Formalization of Continuous Probability Distributions

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. Continuous probability distributions are widely used to mathematically describe random phenomena in engineering and physical sciences. In this paper, we present a methodology that can be used to formalize any continuous random variable for which the inverse of the cumulative distribution function can be expressed in a closed mathematical form. Our methodology is primarily based on the Standard Uniform random variable, the classical cumulative distribution function properties and the Inverse Transform method. The paper includes the higher-order-logic formalization details of these three components in the HOL theorem prover. To illustrate the practical effectiveness of the proposed methodology, we present the formalization of Exponential, Uniform, Rayleigh and Triangular random variables.

1 Introduction

Theorem proving [7] is an interactive verification approach that can be used to prove mathematical theorems in a computer based environment. Due to its inherent soundness, theorem proving is capable of providing precise answers and is thus more powerful than testing or simulation-based system analysis techniques. In this paper, we propose to perform probabilistic analysis within the environment of a higher-order-logic theorem prover in order to overcome the inaccuracy and enormous CPU time requirement limitations of state-of-the-art simulation based probabilistic analysis approaches.

The foremost criteria for constructing a theorem-proving based probabilistic analysis framework is to be able to formalize the commonly used random variables in higher-order logic. This formalized library of random variables can be utilized to express random behavior exhibited by systems and the corresponding probabilistic properties can then be proved within the sound environment of an interactive theorem prover. Random variables are basically functions that map random events to numbers and they can be expressed in a computerized environment as probabilistic algorithms. In his PhD thesis, Hurd [14] presented a methodology for the verification of probabilistic algorithms in the higher-order-logic (HOL) theorem prover [8]. Hurd was also able to formalize a few discrete random variables and verify their corresponding distribution properties. On the

F. Pfenning (Ed.): CADE 2007, LNAI 4603, pp. 3–18, 2007.

other hand, to the best of our knowledge, no higher-order-logic formalization of continuous random variables exists in the open literature so far.

In this paper, we propose a methodology for the formalization of continuous random variables in HOL. Our methodology utilizes Hurd's formalization framework and is based on the concept of the nonuniform random number generation [5], which is the process of obtaining random variates of arbitrary distributions using a Standard Uniform random number generator. The main advantage of this approach is that we only need to formalize one continuous random variable from scratch, i.e., the Standard Uniform random variable, which can be used to model other continuous random variables by formalizing the corresponding nonuniform random number generation method.

Based on the above methodology, we now present a framework, illustrated in Figure 1, for the formalization of continuous probability distributions for which the inverse of the *Cumulative Distribution Function* (CDF) can be represented in a closed mathematical form. Firstly, we formally specify the Standard Uniform random variable and verify its correctness by proving the corresponding CDF and measurability properties. The next step is the formalization of the CDF and the verification of its classical properties. Then we formally specify the mathematical concept of the inverse function of a CDF. This formal specification, along with the formalization of the Standard Unform random variable and the CDF properties, can be used to formally verify the correctness of the *Inverse Transform Method* (ITM) [5], 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. At this point, the formalized Standard Uniform random variable can be used to formally specify any such continuous random variable and its corresponding CDF can be verified using the ITM.

The rest of the paper is organized as follows: In Section 2, we briefly review Hurd's methodology for the verification of probabilistic algorithms in HOL. The next three sections of this paper present the HOL formalization of the three major steps given in Figure 1, i.e., the Standard Uniform random variable, the CDF and the ITM. In Section 6, we utilize the proposed framework of Figure

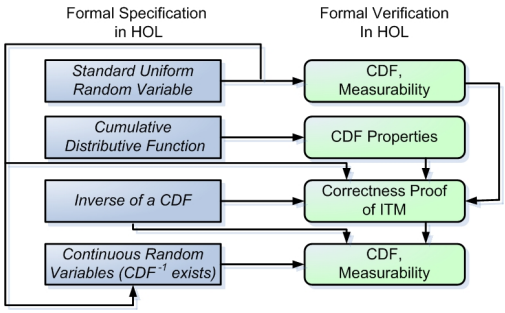

Fig. 1. Proposed Formalization Framework

1 to formalize the Exponential, Uniform, Rayleigh and Triangular random variables. In Section 7, we discuss potential probabilistic analysis applications for the formalized continuous random variables. A review of related work in the literature is given in Section 8 and we finally conclude the paper in Section 9.

2 Verifying Probabilistic Algorithms in HOL

In this section, we provide an overview of Hurd's methodology [14] for the 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 sections.

Hurd [14] proposed to formalize the probabilistic algorithms 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. 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

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

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

$\vdash$ `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 = let (x,s')← f(s) in 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`).

Hurd [14] also formalized some mathematical measure theory in HOL in order 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

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

where the function `fst` selects the first component of a pair. In Hurd's formalization of probability theory, a set of infinite Boolean sequences, S, is said to be measurable if and only if it is in $\mathcal{E}$, i.e., S $\in \mathcal{E}$. Since the probability measure $\mathbb{P}$ is only defined on sets in $\mathcal{E}$, it is very important to prove that sets that arise in verification are measurable. Hurd [14] showed that a function is guaranteed to be measurable if it accesses the infinite Boolean sequence using only the `unit`, `bind` and `sdest` primitives and thus leads to only measurable sets.

Hurd formalized a few discrete random variables and proved their correctness by proving the corresponding *Probability Mass Function* (PMF) properties [14]. The algorithms for these discrete random variables are either guaranteed to terminate or satisfy probabilistic termination, meaning that the probability that the algorithm terminates is 1. Thus, they can be expressed using Hurd's methodology by either well formed recursive functions or the *probabilistic while loop* [14]. On the other hand, the implementation of continuous random variables requires non-terminating programs and hence calls for a different approach.

3 Formalization of the Standard Uniform Distribution

In this section, we present the formalization of the Standard Uniform distribution that is the first step in the proposed methodology for the formalization of continuous probability distributions as shown in Figure 1. The Standard Uniform random variable can be characterized by the CDF as follows:

$$Pr(X \leq x) = \begin{cases} 0 \text{ if } x < 0; \\ x \text{ if } 0 \leq x < 1; \\ 1 \text{ if } 1 \leq x. \end{cases} \tag{1}$$

3.1 Formal Specification of Standard Uniform Random Variable

The Standard Uniform random variable can be formally expressed in terms of an infinite sequence of random bits as follows [11]

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

where, X_k denotes the outcome of the k^{th} random bit; $true$ or $false$ represented as 1 or 0, respectively. The mathematical expression of Equation (2) can be formalized in the HOL theorem prover in two steps. The first step is to define a discrete Standard Uniform random variable that produces any one of the equally spaced 2^n dyadic rationals, of the form $\frac{i}{2^n}$ $(0 \leq i \leq 2^n)$, in the interval $[0, 1-(\frac{1}{2})^n]$ with the same probability $(\frac{1}{2})^n$ using Hurd's methodology.

```
Definition 3.1:
        std_unif_disc: (num → (num → bool) → (real × (num → bool)))
        ⊢ (std_unif_disc 0 = unit 0) ∧
        ∀ n. (std_unif_disc (suc n) =
                bind (std_unif_disc n) (λm. bind sdest
                    (λb. unit (if b then ((1/2)^(n+1) + m) else m))))
```

The function *std_unif_disc* allows us to formalize the *real* sequence of Equation (2) in the HOL theorem prover. Now, the formalization of the mathematical concept of limit of a *real* sequence in HOL [10] can be used to formally specify the Standard Uniform random variable of Equation (2) as follows

```
Definition 3.2:
        std_unif_cont: ((num → bool) → real)
        ⊢ ∀ s. std_unif_cont s = lim (λn. fst (std_unif_disc n s))
```

where, *lim* is the HOL function for the limit of a *real* sequence [10].

3.2 Formal Verification of Standard Uniform Random Variable

The formalized Standard Uniform random variable, *std_unif_cont*, can be verified to be correct by proving its CDF to be equal to the theoretical value given in Equation (1) and its *Probability Mass Function* (PMF) to be equal to 0, which is an intrinsic characteristic of all continuous random variables. For this purpose, it is very important to prove that the sets $\{s \mid std\_unif\_cont\ s \leq x\}$ and $\{s \mid std\_unif\_cont\ s = x\}$ arising in this verification are measurable. The fact that the function *std_unif_disc* accesses the infinite Boolean sequence using only the `unit`, `bind` and `sdest` primitives can be used to prove

```
Lemma 3.1:
        ⊢ ∀ x n. {s | fst (std_unif_disc n s) ≤ x} ∈ 𝓔 ∧
                 {s | fst (std_unif_disc n s)  = x} ∈ 𝓔
```

On the other hand, the definition of the function *std_unif_cont* involves the *lim* function and thus the corresponding sets cannot be proved to be measurable in a very straightforward manner. Therefore, in order to prove this, we leveraged the fact that each set in the sequence of sets $(\lambda n.\{s \mid fst(std\_unif\_disc\ n\ s) \leq x\})$ is a subset of the set before it. In other words, this sequence of sets is a monotonically decreasing sequence. Thus, the countable intersection of all sets in this sequence can be proved to be equal to the set $\{s \mid std\_unif\_cont\ s \leq x\}$

```
Lemma 3.2:
        ⊢ ∀ x. {s | std_unif_cont s ≤ x} =
               ⋂_n (λ n. {s | fst (std_unif_disc n s) ≤ x})
```

Now the set $\{s \mid std\_unif\_cont\ s \leq x\}$ can be proved to be measurable since $\mathcal{E}$ is closed under countable intersections [14] and all sets in the sequence

$(\lambda n.\{s \mid fst(std\_unif\_disc\ n\ s) \leq x\})$ are measurable according to Lemma 1. Using a similar reasoning, the set $\{s \mid std\_unif\_cont\ s = x\}$ can also be proved to be measurable.

```
Theorem 3.1:
        ⊢ ∀ x. {s | std_unif_cont s ≤ x} ∈ 𝓔 ∧
               {s | std_unif_cont s  = x} ∈ 𝓔
```

Theorem 3.1 can now be used along with the *real* number theories [10] to verify the correctness of the function *std_unif_cont* in the HOL theorem prover by proving its *Probability Mass Function* (PMF) and CDF properties [11].

```
Theorem 3.2:
        ⊢ ∀ x. ℙ{s | std_unif_cont s = x} = 0 ∧
               ℙ{s | std_unif_cont s ≤ x} =
                if (x < 0) then 0 else (if (x < 1) then x else 1)
```

4 Formalization of the Cumulative Distribution Function

In this section, we present the verification of classical CDF properties in the HOL theorem prover, which is the second step in the proposed methodology.

4.1 Formal Specification of CDF

The CDF of a random variable, R, is defined by $F_R(x) = Pr(R \leq x)$ for any *real* number x, where Pr represents the probability. It follows from this definition that the CDF can be formally specified in HOL by a higher-order-logic function that accepts a random variable and a *real* argument and returns the probability of the event when the given random variable is less than or equal to the value of the given *real* number.

```
Definition 4.1:
        cdf: (((num → bool) → real) → real → real)
        ⊢ ∀ R x. cdf R x = ℙ {s | R s ≤ x}
```

4.2 Formal Verification of CDF Properties

Using the formal specification of the CDF, we are able to verify classical CDF properties [16] (details are given below) in HOL. The formal proofs for these properties not only ensure the correctness of our CDF specification but also play a vital role in proving the correctness of the ITM as will be discussed in Section 5. The formal proofs of these properties are established using the HOL set, measure, probability [14] and *real* number [10] theories and under the assumption that the set $\{s \mid R\ s \leq x\}$, where R represents the random variable under consideration, is measurable for all values of x. The details of the HOL verification steps for these properties can be found in [12].

CDF Bounds. $(0 \leq F_R(x) \leq 1)$
This property states that if we plot the CDF against its *real* argument x, then the graph of the CDF is between the two horizontal lines $y = 0$ and $y = 1$.

```
Theorem 4.1:
       ⊢ ∀ R x. (0 ≤ cdf R x) ∧ (cdf R x ≤ 1)
```

CDF is Monotonically Increasing. (*if* $a < b$, *then* $F_R(a) \leq F_R(b)$)
For all *real* numbers a and b, if a is less than b, then the CDF value of a random variable, R, at a can never exceed the CDF value of R at b.

```
Theorem 4.2:
       ⊢ ∀ R a b. a < b ⇒ (cdf R a ≤ cdf R b)
```

Interval Probability. (*if* $a < b$ *then* $Pr(a < R \leq b) = F_R(b) - F_R(a)$)
This property is very useful for evaluating the probability of a random variable, R, lying in any given interval (a,b] in terms of its CDF.

```
Theorem 4.3:
       ⊢ ∀ R a b. a < b ⇒ (ℙ {s | (a < R s) ∧ (R s ≤ b)} =
                           cdf R b - cdf R a)
```

CDF at Positive Infinity. $(\lim_{x \to \infty} F_R(x) = 1;$ *that is*, $F_R(\infty) = 1)$
This property states that the value of the CDF for any given random variable, R, always tends to 1 as its *real* argument approaches positive infinity.

```
Theorem 4.4:
       ⊢ ∀ R. lim (λ n. cdf R (&n)) = 1
```

where *lim* M represents the formalization of the limit of a *real* sequence M (i.e., $\lim_{n \to \infty} M(n) = lim\ M$) [10] and "&" represents the conversion function from *natural* to *real* numbers in HOL.

CDF at Negative Infinity. $(\lim_{x \to -\infty} F_R(x) = 0;$ *that is*, $F_R(-\infty) = 0)$
This property states that the value of the CDF for any given random variable, R, always tends to 0 as its *real* argument approaches negative infinity.

```
Theorem 4.5:
       ⊢ ∀ R. lim (λ n. cdf R (-&n)) = 0
```

CDF is Continuous from the Right. $(\lim_{x \to a^+} F_R(x) = F_R(a))$
In this property, $\lim_{x \to a^+} F_R(x)$ is defined as the limit of $F_R(x)$ as x tends to a through values greater than a. Since F_R is monotone and bounded, this limit always exists.

```
Theorem 4.6:
       ⊢ ∀ R a. lim (λ n. cdf R (a + 1/&(n+1))) = cdf R a
```

CDF Limit from the Left. ($\lim_{x \to a^-} F_R(x) = Pr(R < a)$)
In this property, $\lim_{x \to a^-} F_R(x)$ is defined as the limit of $F_R(x)$ as x tends to a through values less than a.

```
Theorem 4.7:
        ⊢ ∀ R a. lim (λ n. cdf R (a - 1/&(n+1))) = ℙ {s | (R s < a})
```

5 Formalization of the Inverse Transform Method

In this section, we present the formal specification of the inverse function for a CDF and the verification of the ITM in HOL. It is the third step in the proposed methodology for the formalization of continuous probability distributions as shown in Figure 1. The ITM is based on the following proposition [21].

> *Let U be a Standard Uniform random variable. For any continuous CDF F, the random variable X defined by* $X = F^{-1}(U)$ *has CDF F, where* $F^{-1}(U)$ *is defined to be the value of x such that* $F(x) = U$.

Mathematically,

$$Pr(F^{-1}(U) \leq x) = F(x) \tag{3}$$

5.1 Formal Specification of the Inverse Transform method

We define the inverse function for a CDF in HOL as a predicate *inv_cdf_fn*, which accepts two functions, f and g, of type $(real \rightarrow real)$ and returns true if and only if the function f is the inverse of the CDF g according to the above proposition.

```
Definition 5.1:
        inv_cdf_fn: ((real → real) → (real → real) → bool)
        ⊢ ∀ f g. inv_cdf_fn f g =
            (∀x. (0 < g x ∧ g x < 1) ⇒ (f (g x) = x) ∧
            (∀x. 0 < x ∧ x < 1 ⇒ (g (f x) = x))) ∧
            (∀x. (g x = 0) ⇒ (x ≤ f (0))) ∧
            (∀x. (g x = 1) ⇒ (f (1) ≤ x))
```

The predicate *inv_cdf_fn* considers three separate cases, the first one corresponds to the strictly monotonic region of the CDF, i.e., when the value of the CDF is between 0 and 1. The next two correspond to the flat regions of the CDF, i.e., when the value of the CDF is either equal to 0 or 1, respectively. These three cases cover all possible values of a CDF since according to Theorem 4.1 the value of CDF can never be less than 0 or greater than 1.

The inverse of a function f, $f^{-1}(u)$, is defined to be the value of x such that $f(x) = u$. More formally, if f is a one-to-one function with domain X and range Y, its inverse function f^{-1} has domain Y and range X and is defined by

$f^{-1}(y) = x \Leftrightarrow f(x) = y$, for any y in Y. The composition of inverse functions yields the following result.

$$f^{-1}(f(x)) = x \; for \; all \; x \in X, \quad f(f^{-1}(x)) = x \; for \; all \; x \in Y \tag{4}$$

We use the above characteristic of inverse functions in the predicate *inv_cdf_fn* for the strictly monotonic region of the CDF as the CDF in this region is a one-to-one function. On the other hand, the CDF is not injective when its value is either equal to 0 or 1. Consider the example of some CDF, F, which returns 0 for a *real* argument a. From Theorems 4.1 and 4.2, we know that the CDF F will also return 0 for all *real* arguments that are less than a as well, i.e., $\forall x.\ x \leq a \Rightarrow F(x) = 0$. Therefore, no inverse function satisfies the conditions of Equation (4) for the CDF in these flat regions. When using the paper-and-pencil proof approach, this issue is usually resolved by defining the inverse function of a CDF in such a way that it returns the infimum (inf) of all possible values of the *real* argument for which the CDF is equal to a given value, i.e., $f^{-1}(u) = inf\{x|f(x) = u\}$ [5], where f represents the CDF. Even though this approach has been shown to analytically verify the correctness of the ITM [5], it was not found to be sufficient enough for a formal definition in our case. This is due to the fact that in order to simplify the formalization task, Hurd [14] used the standard *real* numbers $\mathbb{R}$, formalized in HOL by Harrison [10], rather than the extended real numbers $\overline{\mathbb{R}} = \mathbb{R} \bigcup \{-\infty, +\infty\}$ to formalize the mathematical measure theory. Thus, if the inf function is used to define the inverse function, then the problem arises for the case when the value of the CDF is equal to 0. For this case, the set $\{x|f(x) = 0\}$ becomes unbounded at the lower end because of the CDF property given in Theorem 4.5 and thus the value of the inverse function becomes undefined. In order to overcome this problem, we used two separate cases for the two flat regions in the predicate *inv_cdf_fn*. According to this definition the inverse function of a CDF is a function that returns the maximum value of all arguments for which the CDF is equal to 0 and the minimum value of all arguments for which the CDF is equal to 1.

5.2 Formal Verification of the Inverse Transform Method

The correctness theorem for the ITM can be expressed in HOL as follows:

```
Theorem 5.1:
      ⊢ ∀ f g x. (is_cont_cdf_fn g) ∧ (inv_cdf_fn f g) ⇒
                 (ℙ {s | f (std_unif_cont s) ≤ x} = g x)
```

The antecedent of the above implication checks if f is a valid inverse function of a continuous CDF g. The predicate *inv_cdf_fn* has been described in the last section and ensures that the function f is a valid inverse of the CDF g. The predicate *is_cont_cdf_fn* accepts a *real*-valued function, g, of type $(real \rightarrow real)$ and returns true if and only if it represents a continuous CDF. A *real*-valued function can be characterized as a continuous CDF if it is a continuous function and satisfies the CDF properties given in Theorems 4.2, 4.4 and 4.5. Therefore, the predicate *is_cont_cdf_fn* is defined in HOL as follows:

```
Definition 5.2:
        is_cont_cdf_fn: ((real → real) → bool)
        ⊢ ∀ g. is_cont_cdf_fn g =
                (∀ x. (λx. g x) contl x) ∧
                (∀ a b. a < b ⇒ g a ≤ g b) ∧
                (lim (λ n. g (-&n)) = 0) ∧
                (lim (λ n. g (&n)) = 1)
```

where ($\forall\ x.f\ contl\ x$) represents the HOL definition for a continuous function [10] such that the function f is continuous for all x.

The conclusion of the implication in Theorem 5.1 represents the correctness proof of the ITM given in Equation (3). The function std_unif_cont in this theorem is the formal definition of the Standard Uniform random variable, described in Section 3. Theorem 3.2 can be used to reduce the proof goal of Theorem 5.1 to the following subgoal:

```
Lemma 5.1:
        ⊢ ∀ f g x. (is_cont_cdf_fn g) ∧ (inv_cdf_fn f g) ⇒
                   (ℙ {s | f (std_unif_cont s) ≤ x} =
                    ℙ {s | std_unif_cont s ≤ g x})
```

Next, we use the theorems of Section 3 and 4 along with the formalized measure and probability theories in HOL [14] to prove the measurability of the sets that arise in this verification, i.e., they are in $\mathcal{E}$.

```
Lemma 5.2:
        ⊢ ∀ f g x. (is_cont_cdf_fn g) ∧ (inv_cdf_fn f g) ⇒
                   ({s | f (std_unif_cont s) ≤ x} ∈ ℰ) ∧
                   ({s | std_unif_cont s) ≤ g x} ∈ ℰ) ∧
                   ({s | f (std_unif_cont s) = x} ∈ ℰ)
```

Lemma 5.1 can now be proved using Lemma 5.2, the theorems from Section 3 and 4 and Hurd's formalization of probability theory in HOL. The details of the HOL verification steps can be found in [13]. The main advantage of the formally verified ITM (i.e., Theorem 5.1) is the simplification of the verification task of proving the CDF property of a random variable. Originally the verification of the CDF property involves a reasoning based on the measure, probability and *real* number theories and the theorems related to the Standard Uniform random variable. Using the ITM, the CDF verification goal can be broken down to two simpler sub-goals, which only involve a reasoning based on the *real* number theory; i.e., (1) verifying that a function g, of type ($real \rightarrow real$), represents a valid CDF and (2) verifying that another function f, of type ($real \rightarrow real$), is a valid inverse of the CDF g.

6 Formalization of Continuous Probability Distributions

In this section, we present the formal specification of four continuous random variables; Uniform, Exponential, Rayleigh and Triangular and verify the

correctness of these random variables by proving their corresponding CDF properties in the HOL theorem prover.

6.1 Formal Specification of Continuous Random Variables

All continuous random variables for which the inverse of the CDF exists in a closed mathematical form can be expressed in terms of the Standard Uniform random variable according to the ITM proposition given in Section 5. We selected four such commonly used random variables, i.e., Exponential, Uniform, Rayleigh and Triangular, which are formally expressed in terms of the formalized Standard Uniform random variable (*std_unif_cont*) in Table 1 as HOL functions *exp_rv*, *uniform_rv*, *rayleigh_rv* and *triangular_rv*, respectively. The functions *ln*, *exp* and *sqrt* in Table 1 are the HOL functions for *logarithm*, *exponential* and *square root*, respectively [10].

Table 1. Continuous Random Variables (for which CDF^{-1} exists)

Distribution	**CDF**	**Formalized Random Variable**
Exponential(l)	0 if $x \leq 0$; $1 - exp^{-lx}$ if $0 < x$.	$\vdash \forall s\ l.\ exp\_rv\ l\ s\ =$ $-\frac{1}{l} ln(1 - std\_unif\_cont\ s)$
Uniform(a, b)	0 if $x \leq a$; $\frac{x-a}{b-a}$ if $a < x \leq b$; 1 if $b < x$.	$\vdash \forall s\ l.\ uniform\_rv\ a\ b\ s\ =$ $(b - a)(std\_unif\_cont\ s) + a$
Rayleigh(l)	0 if $x \leq 0$; $1 - exp^{\frac{-x^2}{2l^2}}$ if $0 < x$.	$\vdash \forall s\ l.\ rayleigh\_rv\ l\ s\ =$ $l * sqrt(-2ln(1 - std\_unif\_cont\ s))$
Triangular($0, a$)	0 if $x \leq 0$; $(\frac{2}{a}(x - \frac{x^2}{2a}))$ if $x < a$; 1 if $a \leq x$.	$\vdash \forall s\ a\ .\ triangular\_rv\ l\ s\ =$ $a(1 - sqrt(1 - std\_unif\_cont\ s))$

6.2 Formal Verification of Continuous Random Variables

The first step in verifying the CDF property of a continuous random variable, using the correctness theorem of the ITM, is to express the given continuous random variable as $F^{-1}(U\ s)$, where F^{-1} is a function of type $(real \rightarrow real)$ and U represents the formalized Standard Uniform random variable. For example, the Exponential random variable given in Table 1 can be expressed as $(\lambda x. -\frac{1}{l} * ln(1 - x))(std\_unif\_cont\ s)$. Similarly, we can express the CDF of the given random variable as $F(x)$, where F is a function of type $(real \rightarrow real)$ and x is a *real* data type variable. For example, the CDF of the Exponential random variable can be expressed as $(\lambda x.\ if\ x \leq 0\ then\ 0\ else\ 1 - exp^{-\lambda x}))\ x$.

The next step is to prove that the function F defined above represents a valid continuous CDF and the function F^{-1} is a valid inverse function of the CDF F. The predicates *is_cont_cdf_fn* and *inv_cdf_fn*, defined in Section 5, can be used for this verification and the corresponding theorems for the Exponential random variable are given below

```
Lemma 6.1:
       ⊢ ∀ l. is_cont_cdf_fn
              (λx. if x ≤ 0 then 0 else (1 - exp (-l * x)))

Lemma 6.2:
       ⊢ ∀ l. inv_cdf_fn (λ x. -1/l * ln (1 - x))
              (λx. if x ≤ 0 then 0 else (1 - exp (-l * x)))
```

The above lemmas along with Theorem 5.1 and Lemma 5.2 can be used to verify the CDF and the measurability of the sets corresponding to the given continuous random variable, respectively. These theorems for the Exponential random variable are given below

```
Theorem 6.1:
       ⊢ ∀ l x. (0 < l) ⇒ cdf (λs. exp_rv l s) x =
                if x ≤ 0 then 0 else (1 - exp (-l * x))

Theorem 6.2:
       ⊢ ∀ l x. (0 < l) ⇒ ({s | exp_rv r s ≤ x} ∈ 𝓔) ∧
                           ({s | exp_rv r s = x} ∈ 𝓔)
```

The above results allow us to formally reason about interesting probabilistic properties of continuous random variables within a higher-order-logic theorem prover. The measurability of the sets $\{s|\ F^{-1}(U\ s) \leq x\}$ and $\{s|\ F^{-1}(U\ s) = x\}$ can be used to prove that any set that involves a relational property with the random variable $F^{-1}(U\ s)$, e.g., $\{s \mid F^{-1}(U\ s)\ < x\}$ and $\{s \mid F^{-1}(U\ s) \geq\ x\}$, is measurable because of the closed under complements and countable unions property of $\mathcal{E}$. The CDF properties proved in Section 4 can then be used to determine probabilistic quantities associated with these sets [13].

The CDF and measurability properties of the rest of the continuous random variables given in Table 1 can also be proved in a similar way [13]. For illustration purposes the corresponding CDF theorems are given below

```
Theorem 6.3:
       ⊢ ∀ a b x. (a < b) ⇒ cdf (λs. uniform_rv a b s) x =
                if x ≤ a then 0 else (if x < b then (x-a)/(b-a) else 1)

Theorem 6.4:
       ⊢ ∀ x l. (0 < l) ⇒ cdf (λs. rayleigh_rv l s) x =
                if x ≤ 0 then 0 else (1 - exp(x^2)/(2l^2))

Theorem 6.5:
       ⊢ ∀ a x. (0 < a) ⇒ cdf (λs. triangular_rv a s) x =
                if (x ≤ 0) then 0 else
                (if (x < a) then (2/a (x - x^2/2a)) else 1)
```

7 Applications

A distinguishing characteristic of the proposed probabilistic analysis approach is the ability to perform precise quantitative analysis of probabilistic systems. In this section, we first illustrate this statement by considering a simple probabilistic analysis example. Then, we present some probabilistic systems which can be formally analyzed using the continuous random variables defined in Section 6.

Consider the problem of determining the probability of the event when there is no incoming request for 10 seconds in a Web server. Assume that the *interarrival* time of incoming requests is known from statistical analysis and is exponentially distributed with an average rate of requests $\lambda = 0.1$ jobs per second. We know from analytical analysis that this probability is precisely equal to $(\frac{1}{exp\ 1})$. This result can be verified in the HOL theorem prover by considering the probability of the event when the value of the Exponential random variable, with parameter 0.1 (i.e., $\lambda = 0.1$), lies in the interval $[10, \infty)$.

$$\vdash \mathbb{P}\ \{\texttt{s}\ |\ \texttt{10} < \texttt{exp\_rv 0.1 s}\} = \frac{1}{exp\ 1}$$

The first step in evaluating a probabilistic quantity is to prove that the event under consideration is measurable. The set in the above proof goal is measurable since it is the complement of a measurable set $\{s|exp\_rv\ 0.1\ s \leq 10\}$ (Theorem 6.2) and $\mathcal{E}$ is closed under complements and countable unions. The next step is to express the unknown probabilistic quantity in terms of the CDF of the given random variable. This can be done for the above proof goal by using the measurability property of the set under consideration and using the *complement law* of probability function, i.e., $(\mathbb{P}(\bar{S}) = 1 - \mathbb{P}(S)$.

$$\vdash \mathbb{P}\ \{\texttt{s}\ |\ \texttt{10} < \texttt{exp\_rv 0.1 s}\} = \texttt{1 - (cdf (}\lambda\texttt{s. exp\_rv 0.1 s) 10)}$$

The CDF of the Exponential random variable given in Theorem 6.1 can now be used to simplify the right-hand-side of the above equation to be equal to $(\frac{1}{exp\ 1})$. Thus, we were able to determine the unknown probability with 100% precision; a novelty which is not available in simulation based approaches.

The higher-order-logic theorem proving based probabilistic analysis can be applied to a variety of different domains, for instance, the sources of error in computer arithmetic operations are basically quantization operations and are modeled as uniformly distributed continuous random variables [24]. A number of successful attempts have been made to perform the statistical analysis of computer arithmetic analytically or by simulation (e.g., [15]). These kind of analysis form a very useful case study for our formalized continuous Uniform distribution as the formalization of both floating-point and fixed-point numbers already exist in HOL [1]. Similarly, the continuous probability distributions are extensively used for the analysis of probabilistic algorithms and network protocols [18]. Using our formalized models, these kind of analysis can be performed within the sound environment of the HOL theorem prover. The Exponential distribution in particular, due to its memoryless property and its relationship to the Poisson process [23], can be used to formalize the Birth-Death process which

is a Continuous-Time Markov Chain. The higher-order-logic formalization of the Birth-Death process may open the door for the formalized probabilistic analysis of a wide range of queuing systems, e.g., the CSMA/CD protocol [6], the IEEE 802.11 wireless LAN protocol [17], etc.

8 Related Work

Hurd's PhD thesis [14] can be regarded as one of the pioneering works in regards to formalizing probabilistic programs in a higher-order-logic theorem prover. An alternative method has been presented by Audebaud *et. al* [2]. 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. The main contribution of our paper is the extension of Hurd's framework to verify sampling algorithms for continuous probability distributions in HOL, a novelty that has not been available in any higher-order-logic theorem prover so far.

Another promising approach for conducting formal probabilistic analysis is to use probabilistic model checking, e.g., [3], [22]. Like traditional model checking, it involves the construction of a precise mathematical model of the probabilistic system which is then subjected to exhaustive analysis to verify if it satisfies a set of formal properties. This approach is capable of providing precise solutions in an automated way; however, it is limited to systems that can be expressed as a probabilistic finite state machine. It is because of this reason that probabilistic model checking techniques are not capable of providing precise reasoning about quantitative probabilistic properties related to continuous random variables. On the other hand, it has been shown in this paper that higher-order-logic theorem proving provides this capability. Another major limitation of probabilistic model checking is the state space explosion [4], which is not an issue with our approach.

A number of *probabilistic languages*, e.g., `Probabilistic cc` [9], λ_o [19] and `IBAL` [20], can be found in the open literature, which are capable of modeling continuous random variables. These probabilistic languages allow programmers to perform probabilistic computations at the level of probability distributions by treating probability distributions as primitive data types. It is interesting to note that the probabilistic language, λ_o, is based on sampling functions, i.e., a mapping from the unit interval [0,1] to a probability domain $\mathfrak{D}$ and thus shares the main ideas formalized in this paper. The main benefit of these probabilistic languages is their high expressiveness but they have their own limitations. For example, either they require a special treatment such as the lazy list evaluation strategy in `IBAL` and the limiting process in `Probabilistic cc` or they do not support precise reasoning as in the case of λ_o. The proposed theorem proving approach, on the other hand, is not only capable of formally expressing most continuous probability distributions but also to precisely reason about them.

9 Conclusions

In this paper, we have proposed to use higher-order-logic theorem proving for probabilistic analysis as a complementary approach to state-of-the-art simulation based techniques. Because of the formal nature of the models the analysis is free of approximation errors, which makes the proposed approach very useful for the performance and reliability optimization of safety critical and highly sensitive engineering and scientific applications.

We presented a methodology for the formalization of continuous probability distributions, which is a significant step towards the development of formal probabilistic analysis methods. Based on this methodology, we described the construction details of a framework for the formalization of all continuous probability distributions for which the inverse of the CDF can be expressed in a closed mathematical form. The major HOL definitions and theorems in this framework have been included in the current paper and more details can be found in [13]. We demonstrated the practical effectiveness of our framework by formalizing four continuous probability distributions; Uniform, Exponential, Rayleigh and Triangular. To the best of our knowledge, this is the first time that the formalization of these continuous random variables has been presented in a higher-order-logic theorem prover.

For our verification, we utilized the HOL theories of *Boolean Algebra*, *Sets*, *Natural Numbers*, *Real Numbers*, *Measure* and *Probability*. Our results can therefore be used as an evidence for the soundness of existing HOL libraries and the usefulness of theorem provers in proving pure mathematical concepts. The presented formalization can be utilized for the formalization of a number of other mathematical theories as well. For example, the CDF properties can be used along with the derivative function [10] to formalize the Probability Density Function, which is a very significant characteristic of continuous random variables and can be used to formalize the corresponding statistical quantities. Similarly, the formalization of the Standard Uniform random variable can also be transformed to formalize other continuous probability distributions, for which the inverse CDF is not available in a closed mathematical form. This can be done by exploring the formalization of other nonuniform random number generation techniques such as Box-Muller and acceptance/rejection [5]. Another interesting area that needs to be explored is the support of multiple independent continuous random variables.

References

1. Akbarpour, B., Tahar, S.: Formalization of Fixed-Point Arithmetic in HOL. Formal Methods in Systems Design 27(1-2), 173–200 (2005)
2. Audebaud, P., Paulin-Mohring, C.: Proofs of Randomized Algorithms in Coq. In: Uustalu, T. (ed.) MPC 2006. LNCS, vol. 4014, pp. 49–68. Springer, Heidelberg (2006)
3. 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)

4. Clarke, E.M, Grumberg, O., Peled, D.A: Model Checking. MIT Press, Cambridge (2000)
5. Devroye, L.: Non-Uniform Random Variate Generation. Springer, Heidelberg (1986)
6. Gonsalves, T.A, Tobagi, F.A: On the Performance Effects of Station Locations and Access Protocol Parameters in Ethernet Networks. IEEE Trans. on Communications 36(4), 441–449 (1988)
7. Gordon, M.J.C: Mechanizing Programming Logics in Higher-0rder Logic. In: Current Trends in Hardware Verification and Automated Theorem Proving, pp. 387–439. Springer, Heidelberg (1989)
8. Gordon, M.J.C, Melham, T.F: Introduction to HOL: A Theorem Proving Environment for Higher-Order Logic. Cambridge University Press, Cambridge (1993)
9. Gupta, V.T, Jagadeesan, R., Panangaden, P.: Stochastic Processes as Concurrent Constraint Programs. In: Principles of Programming Languages, pp. 189–202. ACM Press, New York (1999)
10. Harrison, J.: Theorem Proving with the Real Numbers. Springer, Heidelberg (1998)
11. Hasan, O., Tahar, S.: Formalization of the Standard Uniform Random Variable. Theoretical Computer Science (to appear)
12. 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)
13. Hasan, O., Tahar, S.: Formalization of Continuous Probability Distributions. Technical Report, Concordia University, Montreal, Canada (February 2007) http://hvg.ece.concordia.ca/Publications/TECH_REP/FCPD_TR07
14. Hurd, J.: Formal Verification of Probabilistic Algorithms. PhD Thesis, University of Cambridge, Cambridge, UK (2002)
15. Kaneko, T., Liu, B.: On Local Roundoff Errors in Floating-Point Arithmetic. ACM 20(3), 391–398 (1973)
16. Khazanie, R.: Basic Probability Theory and Applications. Goodyear (1976)
17. Köpsel, A., Ebert, J., Wolisz, A.: A Performance Comparison of Point and Distributed Coordination Function of an IEEE 802.11 WLAN in the Presence of Real-Time Requirements. In: Proceedings of Seventh International Workshop on Mobile Multimedia Communications, Tokyo, Japan (2000)
18. Mitzenmacher, M., Upfal, E.: Probability and Computing. Cambridge University Press, Cambridge (2005)
19. Park, S., Pfenning, F., Thrun, S.: A Probabilistic Language based upon Sampling Functions. In: Principles of Programming Languages, pp. 171–182. ACM Press, New York (2005)
20. Pfeffer, A.: IBAL: A Probabilistic Rational Programming Language. In: International Joint Conferences on Artificial Intelligence, pp. 733–740. Morgan Kaufmann Publishers, Washington (2001)
21. Ross, S.M: Simulation. Academic Press, San Diego (2002)
22. Rutten, J., Kwaiatkowska, M., Normal, G., Parker, D.: Mathematical Techniques for Analyzing Concurrent and Probabilisitc Systems. CRM Monograph Series. American Mathematical Society, vol. 23 (2004)
23. Tridevi, K.S: Probability and Statistics with Reliability, Queuing and Computer Science Applications. Wiley, Chichester (2002)
24. Widrow, B.: Statistical Analysis of Amplitude-quantized Sampled Data Systems. AIEE Trans. (Applications and Industry) 81, 555–568 (1961)