmetic reasoning, using the formal proof of the ITM. For illustration purposes, the work in [13] presents the formal specification of four continuous random variables; Exponential, Uniform, Rayleigh and Triangular. The correctness of these random variables is also verified in [13] by proving their corresponding CDF properties in the HOL theorem prover.

5 Verification of Statistical Properties

In probabilistic analysis, statistical characteristics, like expectation, variance and tail distribution bounds, play a major role in decision making as they tend to summarize the probability distribution characteristics of a random variable in a single number. Due to their widespread interest, the computation of statistical characteristics has now become one of the core components of every modern probabilistic analysis framework. In this section, we introduce the formalization infrastructure, initially proposed in [11] and presented in Fig. 3, that allows us to formally reason about expectation, variance, and tail distribution properties regarding discrete random variables that attain values in positive integers only.

The first step in the infrastructure, given in Fig. 3, is the formalization of an expression for expectation in higher-order logic. Expectation basically provides the average of a random variable, where each of the possible outcomes of this random variable is weighted according to its probability [2]. The expectation for a function of a discrete random variable, which attains values in the positive integers only, is defined as follows [19].

$$Ex\_fn[f(X)] = \sum_{n=0}^{\infty} f(n) Pr(X = n) \tag{6}$$

where X is the discrete random variable and f represents a function of the random variable X. The above definition only holds if the associated summation is convergent, i.e., $\sum_{n=0}^{\infty} f(n) Pr(X = n) < \infty$. The expression of expectation, given in Equation (6), has been formalized in [11] as a higher-order-logic function

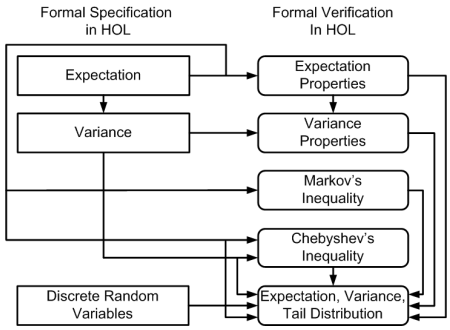

Fig. 3. Formalization Infrastructure for Reasoning about Statistical Properties

using the formalization of the probability function $\mathbb{P}$, explained in Section 3 of this paper. The expected value of a discrete random variable that attains values in positive integers can now be defined as a special case of Equation (6)

$$Ex[X] = Ex\_fn[(\lambda n.n)(X)] \tag{7}$$

when f is an identity function. In order to verify the correctness of the above definitions of expectation, they are utilized in [11] to formally verify the following classical expectation properties using the HOL theorem prover.

$$Ex[\sum_{i=1}^{n} R_i] = \sum_{i=1}^{n} Ex[R_i] \tag{8}$$

$$Ex[a + bR] = a + bEx[R] \tag{9}$$

These properties not only verify the correctness of the above definitions but also play a vital role in verifying the expectation characteristics of discrete random components of probabilistic systems, as will be seen in Section 6 of this paper.

The second step in the framework for the formal verification of statistical characteristics, given in Fig. 3, is the formalization of variance and the verification of its properties in the HOL theorem prover. Variance of a random variable X describes the difference between X and its expected value and thus is a measure of its dispersion. It is defined for a discrete random variable, X, as follows

$$Var[X] = Ex[(X - Ex[X])^2] \tag{10}$$

The above definition of variance has been formalized in higher-order-logic in [11] by utilizing the formal definitions of expectation, given in Equations (6) and (7). This definition is then formally verified to be correct in the HOL theorem prover by proving the following classical variance properties for it.

$$Var[R] = Ex[R^2] - (Ex[R])^2 \tag{11}$$

$$Var[\sum_{i=1}^{n} R_i] = \sum_{i=1}^{n} Var[R_i] \tag{12}$$

Based on the expectation and variance characteristics of a random variable, we can find bounds for the tail distribution, i.e., the probability that a random variable assumes values that are far from its expectation. These bounds are usually calculated using the Markov's or the Chebyshev's inequalities [2]. The Markov's

inequality gives an upper bound for the probability that a non-negative random variable X is greater than or equal to some positive constant.

$$Pr(X \geq a) \leq \frac{Ex[X]}{a} \tag{13}$$

Markov's inequality gives the best tail bound possible, for a nonnegative random variable, using the expectation for the random variable only [21]. This bound can be improved upon if more information about the distribution of the random variable is taken into account. Chebyshev's inequality is based on this principle and it presents a significantly stronger tail bound in terms of variance.

$$Pr(|X - Ex[X]| \geq a) \leq \frac{Var[X]}{a^2} \tag{14}$$

The third and the fourth steps in the framework for the formal verification of statistical characteristics, given in Fig. 3, are the formal verification of Markov's and Chebyshev's inequalities in the HOL the prover, respectively. The verification is based on the formal definitions of expectation and variance and their formally verified properties and is outlined in [12].

The above mentioned formalization and verification allows us to reason about expectation, variance and tail distribution properties of any formalized discrete random variable that attains values in positive integers. For illustration purposes, [12] presents the formal verification of the expectation and variance relations for four discrete random variables: Bernoulli, Uniform Binomial and Geometric.

6 Applications

In this section, we illustrate the usage of the formalization, mentioned so far in this paper, for conducting probabilistic analysis. For this purpose, we present the formal probabilistic analysis of three examples using the HOL theorem prover.

6.1 Probabilistic Analysis of Roundoff Error in a Digital Processor

Consider the roundoff error for a particular digital processor to be uniformly distributed over the interval [-5x10^{-12}, 5x10^{-12}]. Our goal is to formally verify that the probability of the event when the roundoff error in this digital processor is greater than 2x10^{-12} is less than 0.33 and the probability that the final result fluctuates by $\pm$1x10^{-12} with respect to the actual value is precisely equal to 0.2.

The first step for formally analyzing the above mentioned properties is to model the randomness with an appropriate random variable in higher-order-logic. The continuous Uniform random variable, formalized using the infrastructure explained in Section 4, can be used for this purpose. The generalized function for the Uniform random variable can be specialized for the interval [-5x10^{-12}, 5x10^{-12}] and the given properties can be expressed and verified as higher-order-logic theorems as follows

$\vdash \mathbb{P}$ {s | 2x10^{-12} < uniform_rv -5x10^{-12} 5x10^{-12} s } < 0.33

$\vdash \mathbb{P}$ {s | (-1x10^{-12} < uniform_rv -5x10^{-12} 5x10^{-12} s) $\wedge$
(uniform_rv -5x10^{-12} 5x10^{-12} s $\leq$ 1x10^{-12})} = 0.2

where uniform_rv represents the higher-order-logic function corresponding to the Uniform random variable. The proofs are based on CDF properties, given in Table 1, and some basic probability laws, verified in [16].

The above example illustrates the usefulness of formalized continuous random variables in verifying probabilistic quantities with 100% precision.

6.2 Probabilistic Analysis of the Coupon Collector's Problem

The Coupon Collector's problem [21] refers to the problem of probabilistically evaluating the number of trials required to acquire all unique, say n, coupons from a collection of multiple copies of these coupons that are independently and uniformly distributed. The problem is similar to the example when each box of cereal contains one of n different coupons and once you obtain one of every type of coupon, you win a prize.

Our first goal is to verify, using HOL, that the expected value of acquiring all n coupons is $nH(n)$, where $H(n)$ is the *harmonic number* ($\sum_{i=1}^{n} 1/i$). Based on this expectation value, we then reason about the tail distribution properties of the Coupon Collector's problem using the formally verified Markov's and Chebyshev's inequalities.

The Coupon Collector's problem can be formalized by modeling the total number of trials required to obtain all n unique coupons, say T, as a sum of the number of trials required to obtain each distinct coupon, i.e., $T = \sum_{i=1}^{n} T_i$, where T_i represents the number of trials to obtain the i^{th} coupon, while $i-1$ distinct coupons have already been acquired. The advantage of breaking the random variable T into the sum of n random variables $T_1, T_2 \cdots, T_n$ is that each T_i can be modeled by the Geometric random variable function. It is important to note here that the probability of success for these Geometric random variables would be different from one another and would be equal to the probability of finding a new coupon while conducting uniform selection trials on the available n coupons. Thus, the success probability depends on the number of already acquired coupons and can be modeled using the higher-order-logic function for the discrete Uniform random variable. Based on this methodology, [15] models the Coupon Collector's problem as a higher-order-logic function, `coupon_collector`, that accepts a positive integer greater than 0, $n+1$, which represents the total number of distinct coupons that are required to be collected. The function returns the number of trials for acquiring these $n+1$ distinct coupons and utilizes the formalized Geometric and Uniform random variables.

Now, using the formal definitions of expectation and variance and the formally verified corresponding properties, given in Section 5, the following properties can be proved in the HOL theorem prover for the above mentioned Coupon Collector function.

$$\vdash \forall\ \texttt{n. expec (coupon\_collector (n + 1)) = (n + 1) } \left(\sum_{\texttt{i=0}}^{\texttt{n+1}} \frac{1}{\texttt{i+1}}\right)$$

$$\vdash \forall\ \texttt{n a. 0 < a} \Rightarrow \mathbb{P}\ \{\texttt{s | (fst(coupon\_collector (n + 1) s)) } \geq \texttt{a}\} \leq \left(\frac{\texttt{(n+1)}}{\texttt{a}}\ \left(\sum_{\texttt{i=0}}^{\texttt{n+1}} \frac{1}{\texttt{(i+1)}}\right)\right)$$

$$\vdash \forall\ \texttt{n a. 0 < a} \Rightarrow \mathbb{P}\ \{\texttt{s | abs((fst(coupon\_collector (n + 1) s)) - expec (coupon\_collector (n + 1))) } \geq \texttt{a}\} \leq \left(\frac{\texttt{(n+1)}^2}{\texttt{a}^2}\ \left(\sum_{\texttt{i=0}}^{\texttt{n+1}} \frac{1}{\texttt{(i+1)}^2}\right)\right)$$

where `expec` and `abs` represent the HOL functions for expectation and absolute functions, respectively.

The first theorem gives the expectation of the Coupon Collector's problem, while the next two correspond to the tail distribution bounds of the Coupon Collector's problem using Markov and Chebyshev's inequalities, respectively. The above results exactly match the results of the analysis based on paper-and-pencil proof techniques [21] and are thus 100 % precise, which is a novelty that cannot be achieved, to the best of our knowledge, by any existing computer based probabilistic analysis tool.

6.3 Performance Analysis of the Stop-and-Wait Protocol

The Stop-and-Wait protocol [18] utilizes the principles of error detection and retransmission and is a fundamental mechanism for reliable communication between computers. The main idea is that the transmitter keeps on transmitting a data packet (repeating after every t_{out} units of time) unless and until it receives a valid acknowledgement (ACK) of its reception from the receiver. This section describes the formal verification of the Stop-and-Wait protocol's average message delay relation for the sake of performance analysis.

Stop-and-Wait protocol is a classical example of a real-time system and thus involves a subtle interaction of a number of distributed processes. The behavior of these processes over time may be specified by higher-order-logic predicates on positive integers [3] that represent the ticks of a clock counting physical time in any appropriate units, e.g., nanoseconds. The granularity of the clock's tick is believed to be chosen in such a way that it is sufficiently fine to detect properties of interest. Using this methodology, [11] presents a higher-order-logic formalization of the Stop-and-Wait protocol as a logical conjunction of six processes and some initial conditions, which are used to ensure the correct operation of the formal model. The random component in the Stop-and-Wait protocol is channel noise, which is expressed using the formal Bernoulli random variable function.

Now, the formal model of the Stop-and-Wait protocol is used to formally verify the following average message delay relation of the Stop-and-Wait protocol.

$$\frac{(t_f + t_{out})p}{1-p} + t_f + t_{prop} + t_{proc} + t_a + t_{prop} + t_{proc} \tag{15}$$

The variable p, in the above expression, represents the probability of channel error. Whereas, the variables t_f, t_a, t_{prop}, t_{proc} and t_{out} denote the time delays associated with data transmission, ACK transmission, message propagation, message processing and time-out delays, respectively. The verification is based on the formalization of expectation and the formally verified expectation properties. Further details about this verification can be found in [11].

It is important to note here that the result of Equation (15) is not new. In fact its existence dates back to the early days of introduction of the Stop-and-Wait protocol. However, it has always been verified using theoretical paper-and-pencil proof techniques, so far. Whereas, the analysis described in this paper is based on mechanical verification using the HOL theorem prover, which is a superior approach to both paper-and-pencil proofs and simulation based analysis techniques.

7 Conclusions

This paper provides a brief overview of the existing work in the HOL theorem prover related to probabilistic analysis. It also highlights the role of each one of these existing methodologies in the area of probabilistic analysis and while doing so presents a hypothetical model of a theorem proving based probabilistic analysis framework. The main idea behind this framework is to use random variables formalized in higher-order logic to model systems, which need to be analyzed, and to verify the corresponding probabilistic and statistical properties in a theorem prover. Because of the formal nature of the models, the analysis is free of approximation and precision errors and due to the high expressive nature of higher-order logic a wider range of systems can be analyzed. Thus, the theorem proving based probabilistic analysis approach can prove to be very useful for the performance and reliability optimization of safety critical and highly sensitive engineering and scientific applications.

We utilized the above mentioned mathematical foundations to present the formal probabilistic analysis of three examples, i.e., the Roundoff error in a digital processor, the Coupon Collectors Algorithm and the Stop-and-Wait protocol. The analysis results exactly matched the results obtained by paper-and-pencil proof techniques and are thus 100 % precise. The successful handling of these diverse probabilistic analysis problems by the proposed approach clearly demonstrates its feasibility for real-world probabilistic analysis issues.

The main limitation of the proposed approach is the associated significant user interaction, i.e., the user needs to guide the proof tools manually since we are dealing with higher-order logic, which is known to be non-decidable. On the other hand, simulation is capable of handling all sorts of probabilistic analysis problems in an automated way but the solutions provided are not exact. Whereas, probabilistic model is capable of providing exact answers for a subset of probabilistic analysis problems. We believe that all these three techniques have to play together in order to form a successful probabilistic analysis framework. For example, an efficient approach would be to use simulation for the less critical parts of the analysis, model checking for the critical parts that it can handle and theorem proving for the remaining critical parts.

Finally, it is important to note that the presented methodologies and frameworks are not specific to the HOL theorem prover and can be adapted to any other higher-order-logic theorem prover, such as Isabelle, Coq or PVS, as well.

References

1. C. Baier, B. Haverkort, H. Hermanns, and J.P. Katoen. Model Checking Algorithms for Continuous time Markov Chains. *IEEE Transactions on Software Engineering*, 29(4):524–541, 2003.
2. P. Billingsley. *Probability and Measure*. John Wiley, 1995.
3. R. Cardell-Oliver. *The Formal Verification of Hard Real-time Systems*. PhD Thesis, University of Cambridge, Cambridge, UK, 1992.
4. E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. The MIT Press, 2000.
5. L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD Thesis, Stanford University, Stanford, USA, 1997.
6. L. Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, 1986.
7. W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 2. Wiley, 1971.
8. M.J.C. Gordon and T.F. Melham. *Introduction to HOL: A Theorem Proving Environment for Higher-Order Logic*. Cambridge University Press, 1993.
9. A. Gupta. Formal Hardware Verification Methods: A Survey. *Formal Methods in System Design*, 1(2-3):151–238, 1992.
10. J. Harrison. *Theorem Proving with the Real Numbers*. Springer, 1998.
11. O. Hasan. *Formal Probabilistic Analysis using Theorem Proving*. PhD Thesis, Concordia University, Montreal, QC, Canada, 2008.
12. O. Hasan and S. Tahar. Formal Verification of Tail Distribution Bounds in the HOL Theorem Prover. *Mathematical Methods in the Applied Sciences*. In-print.
13. O. Hasan and S. Tahar. Formalization of the Continuous Probability Distributions. In *Automated Deduction*, volume 4603 of *LNAI*, pages 3–18. Springer, 2007.
14. O. Hasan and S. Tahar. Formalization of the Standard Uniform Random Variable. *Theoretical Computer Science*, 382(1):71–83, 2007.
15. O. Hasan and S. Tahar. Verification of Expectation Properties for Discrete Random Variables in HOL. In *Theorem Proving in Higher-Order Logics*, volume 4732 of *LNCS*, pages 119–134. Springer, 2007.
16. J. Hurd. *Formal Verification of Probabilistic Algorithms*. PhD Thesis, University of Cambridge, Cambridge, UK, 2002.
17. M. Kwiatkowska, G. Norman, and D. Parker. Quantitative Analysis with the Probabilistic Model Checker PRISM. *Electronic Notes in Theoretical Computer Science*, 153(2):5–31, 2005. Elsevier.
18. A. Leon Garcia and I. Widjaja. *Communication Networks: Fundamental Concepts and Key Architectures*. McGraw-Hill, 2004.
19. A. Levine. *Theory of Probability*. Addison-Wesley series in Behavioral Science, Quantitative Methods, 1971.
20. D.J.C. MacKay. Introduction to Monte Carlo Methods. In *Learning in Graphical Models, NATO Science Series*, pages 175–204. Kluwer Academic Press, 1998.
21. M. Mitzenmacher and E. Upfal. *Probability and Computing*. Cambridge University Press, 2005.
22. D. Parker. *Implementation of Symbolic Model Checking for Probabilistic System*. PhD Thesis, University of Birmingham, Birmingham, UK, 2001.
23. S. Richter. *Formalizing Integration Theory, with an Application to Probabilistic Algorithms*. Diploma Thesis, Technische Universitat Munchen, Department of Informatics, Germany, 2003.
24. J. Rutten, M. Kwaiatkowska, G. Normal, and D. Parker. *Mathematical Techniques for Analyzing Concurrent and Probabilisitc Systems*, volume 23 of *CRM Monograph Series*. American Mathematical Society, 2004.
25. K. Sen, M. Viswanathan, and G. Agha. VESTA: A Statistical Model-Checker and Analyzer for Probabilistic Systems. In *Proc. IEEE International Conference on the Quantitative Evaluation of Systems*, pages 251–252, 2005.
26. R.D. Yates and D.J. Goodman. *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers*. Wiley, 2005.