Verification of Expectation Properties for Discrete Random Variables in HOL

Osman Hasan and Sofiène Tahar

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

Abstract. One of the most important concepts in probability theory is that of the expectation of a random variable, which basically summarizes the distribution of the random variable in a single number. In this paper, we develop the basic techniques for analyzing the expected values of discrete random variables in the HOL theorem prover. We first present a formalization of the expectation function for discrete random variables and based on this definition, the expectation properties of three commonly used discrete random variables are verified. Then, we utilize the definition of expectation in HOL to verify the linearity of expectation property, a useful characteristic to analyze the expected values of probabilistic systems involving multiple random variables. To demonstrate the usefulness of our approach, we verify the expected value of the Coupon Collector's problem within the HOL theorem prover.

1 Introduction

Probabilistic techniques are increasingly being used in the design and analysis of software and hardware systems, with applications ranging from combinatorial optimization and machine learning to communication networks and security protocols. The concept of a random variable plays a key role in probabilistic analysis. The sources of randomness associated with the system under test are modeled as random variables and then the performance issues are judged based on the properties of the associated random variables. One of the most important properties of random variables is their expectation or expected value. The 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.

Conventional simulation techniques are not capable of conducting the probabilistic analysis in a very efficient way. In fact, simulation based techniques require enormous amounts of numerical computations to generate meaningful results and can never guarantee exact answers. On the contrary, formal methods are capable of providing exact answers in this domain, if the probabilistic behavior can be modeled using a formalized semantics and we have the means to reason about probabilistic properties within a formalized framework.

Hurd's PhD thesis [9] is a pioneering work in regards to the modeling of probabilistic behavior in higher-order-logic. It presents an extensive foundational

K. Schneider and J. Brandt (Eds.): TPHOLs 2007, LNCS 4732, pp. 119–134, 2007.

development of probability, based on the mathematical measure theory, in the higher-order-logic (HOL) theorem prover. This formalization allows us to manipulate random variables and reason about their corresponding probability distribution properties in HOL. The probability distribution properties of a random variable, such as the *Probability Mass Function* (PMF), completely characterize the behavior of their respective random variables. It is frequently desirable to summarize the distribution of a random variable by its average or expected value rather than an entire function. For example, we are more interested in finding 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 this paper, we develop the basic techniques for analyzing the expected values of discrete random variables in the HOL theorem prover. To the best of our knowledge, this is a novelty that has not been presented in the open literature so far. We chose HOL for this purpose in order to build upon the verification framework proposed in [9]. Discrete random variables, such as the Uniform, Bernoulli, Binomial and Geometric, are widely used in a number of probabilistic analysis applications, e.g., analyzing the expected performances of algorithms [13] and efficiency of cryptographic protocols [12], etc. Most of these random variables are also *natural-valued*, i.e., they take on values only in the *natural* numbers, $\mathbb{N} = \{0, 1, 2, \cdots\}$. In order to speed up the formalization and verification process and to be able to target real life applications, we are going to concentrate in this paper on formalizing the expectation for this specific class of discrete random variables.

We first present a formal definition of expectation for *natural-valued* discrete random variables. This definition allows us to prove expectation properties for individual discrete random variables in HOL. To target the verification of expected values of probabilistic systems involving multiple random variables, we utilize our formal definition of expectation to prove the linearity of expectation property [10]. By this property, the expectation of the sum of random variables equals the sum of their individual expectations

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

where Ex denotes the expectation function. The linearity of expectation 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 the HOL theorem prover as a successful probabilistic analysis framework. In order to illustrate the practical effectiveness of the formalization presented in this paper, we analyze the expectation of the Coupon Collector's problem [13], a well know commercially used algorithm, within the HOL theorem prover. We first formalize the Coupon Collector's problem as a probabilistic algorithm using the summation of a list of Geometric random variables. Then, the linearity of expectation property is used to verify its corresponding expected value.

The rest of the paper is organized as follows: Section 2 gives a review of the related work. In Section 3, we summarize a general methodology for modeling and verification of probabilistic algorithms in the HOL theorem prover. Then, we present the formalization of the expectation function for *natural-valued* discrete random variables along with the verification of expectation properties of a few commonly used discrete distributions in Section 4. The results are found to be in good agreement with existing theoretical paper-and-pencil counterparts. Section 5 presents the verification of the linearity of expectation property. The analysis of the Coupon Collector's problem is presented in Section 6. Finally, Section 7 concludes the paper.

2 Related Work

Nędzusiak [14] and Bialas [2] were among the first ones to formalize some probability theory in higher-order-logic. Hurd [9] extended their work and developed a framework for the verification of probabilistic algorithms in HOL. 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. Building upon Hurd's formalization framework, we have been able to successfully verify the sampling algorithms of a few continuous random variables [7] and the classical cumulative distribution function properties [8], which play a vital role in verifying arbitrary probabilistic properties of both discrete and continuous random variables. The current paper also builds upon Hurd's framework and presents an infrastructure that can be used to verify expectation properties of *natural-valued* discrete random variables within a higher-order-logic theorem prover.

Richter [15] formalized a significant portion of the Lebesgue integration theory in higher-order-logic using Isabelle/Isar. In his PhD thesis, Richter linked the Lebesgue integration theory to probabilistic algorithms, developing upon Hurd's [9] framework, and presented the formalization of the first moment method. Due to its strong mathematical foundations, the Lebesgue integration theory can be used to formalize the expectation of most of the discrete and continuous random variables. Though, one of the limitations of this approach is the underlying complexity of the verification using interactive higher-order-logic theorem proving. It is not a straightforward task to pick a random variable and verify its expectation property using the formalized Lebesgue integration theory. Similarly, the analysis of probabilistic systems that involve multiple random variables becomes more difficult. On the other hand, our formalization approach for the expectation function, is capable of handling these kind of problems for discrete random variables, as will be demonstrated in Sections 4 and 6 of this paper, but is limited to discrete random variables only.

Expectation is one of the most useful tools in probabilistic analysis and therefore its evaluation with automated formal verification has also been explored in the probabilistic model checking community [1,16]. For instance, some probabilistic model checkers, such as PRISM [11] and VESTA [17], offer the capability of verifying expected values in a semi-formal manner. 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 these properties is, of course, dependent on the definitions of the rewards themselves and thus there is a significant risk of verifying false properties. On the other hand, there is no such risk involved in verifying the expectation properties using the proposed theorem proving based approach due to its inherent soundness.

Probabilistic model checking is capable of providing exact solutions to probabilistic properties in an automated way though; 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 [3], which is not an issue with the proposed theorem proving based probabilistic analysis approach.

3 Verifying Probabilistic Algorithms in HOL

This section presents the methodology, initially proposed in [9], for the formalization of probabilistic algorithms, which in turn can be used to represent random variables as well. The intent is to introduce the main ideas along with some notation that is going to be used in the next sections.

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 [9]. 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 states are infinite Boolean sequences.

```
⊢ ∀ a,s. unit a s = (a,s)
⊢ ∀ f,g,s. bind f g s = let (x,s')← f(s) ∈ g x s'
```

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).

The work in [9] 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 [18], 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

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

where the function fst selects the first component of a pair and $\{x|C(x)\}$ represents a set of all 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, called *strong function independence*, is introduced in [9] such that if $f \in$ indep, then f will be both measurable and independent. It has been shown in [9] 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.

The above mentioned methodology has been successfully used to verify the sampling algorithms of a few discrete random variables based on the corresponding probability distribution properties [9]. In the current paper, we further strengthen this particular higher-order-logic probabilistic analysis approach by presenting the formalization of an expectation function, which can be utilized to verify expectation properties for discrete random variables.

4 Expectation for Discrete Distributions

There are mainly two approaches that can be used to formalize the expected value of a random variable in a higher-order-logic theorem prover [10]. Since a random variable is 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. This approach leads to the theory of abstract Lebesgue integration. Richter [15] formalized a significant portion of the Lebesgue integration theory in higher-order-logic. Richter's formalization paves the way to manipulate expected values in a higher-order-logic theorem prover but leads to a very complex verification task when it comes to verifying expectation properties of probabilistic systems that involve multiple random variables.

An alternate approach for formalizing the expectation of a random variable is based on the fact that the probability distribution of a random variable X, defined on the real line, can be expressed in terms of the distribution function of X. As a consequence, the expected value of a *natural-valued* discrete random variable can be defined by referring to the distribution of the probability mass on the *real* line as follows

$$Ex[X] = \sum_{i=0}^{\infty} iPr(X = i) \tag{2}$$

where Pr denotes the probability. The above definition only holds if the summation, carried over all possible values of X, is convergent, i.e., $\sum_{i=0}^{\infty} |i| Pr(X = i) < \infty$.

We are going to follow the second approach. This decision not only simplifies the formalization task of expectation for discrete random variables considerably, when compared to the approach involving Lebesgue integration, but also aids in the verification of expectation properties of probabilistic systems that involve multiple random variables in a straight forward manner. The expected value of the *natural-valued* discrete random variables, given in Equation 2, can be formalized in HOL follows

Definition 1. *Expectation of natural-valued Discrete Random Variables*

```
⊢ ∀ X. expec X = suminf (λn. n ℙ{s | fst(X s) = n})
```

where `suminf` represents the HOL formalization of the infinite summation of a *real* sequence [6]. The function `expec` accepts the random variable X with data type $B^{\infty} \rightarrow (natural \times B^{\infty})$, and returns a *real* number.

Next, we build upon the above definition of expectation to verify the expectation properties of Uniform(n), Bernoulli(p) and Geometric(p) random variables in HOL. These random variables have been chosen in such a way that each one of them ranges over a different kind of set, in order to illustrate the generality of our definition of expectation.

4.1 Uniform(n) Random Variable

The Uniform(n) random variable assigns equal probability to each element in the set $\{0, 1, \cdots, (n-1)\}$ and thus ranges over a finite number of *natural* numbers. A sampling algorithm for the Uniform(n) can be found in [9], which has been proven correct by verifying the corresponding PMF property in HOL.

```
⊢ ∀ n m. m < n ⇒ ℙ{s | fst(prob_unif n s) = m} = 1/n
```

where prob_unif represents the higher-order-logic function for the Uniform(n) random variable. The first step towards the verification of the expectation property of discrete random variables that range over a finite number of *natural* numbers, say k, is to transform the infinite summation of Definition 1 to a finite summation over k values. This can be done in the case of the Uniform(n) random variable by using the above PMF property to prove the fact, for all values of n greater than 0, that the Uniform(n) random variable never acquires a value greater than or equal to n.

$\vdash \forall$ n m. (suc n) $\leq$ m $\Rightarrow \mathbb{P}\{$s | fst(prob_unif (suc n) s) = m$\}$ = 0

This property allows us to rewrite the infinite summation of Definition 1, for the case of the Uniform(n) random variable, in terms of a finite summation over n values using the HOL theory of *limit of a real sequence*. The expectation property can be proved now by induction over the variable n and simplifying the subgoals using some basic finite summation properties from the HOL theory of *real* numbers along with the PMF of the Uniform(n) random variable.

Theorem 1. *Expectation of Uniform(n) Random Variable*

$\vdash \forall$ n. expec (λs. prob_unif (suc n) s) = $\frac{n}{2}$

4.2 Bernoulli(p) Random Variable

Bernoulli(p) random variable models an experiment with two outcomes; success and failure, whereas the parameter p represents the probability of success. A sampling algorithm of the Bernoulli(p) random variable has been formalized in [9] as the function prob_bern such that it returns $True$ with probability p and $False$ otherwise. It has also been verified to be correct by proving the corresponding PMF property in HOL.

$\vdash \forall$ p. 0 $\leq$ p $\wedge$ p $\leq$ 1 $\Rightarrow \mathbb{P}\{$s | fst(prob_bern p s)$\}$ = p

The Bernoulli(p) random variable ranges over 2 values of *Boolean* data type. The expectation property of these kind of discrete random variables, which range over a finite number of values of a different data type than *natural* numbers, can be verified by mapping them to the *natural* line. In the case of Bernoulli(p) random variable, we redefined the function prob_bern such that it returns *natural* numbers 1 and 0 instead of the Boolean quantities $True$ and $False$ respectively, i.e., the range of the random variable was changed from *Boolean* data type to *natural* data type. It is important to note that this redefinition does not change the distribution properties of the given random variable. Now, the verification of the expectation can be handled using the same procedure used for the case of random variables that range over a finite number of *natural* numbers. In the case of Bernoulli(p) random variable, we were able replace the infinite summation of Definition 1 with the summation of the first two values of the corresponding *real* sequence using the HOL theory of *limit of a real sequence*. This substitution along with the PMF property of the Bernoulli(p) random variable and some properties from the HOL theories of *real* and *natural* numbers allowed us to verify the expectation of the Bernoulli(p) in HOL.

Theorem 2. *Expectation of Bernoulli(p) Random Variable*

$\vdash \forall$ `p.` $0 \leq$ `p` $\wedge$ `p` $\leq 1 \Rightarrow$ `expec (`λ`s. prob_bernN p s) =` p

where the function `prob_bernN` represents the Bernoulli(p) random variable that ranges over the *natural* numbers 0 and 1.

4.3 Geometric(p) Random Variable

Geometric(p) random variable can be defined as the index of the first success in an infinite sequence of Bernoulli(p) trials [4]. Therefore, the Geometric(p) distribution may be sampled by extracting random bits from the function `prob_bern`, explained in the previous section, and stopping as soon as the first *False* is encountered and returning the number of trials performed till this point. Thus, the Geometric(p) random variable ranges over a countably infinite number of *natural* numbers. This fact makes it different from other random variables that we have considered so far.

Based on the above sampling algorithm, we modeled the Geometric(p) random variable using the *probabilistic while loop* [9] in HOL as follows

Definition 2. *A Sampling Algorithm for Geometric(p) Distribution*

```
⊢ ∀ p s. prob_geom_iter p s = bind (prob_bern (1-p))
           (λb. unit (b, suc (snd s)))
⊢ ∀ 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))
```

It is important to note that p, which represents the probability of success for the Geometric(p) or the probability of obtaining *False* from the Bernoulli(p) random variable, cannot be assigned a value equal to 0 as this will lead to a non-terminating while loop. We verified that the function `prob_geom` preserves *strong function independence* using the HOL theories on probability. This result may be used along with the probability and set theories in HOL to verify the PMF property of the Geometric(p) random variable.

$\vdash \forall$ `n p.` $0 <$ `p` $\wedge$ `p` $\leq 1 \Rightarrow$

$\mathbb{P}\{$`s | fst(prob_geom p s) = (suc n)`$\}$ `=` $p(1-p)^n$

The expectation of Geometric(p) random variable can now be verified by first plugging the above PMF value into the definition of expectation and then using the following summation identity

$$\sum_{n=1}^{\infty} nx^{n-1} = \frac{1}{(1-x)^2} \tag{3}$$

where $0 \leq x < 1$. The proof task for this summation identity was quite involved and the HOL theories of *limit of a real sequence*, *real* and *natural* numbers were mainly used. The verified expectation property of Geometric(p) random variable in HOL is give below.

Theorem 3. *Expectation of Geometric(p) Random Variable*

```
⊢ ∀ n p. 0 < p ∧ p ≤ 1 ⇒
           expec (λs. prob_geom p s) = 1/p
```

5 Verification of Linearity of Expectation Property

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.

5.1 Two Random Variables

The linearity of expectation property can be defined for any two discrete random variables X and Y, according to Equation 1, as follows

$$Ex[X+Y] = Ex[X] + Ex[Y] \tag{4}$$

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)))
```

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.

Theorem 4. *`sum_two_rv` Preserves Strong Function Independence*

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

Now the linearity of expectation property for two discrete random variables can be stated using the `sum_two_rv` function as follows.

Theorem 5. *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
```

Proof: Rewriting the LHS with the definition of the functions `sum_two_rv` and `expec` and removing the monad notation, we obtain the following expression.

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

The set in the above expression can be expressed as the countable union of a sequence of events using the HOL theory of sets.

$$\lim_{k\to\infty} (\sum_{n=0}^{k} n \; \mathbb{P} \bigcup_{i\leq n} \{s \mid (fst(X\ s) \ = i) \ \wedge \ (fst(Y\ (snd(X\ s))) = n - i)\})$$

As all events in this sequence are mutually exclusive, the additive probability law given in the HOL theory of probability, can be used to simplify the expression as follows

$$\lim_{k\to\infty} (\sum_{n=0}^{k} n \; \sum_{i=0}^{n+1} \mathbb{P}\{s \mid (fst(X\ s) \ = i) \ \wedge \ (fst(Y\ (snd(X\ s))) = n - i)\})$$

Using the HOL theories of limit of a real sequence, real and natural number, the above expression can be rewritten as follows

$$\lim_{k\to\infty} (\sum_{a=0}^{k} \sum_{b=0}^{k} (a + b) \; \mathbb{P}\{s \mid (fst(X\ s) \ = a) \ \wedge \ (fst(Y\ (snd(X\ s))) = b)\})$$

Rearranging the terms based on summation properties given in the HOL theory of real numbers, we obtain the following expression.

$$\lim_{k\to\infty} (\sum_{a=0}^{k} \sum_{b=0}^{k} a \; \mathbb{P}\{s \mid (fst(X\ s) \ = a) \ \wedge \ (fst(Y\ (snd(X\ s))) = b)\}) \ +$$
$$\lim_{k\to\infty} (\sum_{b=0}^{k} \sum_{a=0}^{k} b \; \mathbb{P}\{s \mid (fst(X\ s) \ = a) \ \wedge \ (fst(Y\ (snd(X\ s))) = b)\})$$

The two terms in the above expression can now be proved to be equal to the expectation of random variables X and Y respectively, using Theorem 4 and HOL theory of probability, sets, real and and natural numbers. □

5.2 *n* Random Variables

The linearity of expectation property for two discrete random variables, verified in Theorem 5, can now be generalized to verify the linearity of expectation property for n discrete random variables, given in Equation 1, using induction techniques.

The first step in this regard is to define a function, similar to `sum_two_rv`, which models the summation of a list of n random variables.

Definition 4. *Summation of n Random Variables*

```
⊢ (sum_rv_lst [] = unit 0) ∧
∀ h t. (sum_rv_lst (h::t) =
         bind h (λa. bind (sum_rv_lst t)
         λb. unit (a + b)))
```

Next, we prove that the function `sum_rv_lst` preserves *strong function independence*, if all random variables in the given list preserve it. This property can be verified using the fact that the function `sum_rv_lst` accesses the infinite Boolean sequence using the `unit` and `bind` primitives only.

Theorem 6. *`sum_rv_lst` Preserves Strong Function Independence*

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

The predicate `mem` in the above definition returns *True* if its first argument is an element of the list that it accepts as the second argument.

Using induction on the list argument L of the function `sum_rv_lst` and simplifying the subgoals using Theorem 5, we proved, in HOL, that the expected value of the random variable modeled by the function `sum_rv_lst` exists if the expectation of all individual elements of its list argument exists. Here, by the existence of the expectation we mean that the infinite summation in the expectation definition converges.

Theorem 7. *The Expectation of `sum_rv_lst` Exists*

```
⊢ ∀ L. (∀ R. (mem R L) ⇒ R ∈ indep_fn ∧
                summable (λn. n ℙ{s | fst(R s) = n})) ⇒
          summable (λn. n ℙ{s | fst(sum_rv_lst L s) = n})
```

The linearity of expectation property for n discrete random variables can be proved now by applying induction on the list argument of the function `sum_rv_lst`, and simplifying the subgoals using Theorems 5, 6 and 7.

Theorem 8. *Linearity of Expectation for n Discrete Random Variables*

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

where the function `length` returns the length of its list argument and the function `el` accepts a *natural* number, say n, and a list and returns the n^{th} element of the given list. The term (`sum(m,n) f`), in the above theorem, models the summation of n values, corresponding to the arguments $m+n-1, \cdots, m+1, m$, of the real sequence f. Thus, the left-hand-side of Theorem 8 represents the expectation of the summation of a list, L, of random variables. Whereas, the right-hand-side represents the summation of the expectations of all elements in the same list, L, of random variables.

6 Coupon Collector's Problem

The Coupon Collector's problem [13] 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.

Our 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* equal to the summation $\sum_{i=1}^{n} 1/i$. The first step in this regard is to model the Coupon Collector's problem 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 $i-1$ different 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 of this probabilistic algorithm can now be verified using the linearity of expectation property.

We modeled the Coupon Collector's problem in HOL by identifying the coupons with unique *natural* numbers, such that the first coupon acquired by the coupon collector is identified as number 0 and after that each different kind of a coupon acquired with subsequent numbers in numerological order. The coupon collector saves these coupons in a list of *natural* numbers. The following function accepts the number of different coupons acquired by the coupon collector and generates the corresponding coupon collector's list.

Definition 5. *Coupon Collector's List*

```
⊢ (coupon_lst 0 = []) ∧
∀ n. (coupon_lst (suc n) = n :: (coupon_lst n))
```

The next step is to define a list of Geometric random variables which would 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 different coupons acquired so far. Since, every coupon is drawn independently and uniformly at random from the n possibilities and the coupons are identified with *natural* numbers, we can use the Uniform(n) random variable to model each trial of acquiring a coupon. Now we can define the probability of success for a particular Geometric random variable as the probability of the event when the Uniform(n) random variable generates a new value, i.e., a value that is not already present in the coupon collector's list. Using this probability of success, the following function generates the required list of Geometric random variables.

Definition 6. *Geometric Variable List for Coupon Collector's Problem*

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

where the functions prob_geom and prob_unif model the Geometric(p) and Uniform(n) random variables, respectively, which are given in Section 4. The function geom_rv_lst accepts two arguments; a list of *natural* numbers, which represents the coupon collector's list and a *natural* number, which represents the total number of coupons. It returns, a list of Geometric random variables that can be added up to model the coupon collecting process of the already acquired coupons in the given list. The base case in the above recursive definition corresponds to the condition when the coupon collector does not have any coupon and thus the probability of success, i.e., the probability of acquiring a new coupon, is 1.

Using the above definitions 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 7. *Probabilistic Algorithm for Coupon Collector's Problem*

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

The function coupon_collector accepts a *natural* number, say k, which represents the total number of different coupons that are required to be collected and has to be greater than 0. It returns the summation of k Geometric random variables that are used to model the coupon collecting process of acquiring k coupons. The expectation property of the Coupon Collector's problem can now be stated using the function coupon_collector and the function sum, which can be used to express the summation of n values, corresponding to the arguments $m+n-1, \cdots, m+1, m$, of the real sequence f as sum(m,n) f.

Theorem 9. *Expectation of Coupon Collector's Problem*

```
⊢ ∀ n. expec (coupon_collector (suc n)) =
        (suc n) (sum (0,(suc n)) (λi. 1/(suc i)))
```

Proof: The PMF property of the Uniform(n) random variable along with the HOL theories of sets and probability can be used to verify the following probabilistic quantity

$$\forall\ n.\ \mathbb{P}\{s \mid \neg(mem\ (fst(prob\_unif\ (n+1)\ s))\ L)\} = 1 - \frac{length\ L}{(n+1)}$$

for all lists of natural numbers, L, such that all the elements in L are less than (n+1) and none of them appears more than once. The coupon collector's list, modeled by the function `coupon_lst`, satisfies both of these characteristics for a given argument n. Therefore, the probability of succuss for a Geometric random variable that models the acquiring process of a new coupon when the coupon collectors list is exactly equal to L, in the probabilistic algorithm for the Coupon Collector's problem for (n+1) coupons, is $1 - \frac{length\ L}{(n+1)}$. The expectation of such a Geometric random variable can be verified to be equal to

$$\frac{n+1}{(n+1) - (length\ L)}$$

by the expectation property of Geometric(p) random variables, given in Theorem 3. Now, using the above result along with the linearity of expectation property and the strong function independence property of the Geometric random variables, the expectation of the sum of the list of Geometric random variables, given in the LHS of Theorem 9, can be expressed as the summation of the individual expectations of the Geometric random variables as follows

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

The following summation identity, which can be proved using the HOL theory of real and natural numbers, concludes the proof.

$$\forall\ n.\ \sum_{i=0}^{n} \frac{(n+1)}{(n+1) - i} = (n+1) \sum_{i=0}^{n} \frac{1}{(i+1)} \qquad \square$$

Theorem 9 can be used as a formal argument to support the claim that the expected number of trials required to obtain all n coupons is $n \sum_{i=1}^{n} 1/i$. Also, it is worth mentioning that it was due to the linearity of expectation property that the complex task of verifying the expectation property of the Coupon Collector's problem, which involves multiple random variables, was simply proved using the expectation property of a single Geometric(p) random variable.

7 Conclusions

In this paper, we presented some techniques for verifying the expectation properties of discrete random variables in HOL. Due to the formal nature of the models and the inherent soundness of the theorem proving systems, the analysis is guaranteed to provide exact answers, a novelty, which is not supported by most of the existing probabilistic analysis tools. This feature makes the proposed approach very useful for the performance and reliability optimization of safety critical and highly sensitive engineering and scientific applications.

We presented the formalization of expectation for *natural-valued* discrete random variables. This definition was used to verify the expected values of

Uniform(n), Bernoulli(p) and Geometric(p) random variables. These random variables are used extensively in the field of probabilistic analysis and thus their expectation properties can be reused in a number of different domains. Building upon our formal definition of expectation, we also verified a generalized version of the linearity of expectation property in HOL. In order to illustrate the practical effectiveness of our work, we presented the verification of the expectation property for the Coupon Collector's problem. To the best of our knowledge, this is the first time that the Coupon Collector problem has been analyzed within a mechanized theorem prover and the results are found to be in good agreement with existing theoretical paper-and-pencil counterparts.

The HOL theories presented in this paper can be used to verify the expectation properties of a number of other *natural-valued* random variables, e.g., Binomial, Logarithmic and Poisson [10] and commercial computation problems, such as the Chinese Appetizer and the Hat-Check problems [5]. A potential case study is to analyze the two versions of the Quicksort [13] and demonstrate the distinction between the analysis of randomized algorithms and probabilistic analysis of deterministic algorithms within the HOL theorem prover. As a next step towards a complete framework for the verification of randomized algorithms, we need to formalize the concepts of variance and higher moments. These bounds are the major tool for estimating the failure probability of algorithms. We can build upon the definition of expectation given in this paper to formalize these concepts within the HOL theorem prover. A very interesting future work could be to link the formal definition of expectation, presented in this paper, with the higher-order-logic formalization of Lebesgue integration theory [15], which would further strengthen the soundness of the definitions presented in this paper.

For our verification, we utilized the HOL theories of *Boolean algebra*, *sets*, *lists*, *natural* and *real* numbers, *limit of a real sequence* and *probability*. Our results can therefore be regarded as a useful indicator of the state-of-the-art in theorem proving. Based on this experience, we can say that formalizing mathematics in a mechanical system is a tedious work that requires deep understanding of both mathematical concepts and theorem-proving. The HOL automated reasoners aid 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. On the other hand, we found theorem-proving very helpful in book keeping. Another major advantage of 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 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 theorem-proving will soon become a dire need.

References

1. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.P.: Model Checking Algorithms for Continuous time Markov Chains. IEEE Trans. on Software Engineering 29(4), 524–541 (2003)
2. Bialas, J.: The σ-Additive Measure Theory. Journal of Formalized Mathematics 2 (1990)
3. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. MIT Press, Cambridge (2000)
4. DeGroot, M.: Probability and Statistics. Addison-Wesley, Reading (1989)
5. Grinstead, C.M., Snell, J.L.: Introduction to Probability. American Mathematical Society, Providence, RI (1997)
6. Harrison, J.: Theorem Proving with the Real Numbers. Springer, Heidelberg (1998)
7. Hasan, O., Tahar, S.: Formalization of the Continuous Probability Distributions. In: Conference on Automated Deduction. LNCS (LNAI), vol. 4603, pp. 3–18. Springer, Heidelberg (2007)
8. 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, Heidelberg (2007)
9. Hurd, J.: Formal Verification of Probabilistic Algorithms. PhD Thesis, University of Cambridge, Cambridge, UK (2002)
10. Khazanie, R.: Basic Probability Theory and Applications. Goodyear (1976)
11. Kwiatkowska, M., Norman, G., Parker, D.: Quantitative Analysis with the Probabilistic Model Checker PRISM. Electronic Notes in Theoretical Computer Science 153(2), 5–31 (2005)
12. Mao, W.: Modern Cryptography: Theory and Practice. Prentice-Hall, Englewood Cliffs (2003)
13. Mitzenmacher, M., Upfal, E.: Probability and Computing. Cambridge Press, Cambridge (2005)
14. Nedzusiak, A.: σ-fields and Probability. Journal of Formalized Mathematics 1 (1989)
15. Richter, S.: Formalizing Integration Theory, with an Application to Probabilistic Algorithms. Diploma Thesis, Technische Universität München, Department of Informatics, Germany (2003)
16. Rutten, J., Kwaiatkowska, M., Normal, G., Parker, D.: Mathematical Techniques for Analyzing Concurrent and Probabilisitc Systems. CRM Monograph Series, vol. 23. American Mathematical Society, Providence, RI (2004)
17. Sen, K., Viswanathan, M., Agha, G.: VESTA: A Statistical Model-Checker and Analyzer for Probabilistic Systems. In: IEEE International Conference on the Quantitative Evaluation of Systems, Washington, DC, USA, pp. 251–252. IEEE Computer Soceity Press, Los Alamitos (2005)
18. Williams, D.: Probability with Martingales. Cambridge Press, Cambridge (1991)