Verification of Probabilistic Properties in HOL Using the Cumulative Distribution Function

Osman Hasan and Sofiène Tahar

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

Abstract. Traditionally, computer simulation techniques are used to perform probabilistic analysis. However, they provide inaccurate results and cannot handle large-scale problems due to their enormous CPU time requirements. To overcome these limitations, we propose to complement simulation based tools with higher-order-logic theorem proving so that an integrated approach can provide exact results for the critical sections of the analysis in the most efficient manner. In this paper, we illustrate the practical effectiveness of our idea by verifying numerous probabilistic properties associated with random variables in the HOL theorem prover. Our verification approach revolves around the fact that any probabilistic property associated with a random variable can be verified using the classical *Cumulative Distribution Function* (CDF) properties, if the CDF relation of that random variable is known. For illustration purposes, we also present the verification of a couple of probabilistic properties, which cannot be evaluated precisely by the existing simulation techniques, associated with the Continuous Uniform random variable in HOL.

Keywords: Interactive Theorem Proving, Higher-Order-Logic, Probabilistic Systems, Cumulative Distribution Function, HOL.

1 Introduction

Probabilistic analysis has become a tool of fundamental importance to virtually all engineers and scientists as they often have to deal with systems that exhibit significant random or unpredictable elements. The main idea behind probabilistic analysis is to model these uncertainties by random variables and then judge the performance and reliability issues based on the corresponding probabilistic properties.

Random variables are basically functions that map random events to numbers. Every random variable gives rise to a probability distribution, which contains most of the important information about this random variable. The probability distribution of a random variable can be uniquely described by its *Cumulative Distribution Function* (CDF), which is sometimes also referred to as the probability distribution function. The CDF of a random variable R, $F_R(x)$, represents

J. Davies and J. Gibbons (Eds.): IFM 2007, LNCS 4591, pp. 333–352, 2007.

the probability that the random variable R takes on a value that is less than or equal to a real number x

$$F_R(x) = Pr(R \leq x) \tag{1}$$

where Pr denotes the probability. The CDF of a random variable contains complete information about the probability model of the random variable and one of its major significance is that it can be used to characterize both discrete and continuous random variables. A distribution is called discrete if its CDF consists of a sequence of finite or countably infinite jumps, which means that it belongs to a random variable that can only attain values from a certain finite or countably infinite set. Discrete random variables can also be characterized by their *probability mass function* (PMF), which represents the probability that the given random variable R is exactly equal to some value x, i.e., $Pr(R = x)$. A distribution is called continuous if its CDF is continuous, which means that it belongs to a random variable that ranges over a continuous set of numbers that contains all real numbers between two limits. A Continuous random variable can also be characterized by its *probability density function* (PDF), which represents the slope of its CDF, i.e., $\frac{dF_R(x)}{dx}$. Besides characterizing both discrete and continuous random variables, the CDF also allows us to determine the probability that a random variable falls in any arbitrary interval of the real line. Because of these reasons, the CDF is regarded as one of the most useful characteristic of random variables in the field of probabilistic analysis where the main goal is to determine the probabilities for various events.

Today, simulation is the most commonly used computer based probabilistic analysis technique. Most simulation softwares provide a programming environment for defining functions that approximate random variables for probability distributions. The random elements in a given system are modeled by these functions and the system is analyzed using computer simulation techniques, such as the Monte Carlo Method [17], where the main idea is to approximately answer a query on a probability distribution by analyzing a large number of samples. The inaccuracy of the probabilistic analysis results offered by simulation based techniques poses a serious problem in highly sensitive and safety critical applications, such as space travel, medicine and military, where a mismatch between the predicted and the actual system performance may result in either inefficient usage of the available resources or paying higher costs to meet some performance or reliability criteria unnecessarily. Besides the inaccuracy of the results, another major limitation of simulation based probabilistic analysis is the enormous amount of CPU time requirement for attaining meaningful estimates. This approach generally requires hundreds of thousands of simulations to calculate the probabilistic quantities and becomes impractical when each simulation step involves extensive computations.

In order to overcome the limitations of the simulation based approaches, we propose to use higher-order logic interactive theorem proving [9] for probabilistic analysis. Higher-order logic is a system of deduction with a precise semantics and can be used for the development of almost all classical mathematics theories. Interactive theorem proving is the field of computer science and mathematical logic concerned with computer based formal proof tools that require some sort of human assistance. We believe that probabilistic analysis can be performed by specifying the behavior of systems which exhibit randomness in higher-order logic and formally proving the intended probabilistic properties within the environment of an interactive theorem prover. Due to the inherent soundness of this approach, the probabilistic analysis carried out in this way will be capable of providing exact answers. It is important to note here that higher-order-logic theorem proving cannot be regarded as the golden solution in performing probabilistic analysis because of its own limitations. Even though theorem provers have been successfully used for a variety of tasks, including some that have eluded human mathematicians for a long time, but these successes are sporadic, and work on hard problems usually requires a proficient user and a lot of formalization. On the other hand, simulation based techniques are at least capable of offering approximate solutions to these problems. Therefore, we consider simulation and higher-order-logic theorem proving as complementary techniques, i.e., the methods have to play together for a successful probabilistic analysis framework. For example, the proposed theorem proving based approach can be used for the safety critical parts of the design which can be expressed in closed mathematical forms and simulation based approaches can handle the rest.

The foremost conditions for conducting probabilistic analysis within the environment of a higher-order-logic theorem prover are (1) the higher-order-logic formalization of random variables; and (2) to be able to formally verify the probabilistic properties of these random variables within the theorem prover. This paper is mainly targeted towards the second condition above, though the formalization of random variables is also discussed briefly. Our approach for the verification of probabilistic properties, illustrated in Figure 1, is primarily based on the fact that if a random variable is formally specified and its CDF relation is formally verified in a higher-order-logic theorem prover then the classical CDF properties [16] can be used to prove any of its probabilistic properties. The paper presents the verification of these classical CDF properties and the formal proofs for the facts that any probabilistic property for a given random variable, including the PMF and the PDF, can be expressed in terms of its CDF.

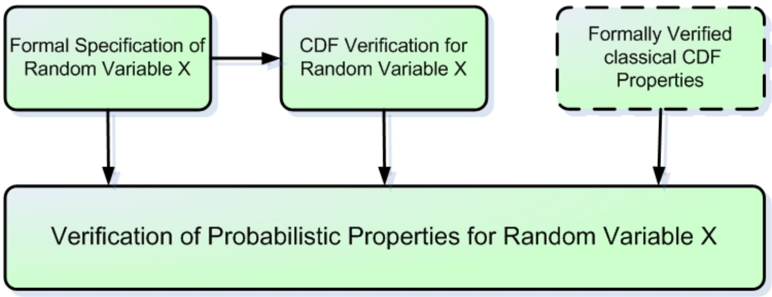
Formal Specification of Random Variable X
CDF Verification for Random Variable X
Formally Verified classical CDF Properties
Verification of Probabilistic Properties for Random Variable X

Fig. 1. Framework for Verifying Probabilistic Properties

We have selected the HOL theorem prover [10] for the current formalization mainly in order to build upon the existing mathematical theories of *Measure* and *Probability.* Hurd [14] developed these theories and also presented a framework for the formalization of probabilistic algorithms in his PhD thesis. Random variables are basically probabilistic algorithms and Hurd's thesis also contains the formalization of some discrete random variable which are verified by proving their corresponding PMF relations in the HOL theorem prover.

The rest of the paper is organized as follows. In Section 2, we present a brief introduction to the HOL theorem prover and an overview of Hurd's methodology for the formalization of probabilistic algorithms in HOL . Then in Section 3, we show how Hurd's formalization framework can be extended to formalize continuous random variables as well by defining the Standard Uniform random variable and proving its CDF relation in the HOL theorem prover. The benefit of the formal definition of the Standard Uniform random variable is that it can be used along with nonuniform random number generation techniques [7] to formalize other continuous random variables in HOL. In Section 4, we formally specify the CDF by a real valued higher-order-logic function and provide the formal verification of its classical properties within the HOL theorem prover. Section 5 illustrates the usefulness of the formally verified CDF properties in constructing a higher-order-logic theorem prover based probabilistic analysis framework. In this section, we have included the HOL proofs for the facts that the CDF relation of a random variable can be used along with the formally verified CDF properties to determine any of its associated probabilistic quantities. Then in Section 6, we outline the process of verifying a couple of probabilistic properties associated with the Continuous Uniform random variable within the HOL theorem prover to illustrate the practical effectiveness of the proposed approach. A review of the related work in the literature is given in Section 7 and we finally conclude the paper in Section 8.

2 Preliminaries

In this section, we provide an overview of the HOL theorem prover and Hurd's methodology [14] for the formalization of probabilistic algorithms in HOL. The intent is to provide a brief introduction to these topics along with some notation that is going to be used in the next sections.

2.1 HOL Theorem Prover

The HOL theorem prover, developed at the University of Cambridge, UK, is an interactive theorem prover which is capable of conducting proofs in higher-order logic. It utilizes the simple type theory of Church [5] along with Hindley-Milner polymorphism [22] to implement higher-order logic. HOL has been successfully used as a verification framework for both software and hardware 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 it with the necessary tactics to prove goals while some of the proof steps are solved automatically by the automatic proof procedures.

In order to ensure secure theorem proving, the logic in the HOL system is represented in the strongly-typed functional programming language ML [24]. 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 these basic axioms and primitive inference rules or any other pre-existing theorems/inference rules.

We selected the HOL theorem prover for the proposed formalization mainly because of its inherent soundness and ability to handle higher-order logic and in order to benefit from the built-in mathematical theories for measure and probability.

2.2 Verifying Probabilistic Algorithms in HOL

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

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

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 function *bit* accepts the infinite Boolean sequence and returns a random number, which is either 0 or 1 together with a sequence of unused Boolean sequence, which in this case is the tail of the sequence. The above methodology can be used to model most probabilistic algorithms. All probabilistic algorithms that compute a finite number of values equal to 2^n, each having a probability of the form $\frac{m}{2^n}$: where m represents the HOL data type *nat* and is always less than 2^n, can be modeled, using Hurd's framework, by well-founded recursive functions. The probabilistic algorithms that do not satisfy the above conditions but are sure to terminate can be modeled by the *probabilistic while loop* proposed in [14].

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 the 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 derive the basic laws of probability in the HOL prover, e,g., the additive law, which represents the probability of two disjoint events as the sum of their probabilities:

```
⊢ ∀ A B. A ∈ ℰ ∧ B ∈ ℰ ∧ A ∩ B = ∅ ⇒
                              ℙ(A ∪ B) = ℙ(A) + ℙ(B)
```

The formalized $\mathbb{P}$ and $\mathcal{E}$ can also be used to prove probabilistic properties for probabilistic programs such as

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

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 of a function is an important concept in probability theory and also a useful practical tool for proving that sets are measurable [3]. 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 four discrete random variables and proved their correctness by proving the corresponding PMF relations [14]. Because of their discrete nature, all these random variables either compute a finite number of values or are sure to terminate. 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, continuous random variables always compute an infinite number of values and therefore would require all the random bits in the infinite Boolean sequence if they are to be represented using Hurd's methodology. The corresponding deterministic functions cannot be expressed by either recursive functions or the probabilistic while loop and it is mainly for this reason that the specification of continuous random variables needs to be handled differently than their discrete counterparts.

3 Formalization of the Standard Uniform Distribution

In this section, we present the formalization of the Standard Uniform distribution in the HOL theorem prover. The Standard Uniform random variable is a continuous random variable and can be characterized by the CDF as follows:

$$Pr(X \leq x) = \begin{cases} 0 \text{ if } x < 0; \\ x \text{ if } 0 \leq x < 1; \\ 1 \text{ if } 1 \leq x. \end{cases} \tag{2}$$

One of the significant aspects of formalizing the Standard Uniform random variable is that it can be utilized along with the nonuniform random number generation techniques [7] to model other continuous random variables in the HOL theorem prover as well. Therefore, it opens the doors of formally verifying the probabilistic properties of systems that exhibit randomness of continuous nature.

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 [13]

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

where, X_k denotes the outcome of the k^{th} random bit; $true$ or $false$ represented as 1 or 0 respectively. The mathematical relation of Equation (3) 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 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 0 = unit 0) ∧
    ∀ n. (std_unif_disc (n+1) =
        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 (3) in the HOL theorem prover. Now, the formalization of the mathematical concept of limit of a real sequence in HOL [12] can be used to formally specify the Standard Uniform random variable of Equation (3) as follows

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

where, *lim M* represents the HOL formalization of the limit of a real sequence [12], such that *lim M* is the limit value of the real sequence M (i.e., $\lim_{n \to \infty} M(n) =$ *lim M*).

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 (2). The first step in this verification is to prove the measurability of the set under consideration, i.e., to prove that the set $\{s \mid std\_unif\_cont\ s \leq x\}$ is in $\mathcal{E}$. Since, the function *std_unif_disc* accesses the infinite boolean sequence using only the `unit`, `bind` and `sdest` primitives, Hurd's formalization framework can be used to prove

```
Lemma 3.1:
⊢ ∀ x,n. {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 the sets in the sequence $(\lambda n.\{s \mid FST(std\_unif\_disc\ n\ s) \leq x\})$ are measurable according to Lemma 3.1.

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

Theorem 3.1 can now be used along with the real number theories [12] to verify the CDF of the probabilistic function *std_unif_cont* in the HOL theorem prover and the verification details can be found in [13].

```
Theorem 3.2:
⊢ ∀ x. 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 formal specification of the CDF and the verification of CDF properties in the HOL theorem prover. The CDF and its properties have been an integral part of the classical probability theory since its early development in the 1930s. The properties are mentioned in most of the probability theory texts, e.g, [16] and have been used successfully in performing analytical analysis of random systems using paper-pencil proofs. Our main contribution is the formalization of these properties in a mechanical theorem prover. The proof process was long and tedious requiring a deep understanding and proficiency in both the mathematical backgrounds (Boolean Algebra, Set Theory, Real Theory, Measure Theory and Probability Theory) as well as the HOL theorem prover. The motivation, on the other hand, is that these formalized properties can now be utilized to obtain a complete, rigorous and communicable description of random components in a system. Also, the formalization allows us to perform machinized proofs regarding probabilistic properties within the framework of a sound theorem-prover environment.

4.1 Formal Specification of CDF

It follows from Equation (1) 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. Hurd's formalization of the probability function $\mathbb{P}$, which maps sets of infinite Boolean sequences to real numbers between 0 and 1, can be used to formally specify the CDF as follows:

```
Definition 4.1:
⊢ ∀ R x. cdf R x = ℙ {s | R s ≤ x}
```

where, R represents the random variable that accepts an infinite Boolean sequence and returns a real number and the set $\{s \mid R\ s \leq x\}$ is the set of all infinite Boolean sequences, s, that satisfy the condition $(R\ s \leq x)$.

4.2 Formal Verification of CDF Properties

Using the formal specification of the CDF, we are able to verify the classical CDF properties [16] within the HOL theorem prover. The formal proofs for these properties not only ensure the correctness of our CDF specification but also play a vital role in proving various probabilistic properties associated with random variables as shown in Figure 1. All properties in the following sections are verified 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.

CDF Bounds. *For any real number x,*

$$0 \leq F_R(x) \leq 1 \tag{4}$$

This property states that if we plot the CDF against its real argument x, then the graph of the CDF, F_R, is between the two horizontal lines $y = 0$ and $y = 1$. In other words, the lines $y = 0$ and $y = 1$ are the bounds for the CDF F_R.

The above characteristic can be verified in HOL using the fact that the CDF is basically a probabilistic quantity along with the basic probability law, verified in [14], that states that the probability of an event is always less than 1 and greater than 0 ($\forall S.\ S \in \mathcal{E} \Rightarrow 0 \leq \mathbb{P}(S) \leq 1$).

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

CDF is Monotonically Increasing. *For any two real numbers a and b,*

$$if\ a < b,\ then\ F_R(a) \leq F_R(b) \tag{5}$$

In mathematics, functions between ordered sets are monotonic if they preserve the given order. Monotonicity is an inherent characteristic of CDFs and the CDF value for a real argument a can never exceed the CDF value of a real argument b if a is less than b.

Using the set theory in HOL, it can be proved that for any two real numbers a and b, if $a < b$ then the set of infinite Boolean sequences $\{s \mid R\ s \leq a\}$ is a subset of the set $\{s \mid R\ s \leq b\}$. Then, using the monotone law of the probability function ($\forall S\ T.\ S \in \mathcal{E}\ \wedge\ T \in \mathcal{E}\ \wedge\ S \subseteq T \Rightarrow (\mathbb{P}(S) \leq \mathbb{P}(T))$, verified in [14], we proved the monotonically increasing property of the CDF in HOL.

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

Interval Probability. *For any two real numbers a and b,*

$$if\ a < b,\ then\ Pr(a < R \leq b) = F_R(b) - F_R(a) \tag{6}$$

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.

Using the set theory in HOL, it can be proved that for any two real numbers a and b, if $a < b$ then the set of infinite Boolean sequences $\{s \mid R\ s \leq b\}$ is equal to the union of the sets $\{s \mid R\ s \leq a\}$ and $\{s \mid (a < R\ s) \wedge (R\ s \leq b)\}$. Now, the above CDF property can be proved in HOL using the additive law of the probability function ($\forall S\ T.\ S \in \mathcal{E}\ \wedge\ T \in \mathcal{E}\ \wedge\ S \cap T = \emptyset\ \Rightarrow (\mathbb{P}(S \cup T) = \mathbb{P}(S) + \mathbb{P}(T))$, verified in [14], along with the closed under complements and countable unions property of $\mathcal{E}$.

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

CDF at Negative Infinity

$$\lim_{x \to -\infty} F_R(x) = 0;\ \textit{that is},\ F_R(-\infty) = 0 \tag{7}$$

This property states that the value of the CDF tends to 0 as its real argument approaches negative infinity or in other words the graph of CDF must eventually approach the line $y = 0$ at the left end of the real axis.

We used the formalization of limit of a real sequence [12] along with the formalization of the mathematical measure theory [14] in HOL to prove this property. The first step is to prove a relationship between the limit value of the probability of a monotonically decreasing sequence of events A_n (i.e, $A_{n+1} \subseteq A_n$ for every n) and the probability of the countable intersection of all events that can be represented as A_n.

$$\forall A_n.\ \lim_{n \to \infty} Pr(A_n) = Pr(\bigcap_n A_n) \tag{8}$$

This relationship, sometimes called the *Continuity Property of Probabilities*, can be used to prove the above CDF property by instantiating it with a decreasing sequence of events represented in Lambda calculus as $(\lambda n.\{s \mid R\ s \leq -(\&n)\})$; where n has the HOL data type *nat*: $\{0, 1, 2, \ldots\}$ and "&" converts it to its corresponding real number. The left hand side of Equation 8, with this sequence, represents the CDF for the random variable R when its real argument approaches negative infinity and thus is equal to the left hand side of our proof goal in Equation 7. Using the monotonically decreasing nature of the events in the sequence $(\lambda n.\{s \mid R\ s \leq\ -(\&n)\})$, the right hand side of Equation 8, with this sequence, can be proved to be equal to the probability of an empty set. The CDF at negative infinity property can now be proved using the basic probability law $(\mathbb{P}(\{\}) = 0)$, verified in [14], which states that the probability of an empty set is 0.

```
Theorem 4.4:
⊢ ∀ R. lim (λ n. CDF R (-&n)) = 0
```

where, *lim* is the HOL function for the limit of a real sequence [12].

CDF at Positive Infinity

$$\lim_{x \to \infty} F_R(x) = 1;\ \textit{that is},\ F_R(\infty) = 1 \tag{9}$$

This property, quite similar to the last one, states that the value of the CDF tends to 1 as its real argument approaches positive infinity or in other words the graph of CDF must eventually approach the line $y = 1$ at the right end of the real axis.

The HOL proof steps for this property are also quite similar to the last one and this time we use the Continuity Property of Probabilities which specifies the relationship between the limit value of the probability of a monotonically increasing sequence of events A_n (i.e, $A_n \subseteq A_{n+1}$ for every n) and the probability of the countable union of all events that can be represented as A_n.

$$\forall A_n. \lim_{n \to \infty} Pr(A_n) = Pr(\bigcup_n A_n) \tag{10}$$

In this case, we instantiate Equation 10 with an increasing sequence of events represented in Lambda calculus as $(\lambda n.\{s \mid R\ s \leq\ (\&n)\})$. The countable union of all events in this sequence is the universal set. The CDF at positive infinity property can now be proved in the HOL theorem prover using the basic probability law ($\mathbb{P}$(UNIV) = 1), verified in [14], which states that the probability of the universal set is 1.

```
Theorem 4.5:
⊢ ∀ R. lim (λ n. CDF R (&n)) = 1
```

CDF is Continuous from the Right. *For every real number a,*

$$\lim_{x \to a^+} F_R(x) = F_R(a) \tag{11}$$

where $\lim_{x \to a^+} F_R(x)$ 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.

In order to prove this property in HOL, we used a decreasing sequence of events represented in Lambda calculus as $(\lambda n.\{s \mid R\ s \leq\ a + \frac{1}{\&(n+1)}\})$. This sequence of events has been selected in such a way that if the Continuity Property of Probabilities, given in Equation 8, is instantiated with this sequence then its left hand side represents the CDF for a random variable, R, when its real argument approaches a through values greater than a. Therefore, with this sequence, the left hand side of the Continuity Property of Probabilities is equal to the left hand side of our proof goal in Equation 11. Using the monotonically decreasing nature of the events in the sequence $(\lambda n.\{s \mid R\ s \leq\ a + \frac{1}{\&(n+1)}\})$, it can also be proved that the countable intersection of all events in this sequence is the set $\{s \mid R\ s \leq\ a\}$. The CDF can now be proved to be continuous from the right as the right hand side of the Continuity Property given in Equation 8, with the sequence $(\lambda n.\{s \mid R\ s \leq\ a + \frac{1}{\&(n+1)}\})$, represents the CDF of random variable at real argument a.

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

CDF Limit from the Left. *For every real number a,*

$$\lim_{x \to a^-} F_R(x) = Pr(R < a) \tag{12}$$

where $\lim_{x \to a^-} F_R(x)$ is defined as the limit of $F_R(x)$ as x tends to a through values less than a.

This property is quite similar to the previous one and can be proved by instantiating the Continuity Property of Probabilities, given in Equation 10, with an increasing sequence of events represented in Lambda calculus as $(\lambda n.\{s \mid R\ s \leq a - \frac{1}{\&(n+1)}\})$. The left hand side of Equation 10, with this sequence, represents the CDF for the random variable R when its real argument approaches a through values less than a and is thus equal to the left hand side of our proof goal in Equation 12. Using the monotonically increasing nature of the events in the sequence $(\lambda n.\{s \mid R\ s \leq\ a - \frac{1}{\&(n+1)}\})$, it can be proved that the countable union of all the events in this sequence is the set $\{s \mid R\ s <\ a\}$ which led us to prove the theorem stating the CDF limit from the left.

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

5 CDF Properties and Probabilistic Analysis

As mentioned in Section 1, probabilistic analysis is basically the process of evaluating performance and/or reliability of a given system by representing its uncertain elements in terms of random variables and characterizing the results in terms of the corresponding probabilistic quantities. We have already seen in Sections 2 and 3 of this paper that both discrete and continuous random variables can be formalized in the HOL theorem prover. In this section, we illustrate the usefulness of the formally verified CDF properties in relevance to evaluating probabilistic quantities while performing probabilistic analysis within the HOL theorem prover.

5.1 Determining Interval Probabilities

The CDF of a random variable, R, along with the CDF properties verified in Section 4 can be used to determine the probability that R will lie in any specified interval of the real line. In this section, we verify this statement in the HOL theorem prover by dividing the real line in three disjoint intervals; $(-\infty, a]$, $(a, b]$ and (b, ∞), and determining the probabilities that a random variable lies in these intervals in terms of its CDF.

Determining the probability for the first interval is quite straightforward since the CDF for a random variable, R, with a real argument, a, can be used directly to find the probability that a random variable, R, will lie in the interval $(-\infty, a]$. Whereas, the probability that a random variable, R, will lie in the interval $(a, b]$ can be determined by its CDF values for the real arguments a and b as has been proved in Theorem 4.3. For the third interval, we first use the set theory in HOL to prove that for any real value b, the set of infinite Boolean sequences $\{s \mid b < R\ s\}$ is the complement of the set $\{s \mid R\ s \leq b\}$. The probability that a random variable, R, lies in the interval (b, ∞) can now be represented in terms of its CDF by using the complement law of the probability function $(\forall S.\ S \in \mathcal{E}\ \Rightarrow \mathbb{P}(S) = 1 - \mathbb{P}(\bar{S}))$, verified in [14], under the assumption that the set $\{s|Rs \leq a\}$ is measurable.

```
Theorem 5.1:
⊢ ∀ R b. ℙ {s | b < R s} = 1 - (CDF R b)
```

5.2 Representing PMF in Terms of the CDF

The PMF can be expressed in terms of the CDF of the corresponding random variable by using the fact that for any real value a the set of infinite Boolean sequences $\{s \mid R\ s \leq a\}$ is equal to the union of the sets $\{s \mid R\ s < a\}$ and $\{s \mid R\,s = a\}$. Now, using Theorems 4.6 and 4.7, the additive law of the probability function $\mathbb{P}$ and the closed under complements and countable unions property of $\mathcal{E}$, the desired relationship can be proved under the assumption that the sets $\{s|Rs = a\}$ and $\{s|Rs \leq a\}$ are measurable.

Theorem 5.2:
$\vdash \forall$ R a. $\mathbb{P}$ {s | R s = a} = lim (λ n. CDF R (a + $\frac{1}{\&(n+1)}$)) -
lim (λ n. CDF R (a - $\frac{1}{\&(n+1)}$))

A unique characteristic for all continuous random variables is that their PMF is equal to 0. Theorem 5.2 along with the formalization of continuous functions allowed us prove this property in the HOL theorem prover.

```
Theorem 5.3:
⊢ ∀ R a. (∀x. (λx. CDF R x) contl x) ⇒ ℙ {s | R s = a} = 0)
```

where, ($\forall$ $x.f$ *contl* x) represents the HOL function definition for a continuous function [12] such that the function f is continuous for all x.

5.3 Representing PDF in Terms of the CDF

The PDF, which is the slope of the CDF, represents the probability distribution of a continuous random variable in terms of integrals. It can be expressed in the HOL theorem prover by using the formal definition of the CDF and the formalization of the mathematical concept of a derivative [12].

```
Definition 5.1:
⊢ ∀ R x. pdf R x = @l. ((λx. CDF R x) diffl l) x
```

where *(f diffl l)* x represents the HOL formalization of the derivative [12], such that l is the derivative of the function f with respect to the variable x, and @x.t represents the Hilbert choice operator in HOL ($\varepsilon x.t$ term), that represents the value of x such that t is true.

Using the above definition of the PDF, we were able to prove the following classical properties of the PDF [16] in the HOL theorem prover 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.

PDF Lower Bound. *For any real number x,*

$$0 \leq f_R(x) \tag{13}$$

This property states that if we plot the PDF against its real argument x, then the graph of the PDF, f_R, will never go below the horizontal line $y = 0$. In other words, the line $y = 0$ is the lower bound for the PDF f_R.

We utilized the monotonically increasing property of the CDF proved in Theorem 4.2 along with the nonnegative characteristic of the derivative of nondecreasing functions to prove this property in the HOL theorem prover.

```
Theorem 5.4:
⊢ ∀ R x. (∀x.(λx. CDF R x) differentiable x) ⇒ (0 ≤ pdf R x)
```

where, the condition (*f differentiable x*) ensures in HOL that a derivative exists for the function f for the variable x.

Interval Probability in Terms of PDF. *For any two real numbers a and b,*

$$if\ a < b,\ then\ Pr(a < R \leq b) = \int_a^b f_R(x)dx \tag{14}$$

We used the HOL formalization of the gauge integral [21], which has all the attractive convergence properties of the Lesbesgue integral, along with the interval property of the CDF, verified in Theorem 4.3, to prove the above property in the HOL theorem prover.

```
Theorem 5.5:
⊢ ∀ R x.(∀x. (λx. CDF R x) differentiable x) ⇒
    (Dint (a,b) (λx. pdf R x)
                (ℙ {s | (a ≤ R s) ∧ (R s ≤ b)}))
```

where *Dint(a,b) f k* represents the HOL formalization of the gauge integral [12] such that the definite integral of the function f over the interval [a,b] is k.

6 Illustrative Example

In this section, we illustrate the practical effectiveness of our approach by presenting a simplified probabilistic analysis example of roundoff error in a digital processor within the HOL theorem prover.

Assume that the roundoff error for a particular digital processor is uniformly distributed over the interval [-5x10^{-12}, 5x10^{-12}]. An engineering team is interested in verifying that the probability of the event when the roundoff error in this digital processor is greater than 2x10^{-12} is less than 0.33 and the probability that the final result fluctuates by $\pm$1x10^{-12} with respect to the actual value is precisely equal to 0.2. We now verify these properties in HOL by following the steps mentioned in Figure 1.

6.1 Formal Specification of the Continuous Uniform Distribution

The first step, in the higher-order-logic theorem proving based formal verification of probabilistic properties, is the formalization of the random variable that is required in the probabilistic analysis under consideration. The example under consideration calls for the Continuous Uniform random variable, which can be characterized by the CDF as follows

$$\mathbb{P}(X \leq x) = \begin{cases} 0 & \text{if } x \leq a; \\ \frac{x-a}{b-a} & \text{if } a < x \leq b; \\ 1 & \text{if } b < x. \end{cases} \tag{15}$$

The Continuous Uniform random variable can be formally expressed in terms of the formalized Standard Uniform random variable of Section 3 using the Inverse Transform Method (ITM) [7]. The ITM is a commonly used nonuniform random number generation technique for generating continuous random variants for probability distributions for which the inverse of the CDF can be expressed in a closed mathematical form.

```
Definition 6.1:
⊢ ∀ a b s. uniform_cont a b s = (b - a) * std_unif_cont s) + a
```

The function *uniform_cont*, which formally represents the Continuous Uniform random variable, accepts two real valued parameters *a*, and *b* and the infinite Boolean sequence *s* and returns a real number in the interval [a,b].

6.2 CDF Verification of the Continuous Uniform Random Variable

The second step in our approach for the verification of probabilistic properties associated with a random variable is the verification of its CDF relationship, as shown in Figure 1. This can be done by proving the CDF of the function *uniform_cont* to be equal to the theoretical value of the CDF of the Continuous Uniform random variable given in Equation 15.

The definition of the function *uniform_cont* and elementary real arithmetic operations may be used to transform the set $\{s|uniform\_cont\ a\ b\ s \ \leq \ x\}$ in such a way that $(std\_unif\_cont\ s)$ is the only term that remains on the left hand side of the inequality, i.e., $(\mathbb{P}\{s|std\_unif\_cont\ s \leq \frac{x-a}{b-a}\})$. Now, the CDF property for the function *std_unif_cont*, proved in Theorem 3.2, along with some simple arithmetic reasoning can be used to prove the desired CDF relationship.

```
Theorem 6.1:
⊢ ∀ a b x. (a < b) ⇒ ℙ{s | uniform_cont a b s ≤ x} =
        if (x ≤ a) then 0 else (if (x ≤ b) then (x-a)/(b-a) else 1)
```

Similarly, the measurability property proved in Theorem 3.1 can be used to prove the measurability property for the set that corresponds to the CDF of the probabilistic function *uniform_cont* in the HOL theorem prover.

```
Theorem 6.2:
⊢ ∀ a b x. (a < b) ⇒ measurable {s | uniform_cont a b s ≤ x}
```

6.3 Verification of Probabilistic Properties

After the completion of the above steps, we are now in the position of formally verifying the given probabilistic properties by modeling the roundoff error as a Continuous Uniform random variable in the interval [-5x10^{-12}, 5x10^{-12}].

We proceed to verify the first probabilistic property, which checks if the probability of the event when the roundoff error in this digital processor is greater than 2x10^{-12} is less than 0.33, by instantiating Theorem 5.1 with the random variable $(\lambda s.\ uniform\_cont\ -5x10^{-12}\ 5x10^{-12}\ s)$ and the real value 2x10^{-12}. Now the property can be verified by simplifying the result using the formal definition of the CDF, given in Definition 4.1, Theorems 6.1 and 6.2 and the real number theories in HOL [12].

Theorem 6.3:
$\vdash \mathbb{P}\ \{\texttt{s} \mid 2\texttt{x}10^{-12} < \texttt{uniform\_cont -5x}10^{-12}\ \texttt{5x}10^{-12}\ \texttt{s}\ \} < 0.33$

Similarly the second property, which checks if the probability of the final result fluctuating by ±1x10^{-12} with respect to the actual value is precisely equal to 0.2, can be verified by checking if the probability of the Continuous Uniform random variable, defined in the interval [-5x10^{-12}, 5x10^{-12}], falling in the interval [-1x10^{-12}, 1x10^{-12}] is equal to 0.2. This can be done by using the definition of CDF, Theorems 6.1 and 6.2 and instantiating the CDF property verified in Theorem 4.3 by the real values -1x10^{-12}, 1x10^{-12} for variables a, b and the random variable $(\lambda s.\ uniform\_cont\ -5x10^{-12}\ 5x10^{-12}\ s)$ for variable R.

Theorem 6.4:
$\vdash \mathbb{P}\ \{\texttt{s} \mid (\texttt{-1x}10^{-12} < \texttt{uniform\_cont -5x}10^{-12}\ \texttt{5x}10^{-12}\ \texttt{s}) \wedge$
$\quad (\texttt{uniform\_cont -5x}10^{-12}\ \texttt{5x}10^{-12}\ \texttt{s} \leq \texttt{1x}10^{-12})\} = 0.2$

The above example illustrates the fact that the interactive theorem proving based approach is capable to verify probabilistic quantities, which can be expressed in a closed mathematical form, with 100% precision; a novelty which is not available in the simulation based techniques. Thus, by integrating the higher-order-logic theorem proving capability to the simulation based tools, the level of the overall accuracy of the results can be raised. This added benefit comes at the cost of a significant amount of time and effort spent, while formalizing the system behavior, by the user.

7 Related Work

Due to the vast application domain of probability, many researchers around the world are trying to improve the quality of computer based probabilistic analysis. The ultimate goal is to come up with tools that are capable of providing accurate analysis, can handle large-scale problems and are easy to use. In this section, we provide a brief account of the state-of-the-art and discuss some related work in the field of probabilistic analysis.

Modern probability and statistics is supported by computers to perform some of the very large and complex calculations using simulation techniques. All commonly used commercial probabilistic and statistical software packages available these days, e.g., SAS [27], SPSS [28], Microsoft's Excel [8], etc. contain a large collection of discrete and absolutely continuous univariate and multivariate distributions which in turn can be used to form complicated random models. The models can then be analyzed using simulation techniques. These packages are capable of automatically evaluating probabilistic quantities but the results are less accurate. McCullough [18] proposed a collection of intermediate-level tests for assessing the numerical reliability of a statistical package and uncovered flaws in most of the mainframe statistical packages [19] and [20]. Our proposed approach, on the other hand, is capable of determining precise probabilistic quantities at the cost of significant user interaction.

A number of *probabilistic languages*, e.g., `Probabilistic cc` [11], λ_o [23] and `IBAL` [25], have been proposed that are capable of modeling random variables. Probabilistic languages treat probability distributions as primitive data types and abstract from their representation schemes. Therefore, they allow programmers to perform probabilistic computations at the level of probability distributions rather than representation schemes. These probabilistic languages are quite expressive and can be used to perform probabilistic analysis based on the distribution properties of random variables 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 theorem proving based approach proposed in this paper, on the other hand, is capable of modeling most probability distributions due to the high expressive of the higher-order-logic and also provides precise reasoning based on its inherent soundness.

Another alternative for the formal verification of probabilistic properties is to use probabilistic model checking techniques, e.g., [2], [26]. Like the 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 probabilistic properties. This approach is capable of providing precise solutions in an automated way; however it is limited for systems that can only be expressed as a probabilistic finite state machine and is incapable of handling large systems due to the state space explosion [6] problem. Our proposed theorem proving based approach, in contrast, is capable of handling all kinds of probabilistic systems because of the high expressiveness of the higher-order-logic and the verification of probabilistic properties is independent of the size of the model since state space explosion is not an issue.

Hurd's PhD thesis [14] can be regarded as one of the pioneering works in regards to formalizing probabilistic systems in higher-order-logic. The thesis also presents the tools, based on the mathematical probability theory, for reasoning about the correctness of probabilistic systems and this is the area that we extended to verify interval properties of probabilistic systems in HOL. Hurd *et. al* [15] also formalized the *probabilistic guarded-command language* (*pGCL*) in HOL. The *pGCL* contains both demonic and probabilistic nondeterminism and thus makes it suitable for reasoning about distributed random algorithms. Celiku [4] 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.

8 Conclusions

In this paper, we propose to use higher-order-logic theorem proving as a complement to state-of-the-art simulation based approaches for a more reliable and efficient probabilistic analysis framework. The inherent soundness of the theorem-proving based analysis allows us to acquire exact answers to probabilistic properties, which can be expressed in a closed mathematical form, in an interactive manner and is thus quite useful for the analysis of safety critical and highly sensitive sections of the system under test. Simulation techniques, on the other hand, are capable of handling analytically complex sections in an automated way but provide approximate answers and thus can be used to efficiently handle the less critical sections of the system.

We presented a formal definition of the *Cumulative Distribution Function* of random variables along with the verification of its properties in the HOL theorem prover. This is a very significant step towards verification of probabilistic properties in a formalized probabilistic analysis framework, as has been shown in Section 5 of this paper. We also briefly described the formalization of the Standard Uniform random variable in the HOL theorem prover and illustrated with an example that it can be used to formalize other continuous random variables as well.

To the best of our knowledge, the paper presents the first attempt to formally verify the CDF properties in a higher-order-logic theorem prover. For this verification, we utilized the HOL theories of *Sets*, *Boolean Algebra*, *Natural Numbers*, *Real Analysis*, *Measure* and *Probability.* Our results can therefore be used as an evidence for the usefulness of theorem provers in proving pure mathematics and the soundness of the existing HOL libraries. Besides being the first step towards a formalized probabilistic analysis framework, the presented formalization is also a significant step towards an attempt to reconstruct mathematical knowledge in a computer-oriented environment and therefore is also a contribution to the QED project, which calls for a computer system that effectively represents all important mathematical knowledge and techniques [1].

References

1. The QED Manifesto. In: CADE-12: Proceedings of the 12th International Conference on Automated Deduction, pp. 238–251. Springer, Heidelberg (1994)
2. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.P.: Model Checking Algorithms for Continuous time Markov Chains. IEEE Transactions on Software Engineering 29(4), 524–541 (2003)
3. Billingsley, P.: Probability and Measure. John Wiley, Chichester (1995)
4. Celiku, O.: Quantitative Temporal Logic Mechanized in HOL. In: International Colloquium Theoretical Aspects of Computing, pp. 439–453 (2005)
5. Church, A.: A Formulation of the Simple Theory of Types. Journal of Symbolic Logic 5, 56–68 (1940)

6. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. The MIT Press, Cambridge (2000)
7. Devroye, L.: Non-Uniform Random Variate Generation. Springer, Heidelberg (1986)
8. Microsoft Excel (2007) `http://office.microsoft.com`
9. 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)
10. Gordon, M.J.C., Melham, T.F.: Introduction to HOL: A Theorem Proving Environment for Higher-Order Logic. Cambridge University Press, Cambridge (1993)
11. 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)
12. Harrison, J.: Theorem Proving with the Real Numbers. Springer, Heidelberg (1998)
13. Hasan, O., Tahar, S.: Formalization of Standard Uniform Random Variable. Technical Report, Concordia University, Montreal, Canada (December 2006) `http://hvg.ece.concordia.ca/Publications/TECH_REP/SURV_TR06`
14. Hurd, J.: Formal Verification of Probabilistic Algorithms. PhD Thesis, University of Cambridge, Cambridge, UK (2002)
15. Hurd, J., McIver, A., Morgan, C.: Probabilistic Guarded Commands Mechanized in HOL. Theoretical Computer Science 346, 96–112 (2005)
16. Khazanie, R.: Basic Probability Theory and Applications. Goodyear (1976)
17. MacKay, D.J.C.: Introduction to Monte Carlo methods. In: Learning in Graphical Models. NATO Science Series, pp. 175–204. Kluwer Academic Publishers, Dordrecht (1998)
18. McCullough, B.D.: Assessing the Reliability of Statistical Software: Part I. The. American Statistician 52(4), 358–366 (1998)
19. McCullough, B.D.: Assessing the Reliability of Statistical Software: Part II. The. American Statistician 53(2), 149–159 (1999)
20. McCullough, B.D., Wilson, B.: On the Accuracy of Statistical Procedures in Microsoft Excel 2003. Computational Statistics and Data. Analysis 49, 1244–1252 (2005)
21. McShane, E.J.: A Unified Theory of Integration. The. American Mathematical Monthly 80, 349–357 (1973)
22. Milner, R.: A Theory of Type Polymorphism in Programming. Journal of Computer and System Sciences 17, 348–375 (1978)
23. 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)
24. Paulson, L.C.: ML for the Working Programmer. Cambridge University Press, Cambridge (1996)
25. Pfeffer, A.: IBAL: A Probabilistic Rational Programming Language. In: International Joint Conferences on Artificial Intelligence, pp. 733–740. Morgan Kaufmann Publishers, San Francisco (2001)
26. Rutten, J., Kwaiatkowska, M., Normal, G., Parker, D.: Mathematical Techniques for Analyzing Concurrent and Probabilisitc Systems. CRM Monograph, 23, (2004)
27. SAS. (2007) `http://sas.com/technologies/analytics/statistics/stat/index.html`
28. SPSS (2007) `http://www.spss.com/`