Formalization of Discret-Time Markov Chains in HOL

Liya Liu, Osman Hasan, and Sofiène Tahar

Department of Electrical and Computer Engineering,
Concordia University, Montreal, Canada
{liy_liu,o_hasan,tahar}@ece.concordia.ca

Technical Report

April 2011

Abstract

The mathematical concept of Markov chains is widely used to model and analyze many engineering and scienti?c problems. Markovian models are usually analyzed using computer simulation, and more recently using probabilistic model-checking but these methods either do not guarantee accurate analysis or are not scalable. As an alternative, we propose to use higher-order-logic theorem proving to reason about properties of systems that can be described as Markov chains. As the ?rst step towards this goal, this paper presents a formalization of time homogeneous ?nite-state Discrete-time Markov chains and the formal veri?cation of some of their fundamental properties, such as Joint probabilities, Chapman Kolmogorov equation and steady state probabilities, using the HOL theorem prover. For illustration purposes, we utilize our formalization to analyze a simpli?ed binary communication channel.

In probability theory, Markov chains are used to model time varying random phenomena that exhibit the memoryless property [3]. In fact, most of the randomness that we encounter in engineering and scientific domains has some sort of time-dependency. For example, noise signals vary with time, duration of a telephone call is somehow related to the time it is made, population growth is time dependant and so is the case with chemical reactions. Therefore, Markov chains have been extensively investigated and applied for designing systems in many branches of science and engineering. Some of their important applications include functional correctness and performance analysis of telecommunication and security protocols, reliability analysis of hardware circuits, software testing, internet page ranking and statistical mechanics.

Traditionally, simulation has been the most commonly used computer-based analysis technique for Markovian models. The approximate nature of simulation poses a serious problem in highly sensitive and safety critical applications, such as, nuclear reactor control and aerospace software engineering. To improve the accuracy of the simulation results, Markov Chain Monte Carlo (MCMC) methods [15], which involve sampling from desired probability distributions by constructing a Markov chain with the desired distribution, are frequently applied. The major limitation of MCMC is that it generally requires hundreds of thousands of simulations to evaluate the desired probabilistic quantities and becomes impractical when each simulation step involves extensive computations. Other state-based approaches to analyze Markovian models include software packages, such as Markov analyzers and reliability or performance evaluation tools, which are all based on numerical methods [26]. Although these software packages can be successfully applied to analyze large scale Markovian models, the results cannot be guaranteed to be accurate because the underlying iterative methods are not 100% precise. Another technique, *Stochastic Petri Nets* (*SPN*) [9], has been found as a powerful method for modeling and analyzing Markovian systems because it allows local state modeling instead of global modeling. The key limiting factor of the application of SPN models using this approach is the complexity of their analysis.

Formal methods are able to conduct precise system analysis and thus overcome the inaccuracies of the above mentioned techniques. Due to the extensive usage of Markov chains in analyzing safety-critical systems, probabilistic model checking [23] has been recently proposed for analyzing Markov chains. It offers exact solutions but is limited by the state-space explosion problem [2] and the time of analyzing a system is largely dependent on the convergence speed of the underlying algorithms. Similarly, we cannot verify generic mathematical properties using probabilistic model checking due to the inherent state-based nature of the approach. Thus, the probabilistic model checking approach, even though is capable of providing exact solutions automatically, is quite limited in terms of handling a variety of systems and properties.

In this paper, we propose to use higher-order-logic theorem proving [7] as a complementary technique for analyzing Markovian models and thus overcome the limitations of the above mentioned techniques. Time-homogeneousity is an important concept in analyzing Markovian models. In particular, we formalize a time-homogeneous Discrete-Time Markov Chain (DTMC) with finite state space in higher-order logic and then, building upon this definition, formally verify some of the fundamental properties of a DTMC, such as, *Joint Probability Distribution*, *Chapman-Kolmogorov Equation*, and *Steady-state Probabilities* [3]. These properties play a vital role in reasoning about many interesting characteristics while analyzing the Markovian models of real-world systems as well as pave the path to the verification of more advanced properties related to DTMC. In order to illustrate the effectiveness of our work and demonstrate its utilization, we present the formal analysis of a simplified binary communication channel.

1 Related Work

As described above, Markov Analyzers, such as *MARCA* [16] and *DNAmaca* [14], which contain numerous matrix manipulation and numerical solution procedures, are powerful autonomous tools for analyzing large-scale Markovian models. Unfortunately, most of their algorithms are based on iterative methods that begin from some initial approximation and end at some convergent point, which is the main source of inaccuracy in such methods.

Many reliability evaluation software tools integrate simulation and numerical analyzers for modeling and analyzing the reliability, maintainability or safety of systems using Markov methods, which offer simplistic modeling approaches and are more flexible compared to traditional approaches, such as Fault Tree [5]. Some prevalent tool examples are *Möbius* [18] and *Relex Markov* [**?**]. Some other software tools for evaluating performance, e.g. *MACOM* [24] and *HYDRA* [19], take the advantages of a popular Markovian algebra, i.e., *PEPA* [21], to model systems and efficiently compute passage time densities and quantities in large-scale Markov chains. However, the algorithms used to solve the models are based on approximations, which leads to inaccuracies.

Stochastic Petri Nets provide a versatile modeling technique for stochastic systems. The most popular softwares are *SPNP* [4] and *GreatSPN* [8]. These tools can model, validate, and evaluate the distributed systems and analyze the dynamic events of the models using something other than the exponential distribution. Although they can easily manage the size of the system model, the iterative methods employed to compute the stationary distribution or transient probabilities of a model result in inaccurate analysis.

Probabilistic model checking [1, 23] is the state-of-the-art formal Markov chain analysis technique. Numerous probabilistic model checking algorithms and methodologies have been proposed in the open literature, e.g., [6, 20], and based on these algorithms, a number of tools, e.g., *PRISM* [22] and *VESTA* [25] have been developed. They support the analysis of probabilistic properties of DTMC, Continuous-Time Markov chains, Markov decision processes and Semi-Markov Process and have been used to analyze many real-world systems including communication and multimedia protocols. But they suffer from state-space explosion as well as do not support the verification of generic mathematical expressions. Also, because of numerical methods implemented in the tools, the final results cannot be termed 100% accurate. The proposed HOL theorem proving based approach provides another way to specify larger systems and accurate results.

HOL theorem proving has also been used for conducting formal probabilistic analysis. Hurd [13] formalized some measure theory in higher-order logic and proposed an infrastructure to formalize discrete random variables in HOL. Then, Hasan [10] extended Hurd's work by providing the support to formalize continuous random variables [10] and verify the statistical properties, such as, expectation and variance, for both discrete and continuous random variables [10, 11]. Recently, Mhamdi [17] proposed a significant formalization of measure theory and proved Lebesgue integral properties and convergence theorems for arbitrary functions. But, to the best of our knowledge, the current state-of-the-art high-order-logic theorem proving based probabilistic analysis do not provide any theory to model and verify Markov systems and reasoning about their corresponding probabilistic properties. The main contribution of the current paper is to bridge this gap. We mainly build upon Hurd's work to formalize DTMC and verify some of their basic probabilistic properties. The main reason behind choosing Hurd's formalization of probability theory for our work is the availability of formalized discrete and continuous random variables in this framework, as described above. These random variables can be utilized along with our formalization of DTMC to formally represent real-world systems by their corresponding Markovian models in higher-order logic and reason about these models in a higher-order-logic theorem prover.

2 Probability Theory and Random Variables in HOL

A *measure space* is defined as a triple (Ω, Σ, μ) where Ω is a set, called the *sample space*, Σ represents a σ-algebra of subsets of Ω and the subsets are usually referred to as *measurable sets*, and μ is a *measure* with domain Σ. A *probability space* is a measure space $(\Omega, \Sigma, \mathcal{P}r)$ such that the measure, referred to as the probability and denoted by $\mathcal{P}r$, of the sample space is 1.

The measure theory developed by Hurd [13] defines a measure space as a pair (Σ, μ). Whereas the sample space, on which this pair is defined, is implicitly implied from the higher-order-logic definitions to be equal to the universal set of the appropriate data-type. Building upon this formalization, the probability space was also defined in HOL as a pair $(\mathcal{E}, \mathbb{P})$, where the domain of $\mathbb{P}$ is the set $\mathcal{E}$, which is a set of subsets of infinite Boolean sequences $\mathbb{B}^\infty$. 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.

Now, a random variable, which is one of the core concepts in probabilistic analysis, is a fundamental probabilistic function and thus can be modeled in higher-order logic as a deterministic function, which accepts the infinite Boolean sequence as an argument. These deterministic functions make random choices based on the result of popping the top most bit in the infinite Boolean sequence and may pop as many random bits as they need for their computation. When the functions terminate, they return the result along with the remaining portion of the infinite Boolean sequence to be used by other programs. Thus, a random variable which takes a parameter of type α and ranges over values of type β can be represented in HOL by the following function.

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

As an example, consider a Bernoulli($\frac{1}{2}$) random variable that returns 1 or 0 with equal probability $\frac{1}{2}$. It has been formalized in higher-order logic as follows

```
∀ s.  bit s = if shd s then 1 else 0, stl s
```

where the functions `shd` and `stl` are the sequence equivalents of the list operations *'head'* and *'tail'*, respectively. The function `bit` accepts the infinite Boolean sequence s and returns a pair. The first element of the returned pair is a random number that is either 0 or 1, depending on the Boolean value of the top most element of s. Whereas, the second element of the pair is the unused portion of the infinite Boolean sequence, which in this case is the tail of the sequence.

Once random variables are formalized, as mentioned above, we can utilize the formalized probability theory infrastructure to reason about their probabilistic properties. For example, the following Probability Mass Function (PMF) property can be verified for the function `bit` using the HOL theorem prover.

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

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.

The above approach has been successfully used to formally verify most basic probability theorems [13], such as the law of additivity, and conditional probability related properties [12]. For instance, the conditional probability has been formalized as:

Definition: ***Conditional Probability***

```
⊢ ∀ A B.
    cond_prob A B = ℙ(A ⋂ B) / ℙ(B)
```

which plays a vital role in our work. Another frequently used formally verified theorem, in our work, is the *Total Probability Theorem* [12], which is described, for a finite, mutually exclusive, and exhaustive sequence B_i of events and an event A, as follows

$$\mathcal{P}r(A) = \sum_{i=0}^{n-1} \mathcal{P}r(B_i)\mathcal{P}r(A|B_i). \tag{1}$$

We also verified the following closely related property in HOL

$$\mathcal{P}r(B)\mathcal{P}r(A|B) = \mathcal{P}r(A)\mathcal{P}r(B|A) \tag{2}$$

where events A and B are measurable. This property will be used in verifying some important Markov chain properties later.

3 Formal Modeling of Discrete-Time Markov Chains

Given a probability space, a stochastic process $\{X_t : \Omega \to S\}$ represents a sequence of random variables X, where t represents the time that can be discrete (represented by non-negetive integers) or continuous (represented by real numbers) [3]. The set of values taken by each X_t, commonly called states, is referred to as the *state space*. The *sample space* Ω of the process consists of all the possible sequences based on a given state space S. Now, based on these definitions, a *Markov process* can be defined as a stochastic process with Markov property. If a Markov process has finite or countably infinite state space, then it is called a *Markov chain* and satisfies the following Markov property.

For all k and p, if $p < t$, $k < p$ and x_{t+1} and all the states x_i ($i \in [k, t)$) are in the state space, then

$$\begin{aligned}\mathcal{P}r\{X_{t+1} = x_{t+1}|X_t = x_t, \ldots, X_p = x_p \ldots, X_k = x_k\} = \\ \mathcal{P}r\{X_{t+1} = x_{t+1}|X_t = x_t\}.\end{aligned} \tag{3}$$

Additionally, if t ranges over nonnegative integers or, in other words, the time is a discrete quantity, and the states are in a finite state space, then such a Markov chain is called a *Finite-state Discrete-Time Markov Chain.* A Markov chain, if with the same conditional probabilities $\mathcal{P}$r(X_{n+1} = a | X_n = b), is referred to as the *time-homogeneous Markov chain* [3]. Time-homogeneousity is an important concept in analyzing Markovian models and therefore, in our development, we focus on formalizing Time-homogeneous Discrete-Time Markov Chain with finite space, which we refer to in this paper as DTMC. A DTMC is usually expressed by specifying:

- an initial distribution defined by $\pi_0(s) = \mathcal{P}r(X_0 = s)$, $\pi_0(s) \geq 0 (\forall s \in S)$, and $\sum_{s \in S} \pi_0(s) = 1$.

- transition probabilities p_{ij} defined as $\forall i, j \in S$, $p_{ij} = \mathcal{P}r\{X_{t+1} = j | X_t = i\}$, $p_{ij} \geq 0$ and $\sum_{j \in S} p_{ij} = 1$

Based on the above mentioned definitions, we formalize the notion of a DTMC in HOL as the following predicate:

Definition 1:
Time_homogeneous Discrete-Time Markov Chain with Finite state space

```
⊢ ∀ f l x Linit Ltrans.
  Time_homo_mc f l x Linit Ltrans =
  (∀ i.  (i < l) ⇒
     (ℙ{s | FST (f 0 s) = x_i} = EL i Linit) ∧
     (∑_{k=0}^{l-1}(EL i Linit = 1))) ∧
  (∀ t i j.  (i < l) ∧ (j < l) ⇒
     (ℙ{s | FST (f (t + 1) s) = x_j}|{s | FST (f t s) = x_i} =
               (EL (i * l + j) Ltrans)) ∧
     (∑_{k=0}^{l-1}(EL (i * l + k) Ltrans = 1))) ∧
  (∀ t k.  (k < l) ⇒ measurable {s | FST (f t s) = x_k}) ∧
  (∀ t.  ⋃_{k=0}^{l-1} {s | FST (f t s) = x_k} = UNIV) ∧
  (∀ t u v.  (u < l) ∧ (v < l) ∧ (u ≠ v) ⇒
     disjoint ({s | FST (f t s) = x_u} {s | FST (f t s) = x_v})) ∧
  (∀ i j m r t w L Lt.
    ((∀ k.  (k ≤ r) ⇒ (EL k L < l)) ∧ (i < l) ∧ (j < l) ∧
    (Lt ⊆ [m, r]) ∧ (m ≤ r) ∧ (w + r < t)) ⇒
    (ℙ({s | FST (f (t + 1) s) = x_j}|{{s | FST (f t s) = x_i} ⋂
       (⋂_{k∈Lt} {s | FST (f (w + k) s) = x_(EL k L)})}) =
     ℙ({s | FST (f (t + 1) s) = x_j}|{s | FST (f t s)= x_i}))) ∧
  (∀ t n i j.
      (i < l) ∧ (j < l) ⇒
      (ℙ({s | FST (f (t + 1) s) = x_j}|{s | FST (f t s) = x_i}) =
       ℙ({s | FST (f (n + 1) s) = x_j}|{s | FST (f n s) = x_i})))
```

The function `Time_homo_mc` accepts a sequence of random variables `f`, the cardinality of the set of their possible states `l`, a function `x` that accepts the index and returns the state corresponding to the given DTMC, and two real lists: the initial states probability distribution `Linit` and the transition probabilities `Ltrans`.

The predicate `Time_homo_mc` adheres to following five conditions:

- the DTMC must follow the given initial distribution `Linit`, in which the summation of all the elements is 1. The transition probabilities `Ltrans`, in which the summation of each l elements is 1, is an intrinsic characteristic of a stochastic matrix.

- all events involving the Markov chain random variables are measurable (`∀ t k. (k < l) ⇒ measurable {s | FST (f t s) =` x_k`}`).

- the union of all states forms the state space as a universal set `UNIV` (`∀ t.` $\bigcup_{k=0}^{l-1}$ `{s | FST (f t s) =` x_k`} = UNIV`).

- the fifth condition ensures that the states in the state space of a given Markov chain are mutually exclusive (`∀ t u v. (u < l) ∧ (v < l) ∧ (u ≠ v) ⇒ disjoint ({s | FST (f t s) =` x_u`} {s | FST (f t s) =` x_v`})`).

- the sixth condition corresponds to the memoryless property in Equation (3). Mathematically, if x_{t+1}, x_t, x_i and x_j are the states in the state space, and $w + k < t$, then the following equation holds

$$\begin{aligned} \mathcal{P}r\{X_{t+1} = x_{t+1} | X_t = x_t, \ldots, X_{w+k} = x_i, X_k = x_j, \ldots\} = \\ \mathcal{P}r\{X_{t+1} = x_{t+1} | X_t = x_t\}. \end{aligned} \tag{4}$$

  We model history of states in our formalization by a list L, which contains the state elements ranging from 0 to $l-1$. Thus, the list L, with $r + 1$ elements or less, represents the indices of passed states and its elements have to be less than `l` (`∀ k. (k ≤ r) ⇒ (EL k L < l)`). In ($\bigcap_{k \in Lt}$ `{s | FST (f (w + k) s) =` $x_{(EL\ \ k\ \ L)}$`}`), where the function (`EL k L`) returns the k^{th} element of the list L, it gives a general time index of every event and a flexible length of the event sequence. (`k ϵ Lt`) makes sure that the passed states can be freely chosen from a set `Lt`, which includes natural numbers and is a subset of the interval [m, r] (`Lt ⊆ [m, r]`). Condition (`w + r < t`) ensures that the states in this intersection set are passed states.

- the last condition represents the time homogeneousity of a discrete-time Markov chain f.

It is important to note that for generality our definition can work with discrete-time random variables of any data type.

4 Verification of Discrete-Time Markov Chain Properties

In this section, we present the formal verification of the most important properties of discrete-time Markov Chain.

4.1 Joint Probability of a Markov Chain

The joint probability of a Markov chain defines the probability of events involving two or more random variables associated with a chain. Joint probability is very useful in analyzing multi-stage experiments when an event chain happens, and reversible stochastic processes. Also, this concept is the basis for joint probability generating function, which is used in many different fields. Mathematically, the joint probability of $n + 1$ discrete random variables X_0, X_1, …, X_n in a Markov chain can be expressed as [3]:

$$\begin{aligned} \mathcal{P}r\{X_t = x_0, \cdots, X_{t+n} = x_n\} = \\ \prod_{k=0}^{n-1} \mathcal{P}r\{X_{t+k+1} = x_{k+1} | X_{t+k} = x_k\} \mathcal{P}r\{X_t = x_0\}. \end{aligned} \tag{5}$$

In Equation (5), $\mathcal{P}r\{X_{t+k+1} = x_{k+1} | X_{t+k} = x_k\}$ can be found in the given one-step transition probabilities.

We formalize this property in HOL as following theorem:

Theorem 1: *Joint Probability*

```
⊢ ∀ f l x t n L Linit Ltrans.
    (Time_homo_mc f l x Linit Ltrans) ∧
    (EVERY (λa.  a < l) L) ∧ (n + 1 ≤ LENGTH L) ⇒
```

$$\begin{aligned}
&\mathbb{P}(\textstyle\bigcap_{k=0}^{n}\{\mathtt{s\ \mid\ FST\ (f\ (t\ +\ k)\ s)\ =\ }x_{(EL\ \ k\ \ L)}\})\ \mathtt{=}\\
&\textstyle\prod_{k=0}^{n-1}\mathbb{P}(\{\mathtt{s\ \mid\ FST\ (f\ (t\ +\ k\ +\ 1)\ s)\ =\ }x_{(EL\ \ (k+1)\ \ L)}\}\mid\\
&\qquad\quad \{\mathtt{s\ \mid\ FST\ (f\ (t\ +\ k)\ s)\ =\ }x_{(EL\ \ k\ \ L)}\})\\
&\qquad \mathbb{P}\{\mathtt{s\ \mid\ FST\ (f\ t\ s)\ =\ }x_{(EL\ \ 0\ \ L)}\}
\end{aligned}$$

The variables above are used in the same context as Definition 1. The first assumption ensures that f is a Markov chain. All the elements of the indices sequence L are less than l and the length of L is larger than or equal to the length of the segment considered in the joint events. The conclusion of the theorem represents Equation (5) in higher-order logic based on the probability theory formalization, presented in Section 2. The proof of Theorem 1 is based on induction on the variable n, Equation (1) and some arithmetic reasoning.

4.2 Chapman-Kolmogorov Equation

The well-known Chapman-Kolmogorov equation [3] is a widely used property of time homogeneous Markov chains as it facilitates the use of a matrix theory for analyzing large Markov chains. It basically gives the probability of going from state i to j in $m+n$ steps. Assuming the first m steps take the system from state i to some intermediate state k, which is in the state space Ω and the remaining n steps then take the system from state k to j, we can obtain the desired probability by adding the probabilities associated with all the intermediate steps.

$$p_{ij}(m+n) = \sum_{k\in\Omega} p_{kj}(n)p_{ik}(m) \tag{6}$$

The notation $p_{ij}(n)$ denotes the n-step transition probabilities from state i to j.

$$p_{ij}(n) = \mathcal{P}r\{X_{t+n} = x_j | X_t = x_i\} \tag{7}$$

Based on Equation (6), and Definition 1, the Chapman-Kolmogorov equation is formalized as follows

Theorem 2: *Chapman-Kolmogorov Equation*

$$\begin{aligned}
&\vdash \forall\ \mathtt{f\ i\ j\ x\ l\ m\ n\ Linit\ Ltrans.}\\
&\quad \mathtt{(Time\_homo\_mc\ f\ l\ x\ Linit\ Ltrans)}\ \wedge\ \mathtt{(i\ <\ l)}\ \wedge\ \mathtt{(j\ <\ l)}\ \wedge\\
&\quad (\forall\ \mathtt{r.\ \ (r\ <\ l)}\ \Rightarrow\ \mathtt{(0\ <\ }\mathbb{P}\{\mathtt{s\ \mid\ FST\ (f\ 0\ s)\ =\ }x_r\}))\ \Rightarrow\\
&\quad \mathbb{P}(\{\mathtt{s\ \mid\ FST\ (f\ (m\ +\ n)\ s)\ =\ }x_j\}\mid\{\mathtt{s\ \mid\ FST\ (f\ 0\ s)\ =\ }x_i\})\ \mathtt{=}\\
&\quad \textstyle\sum_{k=0}^{l-1}(\mathbb{P}(\{\mathtt{s\ \mid\ FST\ (f\ n\ s)\ =\ }x_j\}\mid\{\mathtt{s\ \mid\ FST\ (f\ 0\ s)\ =\ }x_k\})\\
&\qquad\qquad \mathbb{P}(\{\mathtt{s\ \mid\ FST\ (f\ m\ s)\ =\ }x_k\}\mid\{\mathtt{s\ \mid\ FST\ (f\ 0\ s)\ =\ }x_i\}))
\end{aligned}$$

The variables m and n denote the steps between two states and both of them represent time. The first assumption ensures that the random process f is a time homogeneous DTMC, using Definition 1. The following two assumptions, $i < l$ and $j < l$, define the allowable bounds for the index variables. The last assumption is used to exclude the case when $\mathcal{P}r\{X_0 = x_j\} = 0$. Because it makes no sense to analyze the conditional probability when the probability of a state existing is 0. The conclusion of the theorem formally represents Equation (6).

The proof of Theorem 2 again involves induction on the variable n and both of the base and step cases are discharged using the following lemma.

Lemma 1: *Multistep Transition Probability*

$$\vdash \forall\ \mathtt{f\ i\ j\ x\ n\ Linit\ Ltrans.}$$

```
  (Time_homo_mc f l x Linit Ltrans) ∧ (i < l) ∧ (j < l) ∧
  (0 < ℙ{s | FST (f 0 s) = $x_i$}) ⇒
  ℙ({s | FST (f (n + 1) s) = $x_j$}|{s | FST (f 0 s) = $x_i$}) =
  $\sum_{k=0}^{l-1}$ℙ({s | FST (f 1 s) = $x_j$}|{s | FST (f 0 s) = $x_k$})
       ℙ({s | FST (f n s) = $x_k$}|{s | FST (f 0 s) = $x_i$})
```

The proof of Lemma 1 is primarily based on the Total Probability theorem (1).

4.3 Absolute Probabilities

The unconditional probabilities associated with a Markov chain are referred to as the absolute probabilities [3]. If the initial probability distribution of the system being in a state, which has index k is given by $\mathcal{P}r\{X_0 = x_k\}$, then the absolute probability of the system being in state j is given by

$$p_j(n) = \mathcal{P}r\{X_n = x_j\} = \sum_{k=0}^{l-1} \mathcal{P}r\{X_0 = x_k\}\mathcal{P}r\{X_n = x_j | X_0 = x_k\}. \quad (8)$$

This shows that, given an initial probability distribution and the n-step transition probabilities, the absolute probabilities in the state j after n step from the start time 0 can be obtained by using this equation.

Based on our formal Markov chain definition, this property has been formalized as the following theorem:

Theorem 3: *Absolute Probability*

```
⊢ ∀ f j x l n t Linit Ltrans.
  (Time_homo_mc f l x Linit Ltrans) ∧ (j < l) ∧
  (∀ r.  (r < l) ⇒ (0 < ℙ{s | FST (f 0 s) = $x_r$})) ⇒
  ℙ{s | FST (f n s) = $x_j$} =
  $\sum_{k=0}^{l-1}$ℙ{s | FST (f 0 s) = $x_k$}
       ℙ({s | FST (f n s) = $x_j$}|{s | FST (f 0 s) = $x_k$})
```

The proof of Theorem 3 is based on the Total Probability theorem along with some basic arithmetic and probability theoretic reasoning.

4.4 Steady State Probabilities

In many applications, analyzing the stability of Markovian models is of prime importance. For example, we are interested in the probability of states as time tends to infinity under certain conditions, like irreducibility and aperiodicity.

Let X_n, $n \geq 0$, be a Markov chain having state space Ω and one-step transition probability $P(x, y)$ for going from state with value x to a state with value y. If $\pi(x)$, $x \in \Omega$, are nonnegative numbers summing to one, and if

$$\pi(y) = \sum_{x \in \Omega} \pi(x)P(x, y), y \in \Omega \quad (9)$$

then π is called a *stationary distribution*. The corresponding HOL definition is as follows. In this definition, x_k and x_i represent the variables x and y of Equation (9), respectively.

Definition 2: *Stationary Distribution*

```
⊢ ∀ p f n x l.  stationary_dist p f n x l =
  ∀ i.
    (0 ≤ (p $x_i$)) ∧($\sum_{k=0}^{l-1}$ (p $x_k$) = 1) ∧
    (p $x_i$ = $\sum_{k=0}^{l-1}$(p $x_k$)$\mathbb{P}$({s | FST (f (n + 1) s) = $x_i$}|
                                    {s | FST (f n s) = $x_k$}))
```

As a finite Markov chain, the steady state probabilities are defined to be a vector $V_j = \lim_{n \to \infty} \mathbb{P}(n)$. For a time homogeneous finite Markov chain with one-step transition probability $P(x, y)$, if V_j exists for all j $\in \Omega$, then V_j is known as the stationary probability vector of that Markov chain. In other words, V_j is a stationary distribution of a Markov chain if

- $\lim_{n \to \infty} p_j(n) = \sum_{i=0}^{l-1} \lim_{n \to \infty} p_i(n) p_{ij}$, $j = 0, 1, 2, \cdots,$ (l - 1)
- $\sum_{i=0}^{l-1} \lim_{n \to \infty}\ p_i(n) = 1$
- $0 \leq \lim_{n \to \infty}\ p_j(n)$

The steady state probability is formalized in HOL as follows

Theorem 4: *Steady State Probability*

```
⊢ ∀ f n x l Linit Ltrans.
  (Time_homo_mc f l x Linit Ltrans) ∧
  (∀ x j.  ∃u.  $\mathbb{P}${s | FST (f n s) = $x_j$} → u) ⇒
  (stationary_dist (λx k.  $\lim_{n \to \infty}\mathbb{P}${s | FST (f n s) = $x_k$}) f n x l)
```

The proof of Theorem 4 is primarily based on the linearity of limit of a sequence and the linearity of real summation.

4.5 Generalized Stationary Distribution

If a Markov chain with state space Ω and one-step transition probability $P(x, y)$ has a probability π that satisfies the detailed balance equations, given below, then this distribution π is stationary for $P(x, y)$. This theorem is called a *generalized stationary theorem* and can be mathematically described as follows:

$$\pi(x)P(x, y) = \pi(y)P(y, x), \forall x, y \in \Omega \tag{10}$$

The detailed balance equations can be formalized as follows, where x_i and x_j represent variables x and y of Equations (10), respectively.

Definition 3: *Detailed Balance Equations*

```
⊢ ∀ p f l.  db_equations p f l =
  ∀ x i j n.
   (i < l) ∧ (j < l) ∧
   ((p $x_i$)$\mathbb{P}$({s | FST (f (n + 1) s) = $x_j$}|{s | FST (f n s) = $x_i$}) =
     (p $x_j$)$\mathbb{P}$({s | FST (f (n + 1) s) = $x_i$}|{s | FST (f n s) = $x_j$}))
```

The first input variable p in the above predicate is a function that accepts the state as the parameter and returns the probability given in Equation (10). Based on this definition, the stationary theorem can be defined as follows:

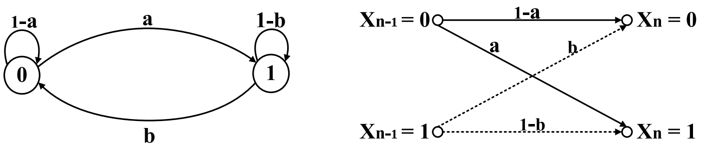

Fig. 1: State Diagram and Channel Diagram of the Binary Channel Model

Theorem 5: *Generalized Stationary Distribution*
$\vdash \forall$ f x l n Linit Ltrans.
  (db_equations (λx i. $\mathbb{P}\{$s | FST (f n s) = $x_i\}$) f l) $\wedge$
  (Time_homo_mc f l x Linit Ltrans) $\Rightarrow$
  (stationary_dist (λx k. $\mathbb{P}\{$s | FST (f n s) = $x_k\}$) f n x l)

Here, $\pi(x)$ is specified as a function $\lambda x\ i.\ \mathbb{P}\{$s | FST (f n s) $= x_i\}$. The proof of Theorem 5 is based on the Total Probability theorem, given in Equation (1), and the following Lemma:

Lemma 3: *Summation of Transition Probability*
$\vdash \forall$ f x l i n Linit Ltrans.
  (Time_homo_mc f l x Linit Ltrans) $\wedge$ (i $<$ l) $\Rightarrow$
  $\sum_{j=0}^{l-1}\mathbb{P}$($\{$s | FST (f n s) = $x_j\}$|$\{$s | FST (f 0 s) = $x_i\}$ = 1

The proof script[1]for the formalization of Markov chain, presented in this section, consists of approximately 2600 lines of HOL code. These results not only ensure the correctness of our formal Markov chain definitions, presented in Section 3, but also play a vital role in analyzing real-world systems that are modeled by DTMC, as will be demonstrated in the next section.

5 Application: Binary Communication Channel

In order to illustrate the usefulness of the proposed approach, we use our results to analyze a simplified binary communication channel model [27]. Also, we compare the analysis of the same example using probabilistic model checking.

A binary communication channel is a channel with binary inputs and outputs. The transmission channel is assumed to be noisy or imperfect, i.e., it is likely that the receiver gets the wrong digit. This channel can be modeled as a two-state time homogenous DTMC with the following state transition probabilities.

$\mathcal{P}$r$\{X_{n+1}$ = 0 | X_n = 0$\}$ = 1 - a; $\mathcal{P}$r$\{X_{n+1}$ = 1 | X_n = 0$\}$ = a;
$\mathcal{P}$r$\{X_{n+1}$ = 0 | X_n = 1$\}$ = b; $\mathcal{P}$r$\{X_{n+1}$ = 1 | X_n = 1$\}$ = 1 - b

The corresponding state diagram and channel diagram are given in Fig. 1. The binary communication channel is widely used in telecommunication theory as more complicated channels are modeled by cascading several of them. Here, variables X_{n-1} and X_n denote the digits leaving the systems $(n-1)^{th}$ stage and entering the n^{th} one, respectively. a and b are the crossover bit error probabilities. Because variables X_0 is also a random variable, the initial state is not determined, $\mathcal{P}$r $(X_0 = 0)$ and $\mathcal{P}$r $(X_0 = 1)$ could not be 0 or 1.

Although the initial distribution is unknown, the given binary communication channel has been formalized in HOL as a generic model, using Definition 2.

[1] Available at http://users.encs.concordia.ca/∼liy_liu/code.html

Definition 4: *Binary Communication Channel Model*

```
⊢ ∀ f x a b p q.
  (binary_communication_channel_model f a b p q) =
    (Time_homo_mc f (2:num) x [p; q] [1 - a; a; b; 1 - b]) ∧
    (|1 - a - b| < 1) ∧ (0 ≤ a ≤ 1) ∧ (0 ≤ b ≤ 1) ∧
    (p + q = 1) ∧ (0 < p < 1) ∧ (0 < q < 1)
```

In this formal model, variable f represents the Markov chain and variables a, b, p and q are parameters of the functions of initial distribution and transition probabilities. The variable x represents a function that provides the state at a given index.

The first condition ensures that f is a time-homogeneous DTMC, with which the number of states l is 2, because there are only two states in the state space. List [p; q] corresponds to Linit in Definition 1 and another list [1 - a; a; b; 1 - b] gives the one-step transition probability matrix by combining all the rows into a list and corresponds to Ltrans in Definition 1. The next three conditions define the allowable intervals for parameters a and b to restrict the probability terms in [0,1]. It is important to note that, |1 - a - b| < 1 ensures that both a and b cannot be equal to 0 and 1 at the same time and thus avoids the zero transition probabilities. The remaining conditions correspond to one-step transition probabilities.

Next, we use our formal model to reason about the following properties.

Theorem 6: *n^{th} step Transition Probabilities*

⊢ ∀ f x a b n p q.
(binary_communication_channel_model f x a b p q) ⇒
$(\mathbb{P}(\{\texttt{s|FST (f n s)=}x_0\}\texttt{|}\{\texttt{s|FST (f 0 s))=}x_0\})\texttt{=}\frac{b+a(1-a-b)^n}{a+b})\ \wedge$
$(\mathbb{P}(\{\texttt{s|FST (f n s)=}x_1\}\texttt{|}\{\texttt{s|FST (f 0 s))=}x_0\})\texttt{=}\frac{a-a(1-a-b)^n}{a+b})\ \wedge$
$(\mathbb{P}(\{\texttt{s|FST (f n s)=}x_0\}\texttt{|}\{\texttt{s|FST (f 0 s))=}x_1\})\texttt{=}\frac{b-b(1-a-b)^n}{a+b})\ \wedge$
$(\mathbb{P}(\{\texttt{s|FST (f n s)=}x_1\}\texttt{|}\{\texttt{s|FST (f 0 s))=}x_1\})\texttt{=}\frac{a+b(1-a-b)^n}{a+b})$

Theorem 7: *Limiting State Probabilities*

⊢ ∀ f x a b p q.
(binary_communication_channel_model f x a b p q) ⇒
$(\lim_{n\to\infty}\mathbb{P}(\{\texttt{s|FST (f n s)=}x_0\}\texttt{|}\{\texttt{s|FST (f 0 s))=}x_0\})\texttt{=}\frac{b}{a+b})\ \wedge$
$(\lim_{n\to\infty}\mathbb{P}(\{\texttt{s|FST (f n s)=}x_1\}\texttt{|}\{\texttt{s|FST (f 0 s))=}x_0\})\texttt{=}\frac{a}{a+b})\ \wedge$
$(\lim_{n\to\infty}\mathbb{P}(\{\texttt{s|FST (f n s)=}x_0\}\texttt{|}\{\texttt{s|FST (f 0 s))=}x_1\})\texttt{=}\frac{b}{a+b})\ \wedge$
$(\lim_{n\to\infty}\mathbb{P}(\{\texttt{s|FST (f n s)=}x_1\}\texttt{|}\{\texttt{s|FST (f 0 s))=}x_1\})\texttt{=}\frac{a}{a+b})$

Theorem 6 has been verified by performing induction on n and then applying Theorem 2 and Lemma 3 and along with some arithmetic reasoning. Theorem 6 is then used to verify Theorem 7 along with limit of real sequence principles.

Now, we modeled the binary communication channel in the PRISM probabilistic model checker and tried to evaluate the same nth-step transition probabilities, given in Theorems 4 to 7. For this purpose, we have to specify exact values for probabilities a and b and the number of steps n. We experimented by sweeping variables a and b in the interval [0; 1] with a step size 0.1, and setting $n = 4$. The results are depicted graphically in Figure 2.

This small case study clearly illustrates the main strength of the proposed theorem proving based technique against the probabilistic model checking approach. In the proposed approach, we verified the desired probabilistic characteristics as generic theorems that are

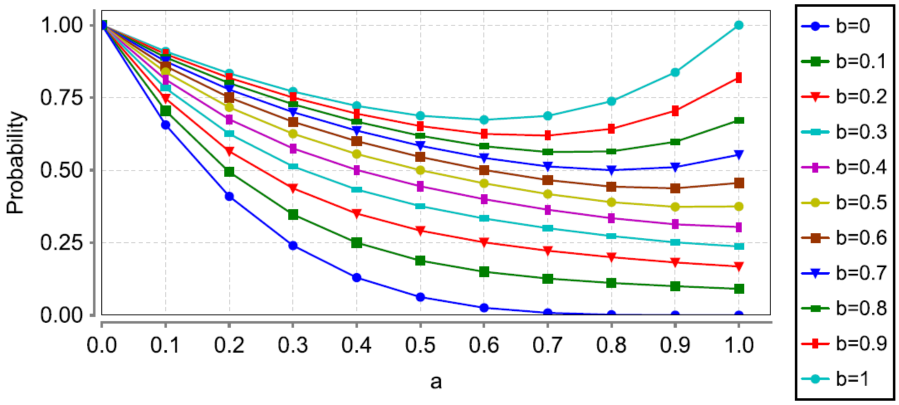

Fig. 2: The 4-step transition probabilities with resolution 0.1

universally quantified for all allowable values of variables a, b and n. Thus, we can evaluate the exact probabilities corresponding to any particular set of these three parameters by simply substituting the desired values in the corresponding theorems. On the other hand, probabilistic model checking provides solutions for a particular set of given parameters and thus we cannot have results for all the possible values of the parameters, which kind of introduces some degree of inaccuracy in the results. Table 1 illustrates this point by providing the analysis time against the resolution of the analysis using a dual quad-core SunRay server running at 2.75GHz with 8GB of RAM. We observed that the analysis time increases exponentially with an increase in the resolution and PRISM was not able to handle resolutions beyond 0.0001. Now, it is important to note here that if we cannot go beyond a resolution of 0.0001 for our case study that is represented by a very small 2-state Markov chain, the erect of increasing the resolution would be much more profound in the case of analyzing larger systems. State-space explosion is another major limitation of probabilistic model checking, as mentioned earlier. Though, we did not experience this problem for our relatively small model for moderate resolutions but this would also become a major bottleneck as the system size increases.

6 Conclusions

This report presents the formal verification of discrete-time Markov chain using theorem proving. The state of the art in the area of analysis of Markov chain models shows a lot of contributions on related research. Hurd [13] developed an infrastructure to specify and verify probabilistic algorithms based on a formalized probabilistic space. Building upon Hurd's formalization, most of commonly used discrete and continuous random variables [10] [11] have been formalized. Their work is fundamental to the proposed formalization of both DTMC and CTMC. We built upon the formalization of discrete-time Markov chains and verified three of the most fundamental discrete-time Markov chains properties using the HOL theorem prover. This exercise convinces us that it is feasible to build up an infrastructure to integrate higher-order-logic theorem proving in the domain of analysis of Markov chain based system models, as a complementary approach to those simulation based or numerical methods based techniques.

For illustration purposes, we analyzed a binary communication channel. Our results exactly matched the corresponding paper-and-pencil based analysis, which ascertains the precise nature of the proposed approach. Some more applications will be verified in the future based on our proposed approach.

To the best of our knowledge, this report proposes research that is the first of its kind and opens the doors for a new, but very promising research direction for the modeling and verification of systems, which behavior can be expressed as Markov chains.

References

[1] C. Baier, B. Haverkort, H. Hermanns, and J.P. Katoen. Model Checking Algorithms for Continuous Time Markov Chains. *IEEE Transactions on Software Engineering*, 29(4):524–541, 2003.

[2] C. Baier and J. Katoen. *Principles of Model Checking.* MIT Press, 2008.

[3] R. N. Bhattacharya and E. C. Waymire. *Stochastic Processes with Applications.* John Wiley & Sons, 1990.

[4] G. Ciardo, J. K. Muppala, and K. S. Trivedi. SPNP: Stochastic Petri Net Package. In *Workshop on Petri Nets and Performance Models*, pages 142–151, 1989.

[5] M. Tessmer D. H. Jonassen and W. H. Hannum. *Task Analysis Methods for Instructional Design.* Lawrence Erlbaum, 1999.

[6] L. de Alfaro. *Formal Verification of Probabilistic Systems.* PhD Thesis, Stanford University, Stanford, USA, 1997.

[7] M.J.C. Gordon. Mechanizing Programming Logics in Higher-0rder Logic. In *Current Trends in Hardware Verification and Automated Theorem Proving*, pages 387–439. Springer, 1989.

[8] GreatSPN. http://www.di.unito.it/∼greatspn/index.html, 2011.

[9] P. J. Haas. *Stochastic Petri Nets: Modelling, Stability, Simulation.* Springer, 2002.

[10] O. Hasan. *Formal Probabilistic Analysis using Theorem Proving.* PhD Thesis, Concordia University, Montreal, QC, Canada, 2008.

[11] O. Hasan, N. Abbasi, B. Akbarpour, S. Tahar, and R. Akbarpour. Formal Reasoning about Expectation Properties for Continuous Random Variables. In *Formal Methods*, volume 5850 of *LNCS*, pages 435–450. Springer, 2009.

[12] O. Hasan and S. Tahar. Reasoning about Conditional Probabilities in a Higher-Order-Logic Theorem Prover. *Journal of Applied Logic*, 9(1):23 – 40, 2011.

[13] J. Hurd. *Formal Verification of Probabilistic Algorithms.* PhD Thesis, University of Cambridge, UK, 2002.

[14] W. J. Knottenbelt. Generalised Markovian Analysis of Timed Transition Systems. Master's thesis, Department of Computer Science, University of Cape Town, South Africa, 1996.

[15] D.J.C. MacKay. Introduction to Monte Carlo Methods. In *Learning in Graphical Models, NATO Science Series*, pages 175–204. Kluwer Academic Press, 1998.

[16] MARCA. http://www4.ncsu.edu/∼billy/MARCA/marca.html, 2011.

[17] T. Mhamdi, O. Hasan, and S. Tahar. On the Formalization of the Lebesgue Integration Theory in HOL. In *Interactive Theorem Proving*, volume 6172 of *LNCS*, pages 387–402. Springer, 2010.

[18] Mobius. http://www.mobius.illinois.edu/, 2011.

[19] W. J. Knottenbelt N. J. Dingle and P. G. Harrison. HYDRA - Hypergraph-based Distributed Response-time Analyser. In *International Conference on Parallel and Distributed Processing Technique and Applications*, pages 215 – 219, 2003.

[20] D. Parker. *Implementation of Symbolic Model Checking for Probabilistic System.* PhD Thesis, University of Birmingham, UK, 2001.

[21] PEPA. http://www.dcs.ed.ac.uk/pepa/, 2011.

[22] PRISM. http://www.prismmodelchecker.org, 2011.

[23] J. Rutten, M. Kwaiatkowska, G. Normal, and D. Parker. *Mathematical Techniques for Analyzing Concurrent and Probabilisitc Systems*, volume 23 of *CRM Monograph Series.* American Mathematical Society, 2004.

[24] M. Sczittnick. MACOM - A Tool for Evaluating Communication Systems. In *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 7–10, 1994.

[25] K. Sen, M. Viswanathan, and G. Agha. VESTA: A Statistical Model-Checker and Analyzer for Probabilistic Systems. In *IEEE International Conference on the Quantitative Evaluation of Systems*, pages 251–252, 2005.

[26] W. J. Steward. *Introduction to the Numerical Solution of Markov Chain.* Princeton University Press, 1994.

[27] K. S. Trivedi. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications.* John Wiley & Sons, 2002.