J Autom Reasoning (2008) 41:295–323
DOI 10.1007/s10817-008-9113-6

Using Theorem Proving to Verify Expectation and Variance for Discrete Random Variables

Osman Hasan · Sofiène Tahar

Received: 14 November 2008 / Accepted: 14 November 2008 / Published online: 9 December 2008
© Springer Science + Business Media B.V. 2008

Abstract Statistical quantities, such as expectation (mean) and variance, play a vital role in the present age probabilistic analysis. In this paper, we present some formalization of expectation theory that can be used to verify the expectation and variance characteristics of discrete random variables within the HOL theorem prover. The motivation behind this is the ability to perform error free probabilistic analysis, which in turn can be very useful for the performance and reliability analysis of systems used in safety-critical domains, such as space travel, medicine and military. We first present a formal definition of expectation of a function of a discrete random variable. Building upon this definition, we formalize the mathematical concept of variance and verify some classical properties of expectation and variance in HOL. We then utilize these formal definitions to verify the expectation and variance characteristics of the Geometric random variable. In order to demonstrate the practical effectiveness of the formalization presented in this paper, we also present the probabilistic analysis of the Coupon Collector's problem in HOL.

Keywords Coupon collector's problem · Higher-order-logic · HOL theorem prover · Probabilistic analysis · Probability theory · Statistical properties

1 Introduction

Probability has become an essential component of performance and reliability analysis in almost every field of science. The random and unpredictable elements

O. Hasan (✉) · S. Tahar
Department of Electrical and Computer Engineering, Concordia University,
1455 de Maisonneuve W., Montreal, Quebec, H3G 1M8, Canada
e-mail: o_hasan@ece.concordia.ca

S. Tahar
e-mail: tahar@ece.concordia.ca

are mathematically modeled by appropriate random variables and the performance and reliability issues are judged based on the corresponding statistical quantities such as mean and variance. Due to the wide application domain of probability, many researchers around the world are trying to improve the existing computer based probabilistic analysis approaches. The ultimate goal is to come up with a probabilistic analysis framework that includes robust and accurate analysis methods, has the ability to perform analysis for large-scale problems and is user friendly.

Nowadays, probabilistic analysis is usually performed using simulation techniques [4], where the main idea is to approximately answer a query on a probability distribution by analyzing a large number of samples. The simulation approach is quite user friendly as most of the analysis can be automated and really shines in handling problems that cannot be solved analytically. On the other hand, the results are usually inaccurate and large problems cannot be handled because of enormous CPU time requirements. The inaccuracy of the results poses a serious problem when some safety-critical section of the system is being analyzed. An alternative is to use probabilistic model checking [5, 33], which is a formal state-based approach. Due to the formal nature of the models and analysis techniques, the results are always accurate but, like traditional model-checking, this approach is limited by the state-space explosion problem [9]. Another recently proposed formal probabilistic analysis approach is to use higher-order-logic theorem proving to verify probabilistic properties [21]. Due to its inherent soundness and the high expressive nature of the higher-order-logic, this approach not only allows us to acquire accurate results but is also capable of handling any probabilistic problem that can be expressed mathematically. The downside is the enormous amount of user guided formalization that is required to handle various probabilistic analysis issues. Though, a positive aspect is that some foundational formalization in this regard is already available in the open literature, such as the formalization of probability theory and the commonly used discrete [21] and continuous [18] random variables.

In this paper, we further strengthen the higher-order-logic probabilistic analysis approach by presenting the formalization of some expectation theory for discrete random variables in the HOL theorem prover [14]. We mainly develop a formal definition of expectation, which is further utilized to formally define variance as well. In probabilistic analysis, expectation and variance are the most useful characteristics of a random variable, which basically present the average and the dispersion of a random variable, respectively. The paper also includes the verification of some classical properties of expectation and variance in HOL. These properties play a vital role in verifying expectation and variance quantities of discrete probabilistic systems within the HOL theorem prover.

Computer science is one of the key application areas of probabilistic analysis. For example, the average case analysis is usually considered more useful in characterizing an algorithm's performance rather than its worst case analysis. Therefore, in order to illustrate the practical effectiveness of the formalization presented in this paper, we present the probabilistic analysis of the Coupon Collector's problem [28], a well known commercially used algorithm. Some of the recent applications of the Coupon Collector's problem include its usage in packet delivery systems [28], load balancing in peer-to-peer networks [2, 13] and a coalescing particle model which is applicable to population biology [1]. In this paper, We present a higher-order-logic formalization of the Coupon Collector's problem as a probabilistic algorithm using the summation

of a list of Geometric random variables. Then, the formally verified expectation and variance properties are used to verify the expectation and a variance bound of the Coupon Collector's problem in HOL.

The rest of the paper is organized as follows: Section 2 gives a review of the related work. In Section 3, we provide some preliminaries including a brief introduction to the HOL theorem prover and some technical background regarding probabilistic analysis in HOL. Next, we present the HOL formalization of the expectation and variance functions for discrete random variables along with the verification of some of their classical properties in Sections 4 and 5, respectively. We utilize these definitions and properties to verify the mean and variance relations of the Geometric random variable in Section 6, which is followed by the probabilistic analysis of the Coupon Collector's problem in Section 7. Finally, Section 8 concludes the paper.

2 Related Work

Nedzusiak [29] and Bialas [6] were among the first ones to formalize some probability theory in higher-order-logic. Hurd [21] extended their work and developed a framework for the verification of probabilistic algorithms in the HOL theorem prover. He demonstrated the practical effectiveness of his formal framework by successfully verifying the sampling algorithms for four discrete probability distributions, some optimal procedures for generating dice rolls from coin flips, the symmetric simple random walk and the Miller-Rabin primality test based on the corresponding probability distribution properties. Hurd et. al [17] also formalized the *probabilistic guarded-command language* ($pGCL$) in HOL. The $pGCL$ contains both demonic and probabilistic nondeterminism and is thus quite suitable for reasoning about distributed random algorithms. Celiku [8] built upon the formalization of the $pGCL$ to mechanize the quantitative Temporal Logic (*qtl*) and demonstrated the ability to verify temporal properties of probabilistic systems in HOL. An alternative method for probabilistic verification in higher-order-logic has been presented by Audebaud and Paulin-Mohring [3]. Instead of using the measure theoretic concepts of probability space, as is the case in Hurd's approach, Audebaud et al. based their methodology on the monadic interpretation of randomized programs as probabilistic distribution. This approach only uses functional and algebraic properties of the unit interval and has been successfully used to verify a sampling algorithm of the Bernoulli distribution and the termination of various probabilistic programs in the Coq theorem prover.

Building upon Hurd's formalization framework, we have been able to successfully verify the sampling algorithms of a few continuous random variables [18] and the classical *Cumulative Distribution Function* (CDF) properties [20], which play a vital role in verifying arbitrary probabilistic properties of both discrete and continuous random variables. The sampling algorithms for discrete random variables are either guaranteed to terminate or they satisfy probabilistic termination, meaning that the probability that the algorithm terminates is 1. Thus, they can be expressed in HOL by either well formed recursive functions or the *probabilistic while loop* [21]. On the other hand, the implementation of continuous random variables requires non-terminating programs and hence calls for a different approach. In [18], we presented a methodology that can be used to formalize any continuous random variable for which the inverse of the CDF can be expressed in a closed mathematical form. The

core components of our methodology are the Standard Uniform random variable and the Inverse Transform method [12], 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. Using the formalized Standard Uniform random variable and the Inverse Transform method, we were able to formalize continuous random variables, such as Exponential, Rayleigh, etc. and verify their correctness by proving the corresponding CDF properties in HOL.

The formalization, mentioned so far, allows us to express random behaviors as random variables in a higher-order-logic theorem prover and verify the corresponding quantitative probability distribution properties, which is a significant aspect of a probabilistic analysis framework. With the probability distribution properties of a random variable, such as the *Probability Mass Function* (PMF) and the CDF, we are able to completely characterize the behavior of their respective random variables. Though 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 probability of the event that the algorithm succeeds within a certain number of steps. In [19], we tackled the verification of expectation properties in HOL for the first time. We extended Hurd's formalization framework with a formal definition of expectation, which can be utilized to verify the expected values associated with discrete random variables that attain values in positive integers only. In the current paper, rather than restricting our higher-order-logic formalization to simply the expected value of a random variable, we consider the formalization of the expected value of a function of a discrete random variable, whereas the function accepts a positive integer and returns a real value. This includes as a special case the identity function, which covers the formalization of the expected value of a random variable that attains values in the positive integers only. The main advantage of this new definition is that it allows us to formally specify and verify variance properties of discrete random variables within a higher-order-logic theorem prover; a novelty that has not been available so far.

Richter [32] formalized a significant portion of the Lebesgue integration theory in higher-order logic using Isabelle/Isar [30]. He also linked the Lebesgue integration theory to probabilistic algorithms, developing upon Hurd's [21] framework, and presented the formalization of the first moment method. The formalization and verification of statistical characteristics regarding continuous random variables in a theorem prover 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 Richter's formalization [32] can be built upon for formalizing the mathematical concepts of expectation and variance for continuous random variables.

Statistical characteristics, such as expectation and variance, are one of the most useful tools in probabilistic analysis and therefore their evaluation within a model checker is being explored in the probabilistic model checking community [5, 33]. Some probabilistic model checkers, such as PRISM [23] and VESTA [35], offer the capability of verifying expected values in a semi-formal manner. For example, in the PRISM model checker, the basic idea is to augment probabilistic models with cost or

rewards: real values associated with certain states or transitions of the model. This way, the expected value properties, related to these rewards, can be analyzed by PRISM. It is important to note that the meaning ascribed to the these properties is, of course, dependent on the definitions of the rewards themselves and thus there is always some risk of verifying false properties. Similarly, to the best of our knowledge, no model checking algorithm exists in the open literature so far that allows us to verify variance properties. On the other hand, the proposed theorem proving based probabilistic analysis can be used to precisely reason about both expectation and variance characteristics due to the high expressivity of higher-order-logic.

Probabilistic model checking is capable of providing exact solutions to probabilistic properties in an automated way; however it is also limited to systems that can only be expressed as a probabilistic finite state machine. In contrast, the theorem proving based probabilistic verification is an interactive approach but is capable of handling all kinds of probabilistic systems including the *unbounded* ones. Another major limitation of the probabilistic model checking approach is the state space explosion [9], which is not an issue with the proposed theorem proving based probabilistic analysis approach.

3 Preliminaries

In this section, we provide a brief introduction to the HOL theorem prover and verification of probabilistic algorithms in HOL. The intent is to introduce the main ideas along with some notation that is going to be used in the next few sections.

3.1 HOL Theorem Prover

The HOL theorem prover is an interactive theorem prover that is capable of conducting proofs in higher-order logic. It utilizes the simple type theory of Church [10] along with Hindley-Milner polymorphism [27] to implement higher-order logic. HOL has been successfully used as a verification framework for both software and hardware systems as well as a platform for the formalization of pure mathematics. It supports the formalization of various mathematical theories including sets, natural numbers, real numbers, measure and probability. The HOL theorem prover includes many proof assistants and automatic proof procedures. The user interacts with a proof editor and provides the necessary tactics to prove goals while some of the proof steps are solved automatically by the automatic decision procedures.

In order to ensure secure theorem proving, the logic in the HOL system is represented in the strongly-typed functional programming language ML [31]. The ML abstract data types are then used to represent higher-order-logic theorems and the only way to interact with the theorem prover is by executing ML procedures that operate on values of these data types. Users can prove theorems using a natural deduction style by applying inference rules to axioms or previously generated theorems. The HOL core consists of only basic 5 axioms and 8 primitive inference rules, which are implemented as ML functions. Soundness is assured as every new theorem must be created from the 5 basic axioms and the 8 primitive inference rules or any other pre-existing theorems/inference rules.

Table 1 HOL symbols and functions

HOL symbol	Standard symbol	Meaning
$\wedge$	*and*	Logical *and*
$\vee$	*or*	Logical *or*
$\neg$	*not*	Logical *negation*
::	*cons*	Adds a new element to a list
`el n L`	L_n	n^{th} element of list L
`mem a L`	$a \in L$	True if a is a member of list L
`length L`	$\|L\|$	Length of list L
(a, b)	a x b	A pair of two elements
`fst`	fst (a, b) = a	First component of a pair
`snd`	snd (a, b) = b	Second component of a pair
`λx.t`	$\lambda x.t$	Function that maps x to $t(x)$
{x|P(x)}	$\{\lambda x.P(x)\}$	Set of all x such that $P(x)$
`num`	{0, 1, 2, ...}	Positive Integers data type
`real`	All Real numbers	Real data type
`suminf(λn.f(n))`	$\lim_{k\to\infty} \sum_{n=0}^{k} f(n)$	Infinite summation of f
`summable(λn.f(n))`	$\exists x. \lim_{k\to\infty} \sum_{n=0}^{k} f(n) = x$	Summation of f is convergent

We selected the HOL theorem prover for the proposed formalization mainly because of its inherent soundness, ability to handle higher-order logic and in order to benefit from the in-built mathematical theories for measure and probability. Table 1 summarizes some of the HOL symbols used in this paper and their corresponding mathematical interpretations.

3.2 Verifying Probabilistic Algorithms in HOL

The foremost criterion for developing a higher-order-logic theorem-proving based probabilistic analysis framework is to be able to formalize random variables in higher-order logic. This section presents a methodology, initially proposed in [21], for the formalization of probabilistic algorithms, which in turn can be used to model random variables in HOL.

The probabilistic algorithms can be formalized in higher-order logic by thinking of them as deterministic functions with access to an infinite Boolean sequence $\mathbb{B}^\infty$; a source of infinite random bits with data type $(num \rightarrow bool)$ [21]. 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 algorithms terminate, they return the result along with the remaining portion of the infinite Boolean sequence to be used by other programs. Thus, a probabilistic algorithm which 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 s is the infinite Boolean sequence and `shd` and `stl` are the sequence equivalents of the list operation *'head'* and *'tail'*. The probabilistic programs can also be expressed in the more general state-transforming monad where the states are the infinite Boolean sequences.

```
⊢ ∀ a s. unit a s = (a,s)
⊢ ∀ f g s. bind f g s = g (fst (f s)) (snd (f s))
```

The HOL functions `fst` and `snd`, used above, return the first and second components of a pair, respectively. The `unit` operator is used to lift values to the monad, and the `bind` is the monadic analogue of function application. All monad laws hold for this definition, and the notation allows us to write functions without explicitly mentioning the sequence that is passed around, e.g., function `bit` can be defined as

```
⊢ bit_monad = bind sdest (λb. if b then unit 1 else unit 0)
```

where `sdest` gives the head and tail of a sequence as a pair (*shd* `s`, *stl* `s`).

Hurd [21] also presents some formalization of the 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 probabilistic programs such as

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

where $\{x|C(x)\}$ represents a set of all x that satisfy the condition C in HOL.

The measurability and independence of a probabilistic function are important concepts in probability theory. A property `indep_fn`, called *strong function independence*, is introduced in [21] such that if $f \in$ `indep_fn`, then f will be both measurable and independent. It has been shown in [21] that a function is guaranteed to preserve *strong function independence*, if it accesses the infinite Boolean sequence using only the `unit`, `bind` and `sdest` primitives. All reasonable probabilistic programs preserve *strong function independence*, and these extra properties are a great aid to verification.

4 Expectation for Discrete Random Variables

In this section, we first present a higher-order-logic formalization of the expectation function for discrete random variables. We later utilize this definition to verify a few classical expectation properties in HOL and some details about the proofs are also included.

4.1 Formalization of Expectation in HOL

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 [7]

$$Ex[X] = \sum_i x_i Pr(X = x_i) \tag{1}$$

where Pr and $\sum_i$ denote the probability function and the summation carried over all the possible values of the random variable X, respectively. The above definition only holds if the summation is convergent, i.e., $\sum_i x_i Pr(X = x_i) < \infty$. Instead of formalizing this general definition of expectation based on the principles of probability space, we concentrate on one of its variants that deals with discrete random variables that take on values only in the positive integers, i.e., $\{0, 1, 2, \cdots\}$.

This choice has been made mainly because of two reasons. First of all, in most of the engineering and scientific probabilistic analysis problems, we end up dealing with discrete random variables that attain values in positive integers only. For example, consider the cases of analyzing the performance of algorithms [28], cryptographic [26] and communication protocols [25], etc. Secondly, this simplification allows us to model the expectation function using the summation of a real sequence, which has already been formalized in the HOL theorem prover [16], and thus speed up the associated formalization and verification process by a considerable extent.

The expectation for a function of a discrete random variable, which attains values in the positive integers only, is defined as follows [24]

$$Ex[f(R)] = \sum_{n=0}^{\infty} f(n) Pr(R = n) \tag{2}$$

where R is the discrete random variable and f represents a function of the random variable R. The above definition only holds if the associated summation is convergent, i.e., $\sum_{n=0}^{\infty} f(n) Pr(R = n) < \infty$.

Equation (2) can be formalized in HOL, for a discrete random variable R that attains values in positive integers only and a function f that maps this random variable to a *real* value, as follows

Definition 1 Expectation of Function of a Discrete Random Variable

```
⊢ ∀ f R. expec_fn f R =
suminf (λn. (f n) ℙ{s | fst(R s) = n})
```

where the mathematical notions of the probability function $\mathbb{P}$ and random variable `R` have been inherited from [21], as presented in Section 3.2. The HOL function `suminf` represents the infinite summation of a *real* sequence [16]. The function `expec_fn` accepts two parameters, the function f of type $(num \rightarrow real)$ and the positive integer valued random variable R and returns a *real* number.

Next, we define the expected value of a discrete random variable that attains values in positive integers only as a special case of the expected value of a function of a discrete random variable.

Definition 2 Expectation of a Discrete Random Variable

$$\vdash \forall \texttt{R. expec R = expec\_fn (}\lambda\texttt{n. n) R}$$

where the lambda abstraction function `(λn. n)` implements the identity function. The function `expec` accepts a positive integer valued random variable R and returns its expectation as a *real* number.

4.2 Verification of Expectation Properties in HOL

In this section, we utilize the formal definitions of expectation, developed in the last section, to prove some classical properties of the expectation [34]. These properties not only verify the correctness of our definitions but also play a vital role in verifying the expectation characteristic of discrete random components of probabilistic systems, as will be seen in Section 7 for the case of the Coupon's Collector's problem.

4.2.1 Expectation of a Constant

$$Ex[c] = c \tag{3}$$

where c is a positive integer. The random variable in this case is the degenerate random variable $R \equiv c$, where $R(s) = c$ for every $s \in$ *sample space*. It can be formally expressed as `unit c`, where the monadic operator `unit` is described in Section 3.2. Using this representation and the definition of expectation, given in Definition 2, the above property can be expressed in HOL as follows.

Theorem 1 *Expectation of a Constant*

$$\vdash \forall \texttt{ c. expec (unit c) = c}$$

Rewriting the proof goal of the above property with Definition 2, we get

$$\lim_{\texttt{k}\to\infty}\left(\sum_{\texttt{n=0}}^{\texttt{k}} \texttt{n}\,\mathbb{P}\{\texttt{s} \mid \texttt{fst(unit c s)} = \texttt{n}\}\right) = \texttt{c} \tag{4}$$

where according to the HOL definition of the summation of a real sequence, the expression $\sum_{\texttt{n=a}}^{\texttt{b}}\,\texttt{f}$ means the summation of `b` subsequent terms of the real sequence `f` starting from the term `f(a)`. Thus, in this paper, the term $\sum_{\texttt{n=0}}^{\texttt{k}}\,\texttt{f}$ represents $\texttt{f(0)} + \texttt{f(1)} \cdots + \texttt{f(k} - \texttt{1)}$. Now, the probability term on the *left-and-side* (LHS) of the above subgoal can be expressed as follows

$$\forall\,\texttt{n c}.\ \mathbb{P}\{\texttt{s} \mid \texttt{fst(unit c s)} = \texttt{n}\} = (\texttt{if (c} = \texttt{n) then 1 else 0}) \tag{5}$$

and the proof is based on the basic probability theory laws and the functional independence property of the random variable `unit c`. Using this property, the subgoal of (4) can be rewritten as follows

$$\lim_{\texttt{k}\to\infty}\left(\sum_{\texttt{n=0}}^{\texttt{k}} \texttt{n (if (c} = \texttt{n) then 1 else 0)}\right) = \texttt{c} \tag{6}$$

The summation on the *right-hand-side* (RHS) of the above subgoal can be proved to be convergent since its value remains the same for all values of n that are greater than c. Using this fact and the summation properties of a *real* sequence the above subgoal can be verified in HOL, which concludes the proof of Theorem 1.

4.2.2 Linearity of Expectation for Discrete Random Variables

$$Ex\left[\sum_{i=1}^{n} R_i\right] = \sum_{i=1}^{n} Ex[R_i] \tag{7}$$

where R_i represents a sequence of n discrete random variables. According to the linearity of expectation property, the expectation of a sum of random variables equals the sum of their individual expectations. It is one of the most important properties of expectation as it allows us to verify the expectation properties of random behaviors involving multiple random variables without going into the complex verification of their joint probability distribution properties. Thus, its verification is a significant step towards using HOL as a successful probabilistic analysis framework.

We split the verification of linearity of expectation property in two major steps. Firstly, we verify the property for two discrete random variables and then extend the results by induction to prove the general case. The linearity of expectation property can be defined for any two discrete random variables X and Y as follows.

$$Ex[X + Y] = Ex[X] + Ex[Y] \tag{8}$$

To prove the above relationship in HOL, we proceed by first defining a function that models the summation of two random variables.

Definition 3 Summation of Two Random Variables

```
⊢ ∀ X Y. sum_two_rv X Y
= bind X (λa. bind Y (λb. unit (a + b)))
```

The function, `sum_two_rv`, accepts two random variables and returns one random variable that represents the sum of the two argument random variables. It is important to note that the above definition implicitly ensures that the call of the random variable Y is independent of the result of the random variable X. This is true because the infinite Boolean sequence that is used for the computation of Y is the remaining portion of the infinite Boolean sequence that has been used for the computation of X. This characteristic led us to prove that the function `sum_two_rv` preserves *strong function independence*, which is the most significant property in terms of verifying properties on probabilistic functions.

Lemma 1 *sum_two_rv Preserves Strong Function Independence*

```
⊢ ∀ X Y. X ∈ indep_fn ∧ Y ∈ indep_fn
⇒ ((sum_two_rv X Y) ∈ indep_fn)
```

The above property can be verified in HOL using the fact that the function `sum_two_rv` accesses the infinite Boolean sequence using the `unit` and `bind` operators.

Now, the linearity of expectation property for two discrete random variables, which preserve *strong function independence*, with *well-defined* expectation values, i.e., the summation in their expectation definition is convergent, can be stated in HOL using the `sum_two_rv` function as follows.

Lemma 2 *Linearity of Expectation for Two Discrete Random Variables*

```
⊢ ∀X Y. X ∈ indep_fn ∧ Y ∈ indep_fn
∧ summable(λn. n ℙ{s | fst(X s) = n})
∧ summable(λn. n ℙ{s | fst(Y s) = n})
⇒ (expec (sum_two_rv X Y) = expec X + expec Y)
```

where `summable` *accepts a real sequence and returns True if the infinite summation of this sequence is convergent* (*i.e.*, `summmable M` $= \exists x. \lim_{k\to\infty} (\sum_{n=0}^{k} M(n)) = x$).

Rewriting the proof goal of Lemma 2 with the definitions of the functions `expec`, `sum_two_rv` and `summable`, simplifying it with some infinite summation properties and removing the monad notation, we reach the following subgoal in HOL.

$$\left(\lim_{k\to\infty}\left(\sum_{n=0}^{k}(n\,\mathbb{P}\,\{s \mid \mathtt{fst}(X\ s) = n\})\right) = p\right) \wedge \left(\lim_{k\to\infty}\left(\sum_{n=0}^{k}(n\,\mathbb{P}\,\{s \mid \mathtt{fst}(Y\ s) = n\})\right) = q\right)$$
$$\Rightarrow \left(\lim_{k\to\infty}\left(\sum_{n=0}^{k}(n\,\mathbb{P}\{s \mid \mathtt{fst}(X\ s) + \mathtt{fst}(Y\ (\mathtt{snd}\ (X\ s)) = n\})\right)\right) = (p+q)\right) \tag{9}$$

The set in the conclusion of the above implication can be proved to be equal to the countable union of a sequence of events as follows

$$\begin{aligned}&\forall X\ Y\ n.\ X \in \mathtt{indep\_fn} \wedge Y \in \mathtt{indep\_fn}\\ &\quad\Rightarrow \{s \mid \mathtt{fst}(X\ s) + \mathtt{fst}(Y\ (\mathtt{snd}(X\ s))) = n\}\\ &\quad= \bigcup_{i\le n}\{s \mid (\mathtt{fst}(X\ s) = i) \wedge (\mathtt{fst}(Y(\mathtt{snd}(X\ s))) = n - i)\}\end{aligned} \tag{10}$$

using the properties verified in the HOL theory of sets. All the events in the above sequence of events are mutually exclusive. Thus, (10) along with the additive law

of probability, given in the HOL theory of probability, can be used to simplify the subgoal, given in (9), as follows.

$$\left(\lim_{k\to\infty}\left(\sum_{n=0}^{k}(n\,\mathbb{P}\,\{s \mid \mathtt{fst(X\ s)} = n\})\right) = p\right) \wedge \left(\lim_{k\to\infty}\left(\sum_{n=0}^{k}(n\,\mathbb{P}\,\{s \mid \mathtt{fst(Y\ s)} = n\})\right) = q\right)$$
$$\Rightarrow \lim_{k\to\infty}\left(\sum_{n=0}^{k}\left(n\sum_{i=0}^{n+1}\mathbb{P}\{s \mid (\mathtt{fst(X\ s)} = i) \wedge (\mathtt{fst(Y(snd(X\ s)))} = n - i)\}\right)\right) = (p+q) \tag{11}$$

Next, we found a real sequence that is easier to handle and has the same limit value as the real sequence given in the conclusion of the above implication.

$$\left(\lim_{k\to\infty}\left(\sum_{n=0}^{k} n\left(\sum_{i=0}^{n+1}\mathbb{P}\{s \mid (\mathtt{fst(X\ s)} = i) \wedge (\mathtt{fst(Y\ (snd(X\ s)))} = n - i)\}\right)\right)\right)$$
$$= \left(\lim_{k\to\infty}\left(\sum_{a=0}^{k}\sum_{b=0}^{k}(a+b)\,(\mathbb{P}\{s \mid (\mathtt{fst(X\ s)} = a) \wedge (\mathtt{fst(Y\ (snd(X\ s)))} = b)\})\right)\right) \tag{12}$$

Using this new real sequence and rearranging the terms based on summation properties given in the HOL theories of *real* numbers, we can rewrite the subgoal, given in (11), as follows.

$$\left(\lim_{k\to\infty}\left(\sum_{n=0}^{k}(n\,\mathbb{P}\,\{s \mid \mathtt{fst(X\ s)} = n\})\right) = p\right) \wedge \left(\lim_{k\to\infty}\left(\sum_{n=0}^{k}(n\,\mathbb{P}\,\{s \mid \mathtt{fst(Y\ s)} = n\})\right) = q\right)$$
$$\Rightarrow \left(\lim_{k\to\infty}\left(\sum_{a=0}^{k}\sum_{b=0}^{k} a(\mathbb{P}\{s \mid (\mathtt{fst(X\ s)} = a) \wedge (\mathtt{fst(Y\ (snd(X\ s)))} = b)\})\right) = p\right)$$
$$\wedge \left(\lim_{k\to\infty}\left(\sum_{a=0}^{k}\sum_{b=0}^{k} b(\mathbb{P}\{s \mid (\mathtt{fst(X\ s)} = a) \wedge (\mathtt{fst(Y\ (snd(X\ s)))} = b)\})\right) = q\right) \tag{13}$$

The two limit expressions in the conclusion of the above implication can now be proved to be *True* using some elementary properties in the HOL theories of probability, sets and *real* numbers, which also concludes the proof for Lemma 2.

The next step is to generalize Lemma 2 to verify the linearity of expectation property, given in (7), using induction. For this purpose, we define a function that models the summation of a list of discrete random variables.

Definition 4 Summation of *n* Random Variables

```
⊢ (sum_rv_lst [] = unit0)
∧ ∀ h t. (sum_rv_lst (h::t)
= bind h (λa. bind (sum_rv_lst t)
          (λb. unit (a + b))))
```

The function, `sum_rv_lst`, accepts a list of random variables and returns their sum as a single random variable. Just like the function, `sum_two_rv`, the function `sum_rv_lst` also preserves *strong function independence*, if all random variables in the given list preserve it. This property can be verified using the fact that it accesses the infinite Boolean sequence using the `unit` and `bind` primitives only.

Lemma 3 *`sum_rv_lst` Preserves Strong Function Independence*

```
⊢ ∀ L. (∀ R. (mem R L) ⇒ R ∈ indep_fn)
⇒ ((sum_rv_lst L) ∈ indep_fn)
```

where the predicate `mem` *is defined in the HOL list theory and returns True if its first argument is an element of the list that it accepts as the second argument.*

Now, the linearity of expectation property for *n* discrete random variables, which preserve *strong function independence* and for which the infinite summation in the expectation definition converges, can be stated in HOL as follows

Theorem 2 *Linearity of Expectation Property*

```
⊢ ∀ L. (∀ R. (mem R L) ⇒ (R ∈ indep_fn)
∧ (summable (λn. n ℙ{s | fst(R s) = n})))
⇒ (expec(sum_rv_lst L)
= ∑_{n=0}^{length L} (expec (el (length L - (n+1)) L)))
```

where the function `length`*, defined in the HOL list theory, returns the length of its list argument and the function* `el`*, also defined in the list theory, accepts a positive integer number, say n, and a list and returns the* n^{th} *element of the given list. Thus, the LHS of Theorem* 2 *represents the expectation of the summation of a list L of random variables. Whereas, the RHS represents the summation of the expectations of all elements in the same list L. Theorem* 2 *can be proved by applying induction on the list argument of the function* `sum_rv_lst`*, and simplifying the subgoals using Lemmas* 2 *and* 3*.*

4.2.3 Expectation of a Discrete Random Variable Multiplied by a Constant

$$Ex[aR] = aEx[R] \tag{14}$$

where R is a discrete random variable that attains values in the positive integers only and a is a positive integer. This property can be expressed in HOL for a

random variable R that preserves *strong function independence* and has a *well-defined* expected value as follows.

Theorem 3 *Expectation of a Discrete Random Variable Multiplied by a Constant*

```
⊢ ∀ R a. R ∈ indep_fn ∧ summable(λn. n ℙ{s | fst(R s) = n})
⇒ expec (bind R (λm. unit (a m))) = a (expec R)
```

The HOL proof proceeds by first performing case analysis on the variable a. For the case when a is 0, the RHS of the proof goal becomes 0. Whereas, using the definition of expectation, the LHS reduces to the expression

$$\lim_{k\to\infty}\left(\sum_{n=0}^{k} n\,\mathbb{P}\{s|\ 0\ =\ n\}\right) \tag{15}$$

which is also equal to 0 as ∀n. n ℙ{s | 0 = n} = 0, since ∀n. 0 < n ⇒ ℙ {s | 0 = n}=0. On the other hand, when a is not equal to 0, i.e., $(0 < a)$, the proof goal may be simplified as follows

$$\lim_{k\to\infty}\left(\sum_{n=0}^{k} n\,\mathbb{P}\{s|\ a\ \mathtt{fst}(R\ s) = n\}\right) = a\lim_{k\to\infty}\left(\sum_{n=0}^{k} n\,\mathbb{P}\{s|\ a\ \mathtt{fst}(R\ s) = a\ n\}\right) \tag{16}$$

using the definition of expectation and the multiplication cancelation property of positive integers. Next, we proved in HOL that

$$\forall k.\left(\sum_{n=0}^{k} n\,\mathbb{P}\{s|\ a\ \mathtt{fst}(R\ s) = n\}\right) = a\left(\sum_{n=0}^{B(k)} n\,\mathbb{P}\{s|\ a\ \mathtt{fst}(R\ s) = a\ n\}\right) \tag{17}$$

where B(k) = if (k MOD a = 0) then (k DIV a) else ((k DIV a) + 1) and MOD and DIV represent the *modulo* and *division* functions for positive integers in HOL. This allows us to rewrite our proof goal as follows

$$\lim_{k\to\infty} a\left(\sum_{n=0}^{B(k)} n\,\mathbb{P}\{s|\ a\ \mathtt{fst}(R\ s) = a\ n\}\right) = a\lim_{k\to\infty}\left(\sum_{n=0}^{k} n\,\mathbb{P}\{s|\ a\ \mathtt{fst}(R\ s) = a\ n\}\right) \tag{18}$$

which can be proved using the properties of limit of a real sequence in HOL [16], since both of the real sequences in the above equation converge to the same value as the value of k becomes very very large. This concludes the proof of the expectation property given in Theorem 3.

4.2.4 Expectation of a Discrete Random Variable Added and Multiplied by Constants

$$Ex[a + bR] = a + bEx[R] \tag{19}$$

This property allows us to express the expectation value of a positive integer valued random variable R added and multiplied by two positive integers a and b,

respectively, in terms of the expectation of the random variable R. It can be expressed in HOL for a random variable R that preserves *strong function independence* and has a *well-defined* expected value as follows.

Theorem 4 *Expectation of a Discrete Random Variable Added and Multiplied by Constants*

```
⊢ ∀ R a b. R ∈ indep_fn
∧ summable(λn. n ℙ{s | fst(R s) = n})
⇒ expec (bind R (λm. unit (a + b m)))
= a + b (expec R)
```

Theorem 4 can be proved in HOL using the expectation properties, given in Theorems 1, 2 and 3.

5 Variance for Discrete Random Variables

In this section, we utilize the formal definition of expectation of a function of a random variable, developed in Section 4, to define a variance function for discrete random variables that attain values in positive integers only. We later utilize this definition to verify a couple of classical variance properties in HOL and some details about the proofs are also included.

5.1 Formalization of Variance in HOL

In the field of probabilistic analysis, it is often desirable to summarize the essential properties of distribution of a random variable by certain suitably defined measures. In the previous section, we formalized one such measure, i.e., the expectation, which yields the weighted average of the possible values of a random variable. Quite frequently, along with the average value, we are also interested in finding how typical is the average value or in other words the chances of observing an event far from the average. One possible way to measure the variation, or spread, of these values is to consider the quantity $Ex[|R - Ex[R]|]$, where $||$ denote the *abs* function. However, it turns out to be mathematically inconvenient to deal with this quantity, so a more tractable quantity called *variance* is usually considered, which returns the expectation of the square of the difference between R and its expectation [7].

$$Var[R] = Ex[(R - Ex[R])^2] \tag{20}$$

Now, we formalize this definition of variance in HOL for the case of discrete random variables that can attain values in the positive integers only. For this purpose, we utilize the definitions of expectation, given in Definitions 1 and 2.

Definition 5 Variance of a Discrete Random Variable

```
⊢ ∀ R. variance R = expec_fn (λn. (n - expec R)²) R
```

The function, `variance`, accepts a discrete random variable R that attains values in the positive integers only and returns its variance as a *real* number.

5.2 Verification of Variance Properties in HOL

In this section, we prove two of the most significant and widely used properties of the variance function [28]. These properties not only verify the correctness of our definition but also play a vital role in verifying the variance properties of discrete random variables as will be seen in Sections 6 and 7 of this paper.

5.2.1 Variance in Terms of Moments

$$Var[R] = Ex[R^2] - (Ex[R])^2 \tag{21}$$

where R is a discrete random variable that can attain values in the positive integers only. This alternative definition of variance is much easier to work with than the previous one and thus aids significantly in the process of verifying variance properties for discrete random variables. This property can be stated in HOL using the formal definition of variance and expectation as follows.

Theorem 5 *Variance in Terms of Moments*

```
⊢ ∀ R. R ∈ indep_fn
∧(summable(λn. n ℙ{s | fst(R s) = n}))
∧(summable(λn.n² ℙ{s | fst(R s) = n}))
⇒(variance R = expec_fn (λn. n²) R - (expec R)²)
```

The assumption in Theorem 5 ensures that the random variable R preserves the *strong function independence* and its expectation and second moment are *well-defined*. The theorem can be proved by using the function definitions of `expec_fn`, `expec` and `variance` along with some arithmetic reasoning and properties from the HOL *real* number theories.

5.2.2 Linearity of Variance for Independent Discrete Random Variables

$$Var\left[\sum_{i=1}^{n} R_i\right] = \sum_{i=1}^{n} Var[R_i] \tag{22}$$

where R_i represents a sequence of n independent discrete random variables. Like the linearity of expectation property, the linearity of variance property also allows us to verify the variance properties of probabilistic systems involving multiple random variables without going into the complex verification of their joint probability distribution properties.

The proof steps for the linearity of variance property are quite similar to the proof steps for the linearity of expectation property. We split the verification task

in two major steps. Firstly, we verify the property for two discrete random variables and then extend the results by induction to prove the general case. The linearity of variance property can be defined for any two independent discrete random variables X and Y as follows

$$Var[X + Y] = Var[X] + Var[Y] \tag{23}$$

Using the function `sum_two_rv`, given in Definition 3, the linearity of variance property for two independent discrete random variables, which attain values in the positive integers only, preserve the *strong function independence* and have *well-defined* expectation and second moment, can be stated in HOL as follows.

Lemma 4 *Linearity of Variance for Two Discrete Random Variables*

```
⊢ ∀ X Y. X ∈ indep_fn ∧ Y ∈ indep_fn
∧ (summable(λn. n ℙ{s | fst(X s) = n}))
∧ (summable(λn. n ℙ{s | fst(Y s) = n}))
∧ (summable(λn.n² ℙ{s | fst(X s) = n}))
∧ (summable(λn.n² ℙ{s | fst(Y s) = n}))
⇒(variance(sum_two_rv X Y)
= variance X + variance Y)
```

Rewriting the above theorem with the definitions of the functions `variance`, `expec_fn`, `expec` and `summable`, simplifying it with some infinite summation properties and Theorem 2 and removing the monad notation, we reach the following subgoal.

$$\begin{aligned}
&\left(\lim_{k\to\infty}\sum_{n=0}^{k}(n\,\mathbb{P}\,\{s \mid \mathtt{fst}(X\,s) = n\}) = p\right)\\
&\wedge\left(\lim_{k\to\infty}\sum_{n=0}^{k}(n\,\mathbb{P}\,\{s \mid \mathtt{fst}(Y\,s) = n\}) = q\right)\\
&\wedge\left(\lim_{k\to\infty}\sum_{n=0}^{k}(n^2\,\mathbb{P}\,\{s \mid \mathtt{fst}(X\,s) = n\}) = r\right)\\
&\wedge\left(\lim_{k\to\infty}\sum_{n=0}^{k}(n^2\,\mathbb{P}\,\{s \mid \mathtt{fst}(Y\,s) = n\}) = t\right)\\
&\wedge\left(\lim_{k\to\infty}\left(\sum_{n=0}^{k}((n - \mathtt{expec}\,X)^2\,\mathbb{P}\,\{s \mid \mathtt{fst}(X\,s) = n\})\right) = u\right)
\end{aligned}$$

$$\wedge \left(\lim_{k\to\infty} \left(\sum_{n=0}^{k} ((n - \texttt{expec X})^2\ \mathbb{P}\ \{s \mid \texttt{fst(Y s)} = n\}) \right) = v \right)$$

$$\Rightarrow \lim_{k\to\infty} \sum_{n=0}^{k} ((n - (\texttt{expec X} + \texttt{expec Y}))^2)\ \mathbb{P}\{s \mid \texttt{fst(X s)} + \texttt{fst(Y (snd (X s)))} = n\}$$

$$= (u + v) \tag{24}$$

Using the uniqueness of the limit value of a real sequence, and some properties of summation of real sequences, it can be proved in a straight forward manner that $u = r - p^2$ and $v = t - q^2$ under the given assumptions in the above subgoal. This allows us to rewrite the above subgoal as follows.

$$\left(\lim_{k\to\infty} \sum_{n=0}^{k} (n\ \mathbb{P}\ \{s \mid \texttt{fst(X s)} = n\}) = p \right)$$
$$\wedge \left(\lim_{k\to\infty} \sum_{n=0}^{k} (n\ \mathbb{P}\ \{s \mid \texttt{fst(Y s)} = n\}) = q \right)$$
$$\wedge \left(\lim_{k\to\infty} \sum_{n=0}^{k} (n^2 \mathbb{P}\ \{s \mid \texttt{fst(X s)} = n\}) = r \right)$$
$$\wedge \left(\lim_{k\to\infty} \sum_{n=0}^{k} (n^2\ \mathbb{P}\ \{s \mid \texttt{fst(Y s)} = n\}) = t \right)$$
$$\Rightarrow \lim_{k\to\infty} \sum_{n=0}^{k} ((n^2 - 2(p+q)n + (p+q)^2)\ \mathbb{P}\{s \mid \texttt{fst(X s)} + \texttt{fst(Y (snd (X s)))} = n\})$$
$$= (r + t - (p^2 + q^2)) \tag{25}$$

Next we split the real sequence of the conclusion, in the above subgoal, in a sum of three real sequences, corresponding to the terms n^2, $-2(p+q)n$ and $(p+q)^2$ found on the LHS. Now, using the results of Theorem 2 along with some probability laws, it can be shown that the second and third sequences out of these three converge to $(-2(p+q)(p+q))$ and $(p+q)^2$, respectively. This allows us to rewrite the subgoal of (25) as follows.

$$\left(\lim_{k\to\infty} \left(\sum_{n=0}^{k} n\ \mathbb{P}\ \{s \mid \texttt{fst(X s)} = n\} \right) = p \right)$$
$$\wedge \left(\lim_{k\to\infty} \sum_{n=0}^{k} (n\ \mathbb{P}\ \{s \mid \texttt{fst(Y s)} = n\}) = q \right)$$
$$\wedge \left(\lim_{k\to\infty} \sum_{n=0}^{k} (n^2\ \mathbb{P}\ \{s \mid \texttt{fst(X s)} = n\}) = r \right)$$

$$\wedge \left(\lim_{k \to \infty} \sum_{n=0}^{k} (n^2 \, \mathbb{P} \, \{s \mid \texttt{fst}(Y\ s) = n\}) = t \right)$$

$$\Rightarrow \lim_{k \to \infty} \sum_{n=0}^{k} ((n^2)\, \mathbb{P}\{s \mid \texttt{fst}(X\ s)\ +\ \texttt{fst}(Y\ (\texttt{snd}\ (X\ s))) = n\}) = (r + t + 2pq) \tag{26}$$

Just like the proof of the linearity of expectation property, we replace the real sequence in the conclusion of the above subgoal by a real sequence that is simpler to handle and shares the same limit value as this one, under the given assumptions.

$$\left(\lim_{k \to \infty} \left(\sum_{n=0}^{k} n^2 (\mathbb{P}\{s \mid \texttt{fst}(X\ s) + \texttt{fst}(Y\ (\texttt{snd}(X\ s))) = n\}) \right) \right)$$

$$= \left(\lim_{k \to \infty} \left(\sum_{a=0}^{k} \sum_{b=0}^{k} (a^2 + ab)(\mathbb{P}\{s \mid (\texttt{fst}(X\ s)\ = a) \wedge (\texttt{fst}(Y\ (\texttt{snd}(X\ s))) = b)\} \right. \right.$$

$$\left. \left. + \mathbb{P}\{s \mid (\texttt{fst}(X\ s)\ = b) \wedge (\texttt{fst}(Y\ (\texttt{snd}(X\ s))) = a)\}) \right) \right) \tag{27}$$

The subgoal given in (26) can now be proved using the above result and some arithmetic reasoning in HOL, which concludes the proof of Lemma 4.

The next step is to generalize Lemma 4 to verify the linearity of variance property for n discrete random variables (22), which can be stated in HOL as follows.

Theorem 6 *Linearity of Variance Property*

$$\begin{aligned}
&\vdash \forall\ L.\ (\forall\ R.\ (\texttt{mem}\ R\ L) \Rightarrow ((R \in \texttt{indep_fn}) \\
&\wedge (\texttt{summable}\ (\lambda n.\ n\ \mathbb{P}\{s \mid \texttt{fst}(R\ s) = n\})) \\
&\wedge (\texttt{summable}\ (\lambda n.\ n^2\ \mathbb{P}\{s \mid \texttt{fst}(R\ s) = n\})))) \\
&\Rightarrow (\texttt{variance}\ (\texttt{sum_rv_lst}\ L) \\
&= \sum_{n=0}^{length\ L} (\texttt{variance}\ (\texttt{el}\ (\texttt{length}\ L - (n+1))\ L)))
\end{aligned}$$

Theorem 6 can be proved by applying induction on the list argument of the function `sum_rv_lst`, and simplifying the subgoals using Lemmas 3 and 4.

6 Geometric Random Variable

In this section, we present the formalization and verification of expectation and variance properties for the Geometric random variable in HOL. This exercise illustrates the usefulness of the definitions that we developed in Sections 4 and 5 for the verification of expectation and variance properties associated with discrete random variables, respectively. The theorems developed in the current section, also play a central role in conducting the probabilistic analysis of the Coupon Collector's problem, which is modeled as a list of Geometric random variables, given in Section 7.

6.1 Formalization of Geometric(p) Random Variable in HOL

Geometric(p) random variable can be modeled as a function that returns the index of the first success in an infinite sequence of Bernoulli(p) trials [11]. Therefore, we first need to have a formal definition of the Bernoulli(p) random variable before we consider the formalization of Geometric(p) random variable in HOL. For this purpose, we utilized a sampling algorithm of the Bernoulli(p) random variable, presented in [21], which returns *True* with probability p and *False* otherwise. This sampling algorithm of Bernoulli(p) random variable was verified to be correct by proving its PMF property in HOL [21].

```
⊢ ∀ p. 0 ≤ p ∧ p ≤ 1 ⇒ ℙ {s | fst (prob_bern p s)} = p
```

The Geometric(p) random variable can now be sampled by extracting random bits from the function `prob_bern` and stopping as soon as the first *False* is encountered and returning the number of trials performed till this point. We modeled it using the *probabilistic while loop* [21] in HOL as follows.

Definition 6 A Sampling Algorithm for Geometric(p) Distribution

```
⊢ ∀ p s.prob_geom_iter p n
= bind (prob_bern (1-p)) (λb. unit (b, (snd n) + 1))
⊢ ∀ p. prob_geom_loop p
= prob_while fst (prob_geom_iter p)
⊢ ∀ p. prob_geom p = bind (bind (unit (T, 1))
    (prob_geom_loop p)) (λs. unit (snd s - 1))
```

In the above algorithm, the state is a pair with the first component containing the last value of the Bernoulli random variable, and the second component containing the number of Bernoulli trials performed so far. This pair is initialized to (*True*, 1) and updated by the probabilistic while loop until the first component becomes *False*, at which point the algorithm terminates and outputs the second component (subtracting one because we do not count the final *False*).

The function, `prob_geom`, accepts a real number p, which represents the probability of success for the Geometric(p) random variable, and returns the corresponding Geometric random variable. It is important to note that p cannot be assigned a value equal to 0 as this will lead to a non-terminating while loop.

We verify the PMF property of the Geometric(p) random variable using the fact that the function `prob_geom` preserves *strong function independence* along with some theorems from probability and set theories in HOL.

Theorem 7 *PMF of Geometric random variable*

```
⊢ ∀ n p. 0 < p ∧ p ≤ 1
⇒ ℙ {s | fst (prob_geom p s) = (n + 1)} = p (1 - p)^n
```

6.2 Verification of Expectation of Geometric(p) Random Variable

The expectation property of Geometric(p) random variable can be stated in terms of Definitions 2 and 6 as follows.

Theorem 8 *Expectation of Geometric Random Variable*

$$\vdash \forall\, \mathtt{p.}\ 0 < \mathtt{p} \wedge \mathtt{p} \leq 1 \Rightarrow \mathtt{expec}\ (\lambda \mathtt{s.}\ \mathtt{prob\_geom}\ \mathtt{p}\ \mathtt{s}) = \tfrac{1}{\mathtt{p}}$$

Rewriting the above proof goal with the definition of expectation and simplifying using the PMF relation for the Geometric(p) random variable, given in Theorem 7, along with some arithmetic reasoning, we reach the following subgoal.

$$\lim_{k\to\infty}\left(\sum_{n=0}^{k}((n+1)p(1-p)^{n})\right) = \frac{1}{p} \tag{28}$$

Substituting $1-q$ for p and after some rearrangement of the terms, based on arithmetic reasoning, the above subgoal can be rewritten as follows.

$$\lim_{k\to\infty}\left(\sum_{n=0}^{k}((n+1)q^{n})\right) = \frac{1}{(1-q)^2} \tag{29}$$

Now, using the properties of summation of a real sequence in HOL, we proved the following relationship

$$\forall q\ k.\ \sum_{n=0}^{k}((n+1)q^{n}) = \sum_{n=0}^{k}\left(\sum_{i=0}^{k}q^{i} - \sum_{i=0}^{n}q^{i}\right) \tag{30}$$

which allows us to rewrite the subgoal under consideration, given in (29) as follows.

$$\lim_{k\to\infty}\left(\sum_{n=0}^{k}\left(\sum_{i=0}^{k}q^{i} - \sum_{i=0}^{n}q^{i}\right)\right) = \frac{1}{(1-q)^2} \tag{31}$$

The above subgoal can now be proved using the summation of a finite geometric series along with some properties of summation and limit of *real* sequences available in the *real* number theories in HOL. This also concludes the proof of Theorem 8 in HOL.

6.3 Verification of Variance of the Geometric(p) Random Variable

The variance property of Geometric(p) random variable can be stated in terms of Definitions 5 and 6 as follows.

Theorem 9 *Variance of Geometric(p) Random Variable*

$$\begin{aligned}&\vdash \forall\, \mathtt{p.}\ 0 < \mathtt{p} \wedge \mathtt{p} \leq 1\\ &\Rightarrow (\mathtt{variance}\ (\lambda \mathtt{s.}\ \mathtt{prob\_geom}\ \mathtt{p}\ \mathtt{s}) = \tfrac{1-\mathtt{p}}{\mathtt{p}^2})\end{aligned}$$

We utilize the variance property, given in Theorem 5, to verify Theorem 9. The foremost step in this regard is to verify the second moment relationship for the Geometric(p) random variable.

$$\forall\, \mathtt{p.}\ 0 < \mathtt{p} \wedge \mathtt{p} \leq 1 \Rightarrow \left(\mathtt{expec\_fn}(\lambda \mathtt{n.}\ \mathtt{n}^2(\lambda \mathtt{s.}\ \mathtt{prob\_geom}\ \mathtt{p}\ \mathtt{s})) = \frac{2}{p^2} - \frac{1}{p}\right) \tag{32}$$

Rewriting the above proof goal with the definition of function `expec_fn` and simplifying using the PMF relation of the Geometric random variable along with some properties from HOL *real* number theories, we reach the following subgoal.

$$\lim_{k\to\infty}\left(\sum_{n=0}^{k}((n+1)^2 p(1-p)^n)\right) = \frac{2}{p^2} - \frac{1}{p} \tag{33}$$

Now, substituting $1 - q$ for p and after some rearrangement of the terms, based on arithmetic reasoning, the above subgoal can be rewritten as follows.

$$\lim_{k\to\infty}\left(\sum_{n=0}^{k}((n+1)^2 q^n)\right) = \frac{2}{(1-q)^3} - \frac{1}{(1-q)^2} \tag{34}$$

Using the properties of summation of a real sequence in HOL, we prove the following

$$\forall q\, k. \sum_{n=0}^{k}((n+1)^2 q^n) = \sum_{n=0}^{k}\left((2n+1)\left(\sum_{i=0}^{k} q^i - \sum_{i=0}^{n} q^i\right)\right) \tag{35}$$

which allows us to rewrite the subgoal under consideration, given in (34), as follows.

$$\lim_{k\to\infty}\left(\sum_{n=0}^{k}\left((2n+1)\left(\sum_{i=0}^{k} q^i - \sum_{i=0}^{n} q^i\right)\right)\right) = \frac{2}{(1-q)^3} - \frac{1}{(1-q)^2} \tag{36}$$

The above subgoal can now be proved using the summation of a finite geometric series along with some properties of summation and limit of real sequences available in the *real* number theories in HOL. This concludes the proof of the second moment relation for the Geometric(p) random variable, which can now be used along with Theorems 5 and 8 and some arithmetic reasoning to prove Theorem 9 in HOL.

7 Coupon Collector's Problem

In this section, we utilize the HOL formalizations presented so far to verify the expectation and variance properties of the Coupon Collector's problem [28]. Firstly, we present a brief overview of the algorithm and present its formalization in HOL. This is followed by the details about the verification steps.

7.1 Formalization of Coupon Collector's Problem in HOL

The Coupon Collector's problem is motivated by "*collect all n coupons and win*" contests. Assuming that a coupon is drawn independently and uniformly at random from n possibilities, how many times do we need to draw new coupons until we find them all? This simple problem arises in many different scenarios. For example, suppose that packets are sent in a stream from source to destination host along a fixed path of routers. It is often the case that the destination host would like to know all routers that the stream of data has passed through. This may be done by appending the identification of each router to the packet header but this is not a

practical solution as usually we do not have this much room available. An alternate way of meeting this requirement is to store the identification of only one router, uniformly selected at random between all routers on the path, in each packet header. Then, from the point of view of the destination host, determining all routers on the path is like a Coupon Collector's problem.

The Coupon Collector's problem can be modeled as a probabilistic algorithm in higher-order logic. Let X be the number of trials until at least one of every type of coupon is obtained. Now, if X_i is the number of trials required to obtain the i^{th} coupon, while we had already acquired $i-1$ distinct coupons, then clearly $X = \sum_{i=1}^{n} X_i$. The advantage of breaking the random variable X into the sum of n random variables $X_1, X_2 \cdots, X_n$ is that each X_i can be modeled as a Geometric random variable, which enables us to represent the Coupon Collector's problem as a sum of Geometric random variables. Furthermore, the expectation and variance of this probabilistic algorithm can then be verified using the linearity of expectation and variance properties, which we have already verified in Sections 4 and 5, respectively.

The first step in the formalization of the Coupon Collector's problem is to define a list of Geometric random variables in order to model the X_i's mentioned above. It is important to note that the probability of success for each one of these Geometric random variables is different from one another and depends on the number of distinct coupons acquired so far. Since, every coupon is drawn independently and uniformly at random from the n possibilities, we can use the Uniform(n) random variable, which returns any positive integer in the interval [0, n-1] with the same probability, to model the probability of acquiring a new coupon or the probability of succuss for each one of the X_i's mentioned above. For this purpose, we identify distinct coupons in numerical order as they are acquired, i.e., the first coupon acquired is identified by number 0, the second by 1 and so on. Now, the probability of success for acquiring the k^{th} coupon, in a Coupon Collector problem with n distinct coupons, can be modeled as the probability of the event when the outcome of the Uniform(n) random variable is greater than or equal to $k-1$, where the Uniform(n) random variable is used to represent the coupon identification numbers. Based on the above proposition, the following higher-order-logic function generates the list of Geometric random variables that can be added to model the Coupon Collecting process of n distinct coupons.

Definition 7 Geometric Variable List for Coupon Collector's Problem

```
⊢ ∀ n. (geom_rv_lst 0 n = [])
∧ ∀ h t n. (geom_rv_lst (k+1) n
= (prob_geom ℙ{s | k ≤ fst (prob_unif n s) }) ::
     (geom_rv_lst k n))
```

In the above definition, the function `prob_unif` represents the HOL definition of the Uniform(n) random variable, which has been formalized in [21]. The function `geom_rv_lst` accepts two arguments; a positive integer `n` that represents the total number of distinct coupons and a positive integer, say `k`, that represents the number of distinct coupons acquired by the coupon collector out of the all possible `n` coupons at any particular instant. It returns, a list of Geometric random variables that can be

added to model the number of trials required to collect k coupons in the Coupon Collector's problem. The base case in the above recursive definition corresponds to the condition when the coupon collector does not have any coupon and thus the corresponding Geometric random variable list is empty. For the particular case when the variable k is assigned a value of 1, i.e., the coupon collector has acquired a single coupon out of the n possible distinct coupons, the function geom_rv_lst will return a list with one Geometric random variable element with probability of success equal to 1, since the probability that a Uniform(n) random variable would generate a number greater than or equal to 0 is 1. This is obviously the intended behavior since we are always certain to acquire a new coupon in the first trial of the Coupon Collector's problem. In a similar way, the function geom_rv_lst generates a list of k Geometric random variables which can be added to find the number of trials to acquire the first k distinct coupons.

Using the above definition along with the function sum_rv_lst, given in Definition 4, the Coupon Collector's problem can be represented now by the following probabilistic algorithm in HOL.

Definition 8 Probabilistic Algorithm for Coupon Collector's Problem

```
⊢ ∀ n. coupon_collector n = (sum_rv_lst (geo_rv_lst n n))
```

The function, coupon_collector, accepts a positive integer n that represents the total number of distinct coupons that are required to be collected. It returns the total number of trials required for collecting all the n coupons by adding the contents of the list of Geometric random variables modeled by the function geo_rv_lst with both arguments equal to n.

7.2 Verification of Expectation for the Coupon Collector's Problem

In this section, we verify that the expected value of acquiring all n distinct coupons in the Coupon Collector's problem can be represented by the following expression.

$$n \sum_{i=0}^{n} \frac{1}{i+1} \tag{37}$$

Sometimes, the mathematical expression of (37) is expressed in terms of the *harmonic number* as $nH(n)$, where $H(n) = \sum_{i=0}^{n} 1/(i+1)$. The expectation property of the Coupon Collector's problem, given in (37), can be stated using the functions coupon_collector and expec as a higher-order-logic theorem as follows.

Theorem 10 *Expectation of Coupon Collector's Problem*

```
⊢ ∀ n. expec (coupon_collector n) = n (∑ n i=0 1 / i+1)
```

We proceed with the verification of the above theorem by simplifying it with the definition of the function `coupon_collector`, given in Definition 8, and splitting the subgoal into two cases, i.e., when the value of `n` is 0 and when it is not 0.

$$\texttt{expec(sum\_rv\_lst (geo\_rv\_lst 0 0))} = \texttt{0} \tag{38}$$

$$\texttt{expec (sum\_rv\_lst (geo\_rv\_lst (n + 1) (n + 1)))} = \texttt{(n + 1)} \sum_{\texttt{i=0}}^{\texttt{n+1}} \frac{\texttt{1}}{\texttt{i + 1}} \tag{39}$$

The subgoal of (38) can be simply proved by using the definitions of the functions `expec`, `sum_rv_lst` and `geo_rv_lst` given in Definitions 2, 4 and 7, respectively, along with some arithmetic and probabilistic reasoning. On the other hand, we utilize the linearity of expectation property, given in Theorem 2, in order to rewrite the subgoal of (39) as follows

$$\sum_{\texttt{j=0}}^{\texttt{n+1}} \texttt{expec(el((n + 1) − (j + 1))(geo\_rv\_lst (n + 1) (n + 1)))}$$

$$= \texttt{(n + 1)} \sum_{\texttt{i=0}}^{\texttt{n+1}} \frac{\texttt{1}}{\texttt{i + 1}} \tag{40}$$

It is important to note that in order to use the linearity of expectation property, in the above step, we had to prove that all elements in the list (`geo_rv_lst (n + 1) (n + 1)`) preserve *strong function independence* and have *well-defined* expectations. Similarly, we also had to prove that the length of the list (`geo_rv_lst (n + 1) (n + 1)`) is equal to `n + 1`.

Next, we verify in HOL that any element *e* of the list `geo_rv_lst k n` can be mathematically expressed as follows.

$$\forall\, \texttt{e n k.}\ \texttt{(0 < k)} \wedge \texttt{(k} \leq \texttt{n)} \wedge \texttt{(e < k)}$$

$$\Rightarrow \texttt{(el e (geo\_rv\_lst k n) = prob\_geom (}\frac{\texttt{n} - \texttt{(k} - \texttt{(e + 1))}}{\texttt{n}}\texttt{))} \tag{41}$$

The above proof is based on the PMF property of the Uniform random variable, verified in [21], along with some arithmetic and probabilistic reasoning. Now, using the result of (41) along with some arithmetic reasoning, the subgoal of (40) can be expressed as follows

$$\sum_{\texttt{j=0}}^{\texttt{n+1}} \texttt{expec(prob\_geom (}\frac{\texttt{(n} - \texttt{j + 1))}}{\texttt{n + 1}}\texttt{))} = \texttt{(n + 1)} \sum_{\texttt{i=0}}^{\texttt{n+1}} \frac{\texttt{1}}{\texttt{i + 1}} \tag{42}$$

The expectation of the Geometric random variable in the above equation can be easily verified to be equal to $\frac{\texttt{n+1}}{\texttt{(n+1)}-\texttt{j}}$, using the results of Theorem 8. The substitution of the expectation value in the subgoal, given in (42), gives us the following expression

$$\sum_{\texttt{j=0}}^{\texttt{n+1}} \frac{\texttt{(n + 1)}}{\texttt{(n + 1)} - \texttt{j}} = \texttt{(n + 1)} \sum_{\texttt{i=0}}^{\texttt{n+1}} \frac{\texttt{1}}{\texttt{(i + 1)}} \tag{43}$$

which can be proved using properties of summation of a real sequence, given in the real number theories in HOL. This also concludes the proof for Theorem 10.

7.3 Verification of Variance Bound for the Coupon Collector's Problem

In this section, we verify the following upper bound on the variance of acquiring all n coupons in the Coupon Collector's problem

$$n^2 \sum_{i=0}^{n} \frac{1}{(i+1)^2} \tag{44}$$

This property can be expressed, using the functions `coupon_collector` and `variance`, as a higher-order-logic theorem as follows.

Theorem 11 *Variance Upper Bound of Coupon Collector's Problem*

$$\vdash \forall\ \mathtt{n.\ variance\ (coupon\_collector\ n)} \leq \mathtt{n}^2 \sum_{i=0}^{n} \left(\frac{1}{(\mathtt{i}+1)^2}\right)$$

The proof steps for the above theorem are quite similar to Theorem 10. The proof is based on the definition of the function `coupon_collector`, the linearity of variance property, given in Theorem 6, the PMF relation for the Uniform random variable and the variance relation of Geometric random variable, given in Theorem 9, along with some arithmetic and probabilistic reasoning.

Thus, we have been able to verify the expectation and variance properties of the Coupon Collector's problem with 100% precision, which is something that cannot be achieved by any existing computer based probabilistic analysis tool. It is also worth mentioning at this point that it is due to the formally verified linearity of expectation and variance properties that the complex task of verifying the expectation property and variance bound of the Coupon Collector's problem, which involves multiple random variables, was simply proved in HOL using summation over the expectation or variance of a single Geometric(p) random variable.

8 Conclusions

This paper presents the formalization of some expectation theory in higher-order-logic using the HOL theorem prover. The formalization can be utilized to verify statistical quantities, such as mean and variance, for probabilistic systems that can be modeled using discrete random variables in HOL. These statistical properties play a vital role in probabilistic analysis and thus the ability of their verification in a theorem-proving environment can be regarded as a significant step towards a complete theorem-proving based probabilistic analysis framework. Due to its inherent soundness, the theorem-proving based probabilistic analysis can prove to be quite useful for the performance and reliability optimization of safety critical and highly sensitive engineering and scientific applications.

In [19], we presented the higher-order-logic definition of an expectation function for discrete random variables that attain values in positive integers only and used this formalization to verify the linearity of expectation property. The current paper

extends that work by first presenting a formal definition of expectation for a function of a discrete random variable that can attain values in positive integers only. The main benefit of this new definition is that it allows us to formalize the mathematical concept of variance. This paper provides the formalization of variance and the verification of four classical properties of expectation and two classical properties of variance using the HOL theorem prover. The theorems corresponding to the classical properties of expectation and variance not only verify the correctness of our expectation and variance definitions but also play a vital role in conducting probabilistic analysis in a higher-order-logic theorem prover. For illustration purposes, we first utilize the formalization presented in this paper to verify the expectation and variance relations of the Geometric(p) random variable. Then, we formalized the Coupon Collector's problem as a probabilistic algorithm in HOL and verified its expectation and variance properties as well. To the best of our knowledge, this is the first time that an approach to verify both expectation and variance properties of probabilistic systems within a higher-order-logic theorem proving environment has been presented in the open literature.

The HOL formalization presented in this paper can be used to verify the expectation and variance properties of a number of other discrete random variables, e.g., Uniform, Bernoulli, Binomial and Poisson [22] and commercial computation problems, such as the Chinese appetizer and the Hat-Check problems [15]. As a potential case study for the formalization presented in this paper, we plan to conduct the analysis of the two versions of Quicksort algorithm [28] in HOL. This project will enable us to establish the distinction between the analysis of randomized algorithms and probabilistic analysis of deterministic algorithms within the HOL theorem prover.

An alternative approach that can be used to formalize the expectation of a random variable in higher-order logic is based on the mathematical concept of probability space. Since every random variable can be expressed as a real-valued function defined on the sample space, S, we can formalize expectation in terms of the probability space $(S, \mathfrak{F}, P)$, where $\mathfrak{F}$ is the sigma field of subsets of S, and P is the probability measure. The main benefit of this approach is that it leads to the formalization of the general definition of expectation, given in (1), for discrete random variables. On the other hand, in this approach we require the formal definition of a summation function for functions with domain in the sample space S. Such definition does not exist in the available HOL theories and thus needs to be formalized from scratch. It would be an interesting future work to formalize this summation and define a higher-order-logic definition of expectation based on the concept of probability space. A formal link may then be established between this generalized definition and the formal definition of expectation for discrete random variables with positive integers as their co-domain, presented in this paper. Such a relationship would further increase the confidence in our definitions.

Summarizing the experience of the work presented in this paper, we can say that formalizing mathematics in a mechanical system is a tedious work that requires deep understanding of both mathematical concepts and mechanical theorem-proving. We often came across proving subgoals that are commonly known to be true but their formal proofs could not be found even after browsing quite a few mathematical texts on that specific topic and thus we had to first develop a formal paper-pencil proof of these lemmas before translating them to HOL. The HOL automated reasoners

help somewhat in the proof process by automatically verifying some of the first-order-logic goals but most of the times we had to guide the tool by providing the appropriate rewriting and simplification rules. Thus, the HOL code for the formalization presented in this paper consists of more than 6000 lines. On the other hand, we found mechanical theorem-proving very efficient in book keeping. For example, it is very common to get confused with different variables and mathematical notations and make human errors when working with large paper-pencil proofs, which leads to the loss of a lot of effort, whereas in the case of mechanical theorem provers such problems do not exist. Another major advantage of mechanical theorem proving is that once the proof of a theorem is established, due to the inherent soundness of the approach, it is guaranteed to be valid and the proof can be readily accessed, contrary to the case of paper-pencil proofs where we have to explore the enormous amount of mathematical literature to find proofs. Thus, it can be concluded that mechanical theorem-proving is a tedious but promising field, which can help mathematicians to cope with the explosion in mathematical knowledge and to save mathematical concepts from corruption. Also, there are areas, such as security critical software, in military or medicine applications for example, where mechanical theorem-proving will soon become a dire need.

References

1. Adler, I., Ahn, H., Karp, R.M., Ross, S.M.: Coalescing times for IID random variables with applications to population biology. Random Struct. Algorithms **23**(2), 155–166 (2003)
2. Adler, M., Halperin, E., Karp, R.M., Vazirani, V.V.: A stochastic process on the hypercube with applications to peer-to-peer networks. In: Proc. 35th Annual ACM Symposium on Theory of Computing, pp. 575–584. ACM, New York (2003)
3. Audebaud, P., Paulin-Mohring, C.: Proofs of randomized algorithms in coq. In: Mathematics of Program Construction. LNCS, vol. 4014, pp 49–68. Springer, New York (2006)
4. Bratley, P., Fox, B.L., Schrage, L.E.: A Guide to Simulation. Springer, New York (1987)
5. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.P.: Model checking algorithms for continuous time markov chains. IEEE Trans. Softw. Eng. **29**(4), 524–541 (2003)
6. Bialas, J.: The σ-additive measure theory. J. Formaliz. Math. **2** (1990)
7. Billingsley, P.: Probability and Measure. Wiley, New York (1995)
8. Celiku, O.: Quantitative temporal logic mechanized in HOL. In: Theoretical Aspects of Computing. LNCS, vol. 3722, pp. 439–453. Springer, New York (2005)
9. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. MIT, Cambridge (2000)
10. Church, A.: A formulation of the simple theory of types. J. Symb. Log. **5**, 56–68 (1940)
11. DeGroot, M.: Probability and Statistics. Addison-Wesley, Reading (1989)
12. Devroye, L.: Non-Uniform Random Variate Generation. Springer, New York (1986)
13. Dimitrov, N.B., Plaxton, C.G.: Optimal cover time for a graph-based coupon collector process. In: Automata, Languages and Programming. LNCS, vol. 3580, pp. 702–716. Springer, New York (2005)
14. Gordon, M.J.C., Melham, T.F.: Introduction to HOL: A Theorem Proving Environment for Higher-Order Logic. Cambridge University Press, Cambridge (1993)
15. Grinstead, C.M., Snell, J.L.: Introduction to Probability. American Mathematical Society, Providence (1997)
16. Harrison, J.: Theorem Proving with the Real Numbers. Springer, New York (1998)
17. Hurd, J., McIver, A., Morgan, C.: Probabilistic Guarded Commands Mechanized in HOL. Theor. Comp. Sci. **346**, 96–112 (2005)
18. Hasan, O., Tahar, S.: Formalization of the continuous probability distributions. In: Automated Deduction. LNAI, vol. 4603, pp. 3–18. Springer, New York (2007)
19. Hasan, O., Tahar, S.: Verification of expectation properties for discrete random variables in HOL. In: Theorem Proving in Higher-Order Logics. LNCS, vol. 4732, pp. 119–134. Springer, New York (2007)

20. Hasan, O., Tahar, S.: Verification of probabilistic properties in HOL using the cumulative distribution function. In: Integrated Formal Methods. LNCS, vol. 4591, pp. 333–352. Springer, New York (2007)
21. Hurd, J.: Formal verification of probabilistic algorithms. PhD Thesis, University of Cambridge, Cambridge (2002)
22. Khazanie, R.: Basic Probability Theory and Applications. Goodyear, Los Angeles (1976)
23. Kwiatkowska, M., Norman, G., Parker, D.: Quantitative Analysis with the Probabilistic Model Checker PRISM. Electron Notes Theor Comp Sci Elsevier **153**(2), 5–31 (2005)
24. Levine, A.: Theory of Probability. Addison-Wesley Series in Behavioral Science, Quantitative Methods. Addison-Wesley, Reading (1971)
25. Leon Garcia, A., Widjaja, I.: Communication Networks: Fundamental Concepts and Key Architectures. McGraw-Hill, New York (2004)
26. Mao, W.: Modern Cryptography: Theory and Practice. Prentice Hall, Englewood Cliffs (2003)
27. Milner, R.: A theory of type polymorphism in programming. J. Comput. Syst. Sci. **17**, 348–375 (1977)
28. Mitzenmacher, M., Upfal, E.: Probability and Computing. Cambridge University Press, Cambridge (2005)
29. Nedzusiak, A.: σ-fields and Probability. J. Formaliz. Math. **1** (1989)
30. Paulson, L.C.: Isabelle: A Generic Theroem Prover, vol. 828 of LNCS. Springer, New York (1994)
31. Paulson, L.C.: ML for the Working Programmer. Cambridge University Press, Cambridge (1996)
32. Richter, S.: Formalizing integration theory, with an application to probabilistic algorithms. Diploma Thesis, Technische Universitat Munchen, Department of Informatics, Germany (2003)
33. Rutten, J., Kwaiatkowska, M., Normal, G., Parker, D.: Mathematical Techniques for Analyzing Concurrent and Probabilisitc Systems, Volume 23 of CRM Monograph Series. American Mathematical Society, Providence (2004)
34. Stirzaker, D.: Elementary Probability. Cambridge University Press, Cambridge (2003)
35. Sen, K., Viswanathan, M., Agha, G.: VESTA: a statistical model-checker and analyzer for probabilistic systems. In: Proc. IEEE International Conference on the Quantitative Evaluation of Systems, pp. 251–252. IEEE, Piscataway (2005)