On the Formal Analysis of HMM Using Theorem Proving

Liya Liu, Vincent Aravantinos, Osman Hasan, and Sofiène Tahar

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

Abstract. Hidden Markov Models (HMMs) have been widely utilized for modeling time series data in various engineering and biological systems. The analyses of these models are usually conducted using computer simulations and paper-and-pencil proof methods and, more recently, using probabilistic model-checking. However, all these methods either do not guarantee accurate analysis or are not scalable (for instance, they can hardly handle the computation when some parameters become very huge). As an alternative, we propose to use higher-order logic theorem proving to reason about properties of discrete HMMs by applying automated verification techniques. This paper presents some foundational formalizations in this regard, namely an extended-real numbers based formalization of finite-state Discrete-Time Markov chains and HMMs along with the verification of some of their fundamental properties. The distinguishing feature of our work is that it facilitates automatic verification of systems involving HMMs. For illustration purposes, we utilize our results for the formal analysis of a DNA sequence.

Keywords: HMMs, HOL4, Theorem Proving, DNA, Probability Theory.

1 Introduction

Hidden Markov Models (HMMs) [16] provide a useful statistical method for analyzing random processes based on their observable output samples. As their name suggests, HMMs assume that the observed samples are generated by a Markov process [3], for which the states are hidden from the observer. Initially HMMs were proposed to solve optimal linear filtering problems as the simplest dynamic Bayesian networks [27]. However, due to their usefulness in effectively analyzing probability distributions over a sequence of observations, HMMs are now extensively used in many applications involving speech recognition, cryptanalysis, molecular biology, data compression, financial market forecasting and artificial intelligence.

Traditionally, simulation has been the most commonly used computer-based analysis technique for HMMs. Based on this technique, HMMs are used to solve three types of problems: 1) evaluating the probability of occurrence of a particular observed sequence; 2) finding the most probable state sequence to generate

S. Merz and J. Pang (Eds.): ICFEM 2014, LNCS 8829, pp. 316–331, 2014.

given observations; and 3) learning parameters in the presumed model. These problems are typically solved by applying complex algorithms, like Forward-Backward, Viterbi, or Baum-Welch algorithms [16], whose implementations are usually not formally verified. This fact, along with the inherent limitations of computer simulation, like usage of computer arithmetic and pseudo random numbers, makes the analysis of HMMs approximate and thus the analysis based on HMMs becomes unreliable. This problem can have severe consequences when it comes to analyzing critical applications like Electrocardiogram Signal Processing [7] or Computational Biology [8], which is mainly used in determining and/or analyzing cancer, tumor and human genome. The analysis results directly affect the treatment of patients and their lifetime.

Formal methods allow to overcome the above mentioned limitations. For instance, probabilistic model checking guarantees precise system analysis by modeling the system behavior, including its random components, in a given logic and reasoning about its probabilistic properties. Some model checking algorithms have been proposed for analyzing HMMs [26]. However, the state-space explosion problem [2] limits the usage of probabilistic model checking to a very small subset of HMM applications. In addition, it cannot verify generic mathematical expressions for probabilistic analysis. Finally, the proposed model checking algorithms for HMMs in [25] are also complex and make use of many optimizations that are difficult to verify, and thus force the user to trust the developer of a given model checker.

The other widely used formal method is theorem proving [10], which provides a conceptually simple formalism with a precise semantics and can express all classical mathematical theories. Due to the highly expressive nature of higher-order logic and the inherent soundness of interactive theorem proving tools, this technique can provide precise analysis of HMMs. Although three chapters of measure theory were formalized in Isabelle/HOL [13] and the formalization of probability theory was simplified in Coq [6][1], to the best of our knowledge, foundational mathematics for HMMs has not been formalized in higher-order logic. Moreover, the interactive nature of higher-order logic theorem proving makes it quite unattractive for engineers and scientists involved in analyzing HMMs. This is one of the main reasons why theorem proving has not been used for the analysis of HMMs despite its ability to provide exact answers.

In this paper, we address both of the concerns mentioned above to facilitate the formal analysis of HMMs using theorem proving. Firstly, we present a higher-order logic formalization of mathematical foundations for HMMs. This includes the formalization of discrete time Markov chains (DTMCs), HMMs and the formal verification of some of their widely used properties. Our formalization of DTMCs is an improved version of the formalization of DTMCs presented in [15] since it is based on a more general probability theory and can handle inhomogeneous DTMCs with generic state spaces, which are the foremost prerequisites for modeling HMMs. Our formalization of HMMs also allows to reduce user intervention in formal modelling and analysis of real-world systems that can be expressed in terms of HMMs. The main challenge of this work is to express the

conditional independency of two stochastic processes in higher-order logic. To facilitate this process further, we introduce some automatic simplifiers to make the proposed method a very practical solution for the formal analysis of HMMs. For illustration purposes, we present a case study about DNA sequence analysis.

2 Related Work

Various simulation-based HMM analysis tools, dedicated to a particular system domain, have been reported in the literature. Some prominent examples include *HMMTool* [12] as part of the *NHMMtoolbox* [21] to predict daily rainfall sequence. *ChIP-Seq* [4], MArkov MOdeling Tool (*MAMOT*) [9] and *HMMER* [11] are some of the popular simulation software in biological research. As mentioned in the previous section, due to their approximate nature, all these simulation techniques are not reliable enough for critical applications.

Probabilistic model checking [22] is the state-of-the-art formal Markov chain analysis technique. Numerous model checkers, e.g., *PRISM* [20], *VESTA* [23], *MRMC* [18], *Ymer* [24], etc., are available and have been used to analyze a variety of systems. In [25], the author defined probability spaces for modeling HMMs and presented model checking algorithms using Probabilistic Observation CTL (POCTL) for specifying properties of parameterized HMMs. The complexity of these algorithms depends on the size of the model and the number of variables involved in the property formula. This factor, coupled with the inherent nature of model checking, severely limits the usage of this algorithm for analyzing real-world examples. In addition, no HMM can be analyzed by model checker PRISM.

Higher-order-logic theorem proving overcomes the limitations of model checking and has been used to successfully formalize DTMCs [15]. However this formalization was not general enough to formalize HMMs. This was due to the fact that the underlying probability theory did not allow the definition of two distinct state spaces, which is a requirement in order to model HMMs. Nevertheless, recent developments have yielded a more general probability theory [17], that we use, in the present work, to develop an improved formalization of DTMCs. This allows, in particular, to define both time-homogeneous and time-inhomogeneous DTMCs, and HMMs, which in turn can be used to conduct formal analysis of HMMs within the sound core of a theorem prover.

3 Formalization of Discrete-Time Markov Chains

A *probability space* is a measure space $(\Omega, \Sigma, \mathcal{P}r)$ such that $\mathcal{P}r(\Omega) = 1$ [3]. Σ is a collection of subsets of Ω (these should satisfy some closure axioms that we do not specify here) which are called *measurable sets*. In [17], a higher-order logic probability theory is developed, where given a probability space `p`, the functions `space` and `subsets` return the corresponding Ω and Σ, respectively. Mathematically, a *random variable* is a measurable function between a probability space and a *measurable space*, which refers to a pair $(S, \mathcal{A})$, where S is a set and $\mathcal{A}$

is a σ-algebra, i.e., a collection of subsets of S satisfying some particular properties [3]. In HOL, we write `random_variable X p s` to state that a function `X` is a random variable on a probability space `p` and the measurable outcome space `s`. Meanwhile, the mathematical probability $\mathcal{P}r$ is denoted as $\mathbb{P}$ in this paper. Building on these foundations, measure theoretic formalizations of probability, Lebesgue integral and information theories are presented in [17]. In this paper, we build upon these results to first formalize DTMCs and then use this to formalize HMMs.

3.1 Definition of Discrete-Time Markov Chains

A *stochastic process* [3] is a function $X : T \to \Omega$ where $T = \mathbb{N}$ (*discrete-time process*) or $T = \mathbb{R}$ (*continuous-time process*) and Ω is a measurable set called the *state space* of X. A *(finite-state) DTMC* is a discrete-time stochastic process that has a finite Ω and satisfies the *Markov property* [5]: for $0 \leq t_0 \leq \cdots \leq t_n$ and $f_0, \cdots, f_{n+1}$ in the state space, then: $\mathcal{P}r\{X_{t_{n+1}} = f_{n+1} | X_{t_n} = f_n, \ldots, X_{t_0} = f_0\} = \mathcal{P}r\{X_{t_{n+1}} = f_{n+1} | X_{t_n} = f_n\}$.

This allows to formalize the Markov property as follows:

Definition 1. (Markov Property)

```
⊢ ∀ X p s.
 mc_property X p s =
 (∀ t. random_variable (X t) p s) ∧
 ∀ f t n.
  increasing_seq t ∧ ℙ(⋂ₖ∈ [0,n−1]{x | X tₖ x = f k}) ≠ 0 ⇒
  (ℙ({x | X tₙ₊₁ x = f (n + 1)}|{x | X tₙ x = f n} ∩
                                  ⋂ₖ∈ [0,n−1]{x | X tₖ x = f k}) =
   ℙ({x | X tₙ₊₁ x = f (n + 1)}|{x | X tₙ x = f n}))
```

where `increasing_seq t` is defined as `∀ i j. i < j ⇒ t i < t j`, thus formalizing the notion of increasing sequence. The first conjunct indicates that the Markov property is based on a random process $\{X_t : \Omega \to S\}$. The quantified variable `X` represents a function of the random variables associated with time `t` which has the type `num`. This ensures the process is a *discrete time* random process. The random variables in this process are the functions built on the probability space `p` and a measurable space `s`. The conjunct $\mathbb{P}(\bigcap_{\mathtt{k}\in[0,\mathtt{n}-1]}\{\mathtt{x} \mid \mathtt{X}\ \mathtt{t_k}\ \mathtt{x} = \mathtt{f}\ \mathtt{k}\}) \neq 0$ ensures that the corresponding conditional probabilities are well-defined, where `f k` returns the $\mathtt{k}^{th}$ element of the state sequence.

A DTMC is usually expressed by specifying: an initial distribution p_0 which gives the probability of initial occurrence $\mathcal{P}r(X_0 = s) = p_0(s)$ for every state; and transition probabilities $p_{ij}(t)$ which give the probability of going from i to j for every pair of states i, j in the state space [19]. For states i, j and a time t, the *transition probability* $p_{ij}(t)$ is defined as $\mathcal{P}r\{X_{t+1} = j | X_t = i\}$, which can be easily generalized to *n-step transition probability*.

$$p_{ij}^{(n)} = \begin{cases} \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} & n = 0 \\ \mathcal{P}r\{X_{t+n} = j | X_t = i\} & n > 0 \end{cases}$$

This is formalized in HOL as follows:

Definition 2. (Transition Probability)

```
⊢ ∀ X p s t n i j.
 Trans X p s t n i j =
 if i ∈ space s ∧ j ∈ space s then
   if (n = 0) then
     if (i = j) then 1 else 0
   else ℙ({x | X (t + n) x = j}|{x | X t x = i})
 else 0
```

We will write $p_{ij}^{(n)}(t)$ for the n-step transition probability (note that the notations $p_{ij}(t)$ and $p_{ij}^{(1)}(t)$ are then equivalent).

Based on the concepts of Markov property and transition probability, the notion of a DTMC can be formalized as follows:

Definition 3. (DTMC)

```
⊢ ∀ X p s p₀ pᵢⱼ.
 dtmc X p s p₀ pᵢⱼ =
 mc_property X p s ∧ (∀ i. i ∈ space s ⇒ {i} ∈ subsets s) ∧
 (∀ i. i ∈ space s ⇒ (p₀ i = ℙ{x | X 0 x = i})) ∧
 (∀ t i j. ℙ{x | X t x = i} ≠ 0 ⇒ (pᵢⱼ t i j = Trans X p s t 1 i j))
```

The first conjunct states that a DTMC satisfies Markov property [19]. The second one ensures that every set containing just one state is measurable. The last two conjuncts indicate that `p₀` is the initial distribution and `pᵢⱼ` are the transition probabilities, respectively. It is important to note that `X` is polymorphic, i.e., it is not constrained to a particular type, which is a very useful advantage of our definition.

In practice, many applications actually make use of *time-homogenous DTMCs*, i.e., DTMCs with finite state-space and time-independent transition probabilities [2]. This is formalized as follows:

Definition 4. (Time-homogeneous DTMC)

```
⊢ ∀ X p s p₀ pᵢⱼ.
  th_dtmc X p s p₀ pᵢⱼ =
  dtmc X p s p₀ pᵢⱼ ∧ FINITE (space s) ∧
  (∀ t i j. ℙ{x | X t x = i} ≠ 0 ∧ ℙ{x | X (t + 1) x = i} ≠ 0 ⇒
     (Trans X p s (t + 1) 1 i j = Trans X p s t 1 i j))
```

where the assumptions $\mathbb{P}\{\texttt{x} \mid \texttt{X t x = i}\} \neq \texttt{0}$ and $\mathbb{P}\{\texttt{x} \mid \texttt{X (t + 1) x = i}\} \neq \texttt{0}$ ensure that the conditional probabilities involved in the last conjunct are well-defined. For time-homogenous DTMCs, $p_{ij}(t) = p_{ij}(t')$ for any t, t', thus $p_{ij}(t)$ will simply be written p_{ij} in this case.

Using these fundamental definitions, we formally verified most of the classical properties of DTMCs with finite state-space using the HOL theorem prover. Because of space limitations, we present only the formal verification of the most important properties in the following subsections and the remaining ones can be found in our proof script [14].

3.2 Joint Probability

The *joint probability* of a DTMC is the probability of a chain of states to occur. It is very useful, e.g., in analyzing multi-stage experiments. In addition, this concept is the basis for joint probability generating functions, which are frequently used in considerable system analysis problems. Mathematically, the joint probability of $n+1$ discrete random variables $X_0, \ldots, X_n$ in a DTMC can be expressed as:

$$\mathcal{P}r(X_t = L_0, \cdots, X_{t+n} = L_n) = \prod_{k=0}^{n-1} \mathcal{P}r(X_{t+k+1} = L_{k+1} | X_{t+k} = L_k) \mathcal{P}r(X_t = L_0)$$

We verified this property in HOL as the following theorem:

Theorem 1. (Joint Probability)

```
⊢ ∀ X p s t L p₀ pᵢⱼ.
   dtmc X p s p₀ pᵢⱼ ⇒
   (ℙ(⋂_{k=0}^{n} {x | X (t + k) x = EL k L}) =
   (∏_{k=0}^{n-1} ℙ({x | X (t + k + 1) x = EL (k + 1) L}|
          {x | X (t + k) x = EL k L}))ℙ{x | X t x = EL 0 L})
```

3.3 Chapman-Kolmogorov Equation

The Chapman-Kolmogorov equation [3] is a widely used property of time-homogeneous Markov chains since it facilitates the use of a matrix theory to analyze 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:

$$p_{ij}^{(m+n)} = \sum_{k \in \Omega} p_{kj}^{(n)} p_{ik}^{(m)} \tag{1}$$

Based on Equation (1) and Definition 4, the Chapman-Kolmogorov equation is formally verified as follows:

Theorem 2. (Chapman-Kolmogorov Equation)

```
⊢ ∀ X p s i j t m n p₀ pᵢⱼ.
   th_dtmc X p s p₀ pᵢⱼ ⇒
   (Trans X p s t (m + n) i j =
   ∑_{k∈space s} (Trans X p s t n k j * Trans X p s t m i k))
```

3.4 Absolute Probabilities

The unconditional probabilities associated with a Markov chain are called *absolute probabilities* which are expressed as follows:

$$p_j^{(n)} = \mathcal{P}r(X_n = j) = \sum_{k \in \Omega} \mathcal{P}r(X_0 = k) \mathcal{P}r(X_n = j | X_0 = k) \tag{2}$$

This property is formally verified as the following theorem:

Theorem 3. (Absolute Probability)

$$\vdash \forall\ \mathtt{X\ p\ s\ j\ n\ p_0\ p_{ij}}.\ \mathtt{dtmc\ X\ p\ s\ p_0\ p_{ij}} \Rightarrow (\mathbb{P}\{\mathtt{x\mid X\ n\ x = j}\} = \sum_{k\in \mathtt{space\ s}} (\mathbb{P}\{\mathtt{x\mid X\ 0\ x = k}\}\mathbb{P}(\{\mathtt{x\mid X\ n\ x = j}\}|\{\mathtt{x\mid X\ 0\ x = k}\})))$$

The formal proof script for the above mentioned properties and many other useful properties is composed of 1200 lines of HOL code, which is used in the interactive verification process. The usefulness of this development is that it can be built upon to formalize HMMs as will be shown in the next section.

4 Formalization of Hidden Markov Models

An HMM [16] is a pair of two stochastic processes $\{X_k, Y_k\}_{k\geq 0}$, where $\{X_k\}_{k\geq 0}$ is a Markov chain and $\{Y_k\}_{k\geq 0}$ denotes an observable sequence, with the *conditional independency* property [27]. The observer can visualize the output of the random process shown in $\{Y_k\}_{k\geq 0}$ but not the underlying states in $\{X_k\}_{k\geq 0}$. That is the reason why the Markov chain involved in this process is called *hidden Markov chain.*

A HMM is defined as a triple $\lambda = (A,\ B,\ \pi(0))$ with the following conditions:

1. A Markov chain $\{X_k\}_{k\geq 0}$ with state space S, the initial distribution $\pi(0) = \{\mathcal{P}r\{X_0 = i\}\}_{i\in S}$ and the transition probabilities $A = \{\mathcal{P}r\{X_{n+1} = j|X_n = i\}\}_{i\in S, j\in S}$.
2. A random process $\{Y_k\}_{k\geq 0}$ with finite state space O. $\{X_k\}_{k\geq 0}$ and $\{Y_k\}_{k\geq 0}$ are associated with the *emission probabilities* B, which is $\{\mathcal{P}r\{Y_n = O_k|X_n = j\}\}_{j\in S, O_k\in O}$.
3. $\{Y_k\}_{k\geq 0}$ is conditional independent of $\{X_k\}_{k\geq 0}$, i.e. Y_k depends only on X_k and not on any X_t, such that $t \neq k$.

In our work, we consider mainly discrete time and finite-state space HMMs, which is the most frequently used case. Now, HMM is formalized as follows:

Definition 5. (HMM)

$$\begin{aligned}
&\vdash \forall\ \mathtt{X\ Y\ p\ s_X\ s_Y\ p_0\ p_{ij}\ p_{XY}}.\\
&\mathtt{hmm\ X\ Y\ p\ s_X\ s_Y\ p_0\ p_{ij}\ p_{XY}} =\\
&\mathtt{dtmc\ X\ p\ s_X\ p_0\ p_{ij}} \wedge (\forall\ \mathtt{t.\ random_variable\ (Y\ t)\ p\ s_Y}) \wedge\\
&(\forall\ \mathtt{i.\ i} \in \mathtt{space\ s_Y} \Rightarrow \{\mathtt{i}\} \in \mathtt{subsets\ s_Y}) \wedge\\
&(\forall\ \mathtt{t\ a\ i.}\ \mathbb{P}\{\mathtt{x\mid X\ t\ x = i}\} \neq 0 \Rightarrow\\
&\quad (\mathbb{P}(\{\mathtt{x\mid Y\ t\ x = a}\}|\{\mathtt{x\mid X\ t\ x = i}\}) = \mathtt{p_{XY}\ t\ a\ i})) \wedge\\
&\forall\ \mathtt{t\ a\ i}\ t_{x_0}\ t_{y_0}\ \mathtt{sts_X\ sts_Y\ ts_X\ ts_Y}.\\
&\quad \mathbb{P}(\{\mathtt{x\mid X\ t\ x = i}\} \cap \bigcap_{\mathtt{k}\in \mathtt{ts_X}} \{\mathtt{x\mid X\ (}t_{x_0}\mathtt{\ +\ k)\ x = EL\ k\ sts_X}\} \cap\\
&\qquad \bigcap_{\mathtt{k}\in \mathtt{ts_Y}} \{\mathtt{x\mid Y\ (}t_{y_0}\mathtt{\ +\ k)\ x = EL\ k\ sts_Y}\}) \neq 0 \Rightarrow\\
&\quad (\mathbb{P}(\{\mathtt{x\mid Y\ t\ x = a}\}|\{\mathtt{x\mid X\ t\ x = i}\} \cap\\
&\qquad\qquad \bigcap_{\mathtt{k}\in \mathtt{ts_X}} \{\mathtt{x\mid X\ (}t_{x_0}\mathtt{\ +\ k)\ x = EL\ k\ sts_X}\} \cap\\
&\qquad\qquad \bigcap_{\mathtt{k}\in \mathtt{ts_Y}} \{\mathtt{x\mid Y\ (}t_{y_0}\mathtt{\ +\ k)\ x = EL\ k\ sts_Y}\}) =\\
&\quad\ \mathbb{P}(\{\mathtt{x\mid Y\ t\ x = a}\}|\{\mathtt{x\mid X\ t\ x = i}\}))
\end{aligned}$$

The variable X denotes the random variable in the underlying DTMC, Y indicates the random observations, and p_{XY} indicates the emission probabilities. The following two conditions define a random process $\{Y_t\}_{t\geq 0}$ with a discrete state space. The fourth condition assigns the emission distributions given by p_{XY}. The last condition ensures the above mentioned conditional independence.

The *time-homogenous HMMs* can also be formalized in a way similar to time-homogenous DTMCs:

Definition 6. (Time-homogeneous HMM)

```
⊢ ∀ X Y p s_X s_Y p_0 p_ij p_XY.
  thmm X Y p s_X s_Y p_0 p_ij p_XY =
  hmm X Y p s_X s_Y p_0 p_ij p_XY ∧ FINITE (space s_X) ∧ FINITE (space s_Y) ∧
  ∀ t a i j. ℙ{x | X t x = i} ≠ 0 ∧ ℙ{x | X (t + 1) x = i} ≠ 0 ⇒
     (Trans X p s_X (t + 1) 1 i j = Trans X p s_X t 1 i j) ∧
     (p_xy (t + 1) i j = p_xy t i j)
```

Next, we verify some classical properties of HMMs, which play a vital role in reducing the user interaction for the formal analysis of systems that can be represented in terms of HMMs.

4.1 Joint Probability of HMMs

The most important property of time homogeneous HMMs is the expression of the joint distribution of a sequence of states and its corresponding observation, which can be expressed using products of its emission probabilities and transition probabilities. This is frequently used to find the best state path or estimate model's parameters. Mathematically, this is expressed as the following equation:

$$\mathcal{P}r(Y_0, \cdots, Y_t, X_0, \cdots, X_t) = \mathcal{P}r(X_0)\mathcal{P}r(Y_0|X_0)\prod_{k=0}^{t-1}\mathcal{P}r(X_{k+1}|X_k)\mathcal{P}r(Y_{k+1}|X_{k+1})$$

and has been formally verified using the HOL theorem prover as follows:

Theorem 4. (Joint Probability of HMM)

```
⊢ ∀ X Y p t s_X s_Y p_0 p_ij p_XY sts_X sts_Y.
  thmm X Y p s_X s_Y p_0 p_ij p_XY ⇒
  (ℙ(⋂_{k=0}^{t}{x | X k x = EL k sts_X} ∩ ⋂_{k=0}^{t}{x | Y k x = EL k sts_Y}) =
   ℙ{x | X 0 x = EL 0 sts_X}
   ℙ({x | Y 0 x = EL 0 sts_Y}|{x | X 0 x = EL 0 sts_X})
   (∏_{k=0}^{t-1}ℙ({x | X (k + 1) x = EL (k + 1) sts_X}|{x | X k x = EL k sts_X})
         ℙ({x | Y (k + 1) x = EL (k + 1) sts_Y}|
           {x | X (k + 1) x = EL (k + 1) sts_X}))
```

4.2 Joint Probability of an Observable Path

In addition to the above property, researchers are often interested in the probability of a particular observation, independently of any underlying state path.

This can be mathematically expressed as:

$$\mathcal{P}r(Y_0, \cdots, Y_t) = \sum_{\substack{X_0, \cdots, X_t \in \\ space\ s}} \mathcal{P}r(X_0)\mathcal{P}r(Y_0|X_0) \prod_{k=0}^{t-1} \mathcal{P}r(X_{k+1}|X_k)\mathcal{P}r(Y_{k+1}|X_{k+1})$$

Using Theorem 4, we can formally verify this equation as follows.

Theorem 5. (Joint Probability of Observable Path)

```
⊢ ∀ X Y p s n sX sY p0 pij pXY stsX.
  thmm X Y p sX sY p0 pij pXY ⇒
    let ℒ = {L | EVERY (λx. x ∈ space sX) L ∧ (|L| = n + 1)} in
  (ℙ(⋂_{k=0}^{n} {x | Y k x = EL k stsY}) =
  ∑_{stsX ∈ ℒ} (ℙ{x | X 0 x = EL 0 stsX}
        ℙ({x | Y 0 x = EL 0 stsY}|{x | X 0 x = EL 0 stsX})
        (∏_{k=0}^{n-1} ℙ({x | X (k + 1) x = EL (k + 1) stsX}|
                {x | X k x = EL k stsX})
              ℙ({x | Y (k + 1) x = EL (k + 1) stsY}|
                {x | X (k + 1) x = EL (k + 1) stsX}))))
```

where `|L|` returns the length of the list `L` and `EVERY p L` is a predicate which is true iff the predicate `p` holds for every element of the list `L`.

One can note that Theorems 4 and 5 provide ways to *compute* the probabilities that are usually desired while analyzing HMMs. Consequently, if the theorems are instantiated with concrete values for their parameters, then a real number can be obtained for the corresponding probability. Thus, it seems natural to try to *automatize* such computations. Moreover, this is extremely useful since, in practice, one is always interested in applying the theorems to concrete situations. In the next subsection, we describe how to automatically acquire interesting probabilities and find the best state path, for a given HMM, using the results of Theorems 4 and 5. This makes the accuracy of theorem proving available even to users with no knowledge about logic or theorem proving, hence making our technique closer to practical usability.

4.3 Automating the HOL Computations

In order to automate the computation associated with Theorem 4, we define an SML function `hmm_joint_distribution ini_distr trans_distr e_distr sts obs` which takes as input the initial distributions, the transition probabilities, the emission distributions, a list of states and a list of observations: When calling this function, these parameters will be automatically substituted to p_0, p_{ij}, p_{XY}, sts_X and sts_Y, respectively, of Theorem 4. We then take `t` to be the length of `sts` (which should be the same as `obs`): this seems to be the most common case in practice, but could be easily relaxed if needed by adding a parameter to the function. We can then compute, using HOL4 theorems about lists, real numbers, etc., the right-hand side of the equation in Theorem 4 in an exact way (as a fraction). In the end, the function returns the corresponding instantiation of HOL4 theorem stating the equality between the joint probability and

its value. Note that the result is really a HOL4 theorem: even the operations between real numbers like multiplication or addition are obtained by deductive reasoning, thus making every single step of the computation completely reliable and *traceable*. An example of this function will be presented in the next section. The implementation of the function `hmm_joint_distribution` requires the development of an intermediate lemma and makes heavy but fine-grain use of rewriting techniques in order to have a reasonable efficiency. We do not go into implementation details due to the lack of space.

The computations associated with Theorem 5 can also be automated similarly, but we can actually go further: A problem which arises very often in practice is to find the state path which has the best probability of generating a given observation sequence. To obtain this, we need to compute the set of all possible state paths, compute the probability of each of these paths as `hmm_joint_distribution` does, and then return the path which has the best probability. Once again, in order to be the most accurate as possible, all these computations shall be done inside HOL4. This can be achieved by an SML function `best_path ini_distr trans_distr e_distr st_ty obs` where `ini_distr`, `trans_distr`, `e_distr`, and `obs` denote the same objects as for `hmm_joint_distribution` and `st_ty` denotes the type of terms representing states. This type should be a non-recursive enumerated type, i.e., defined as $\mathtt{C}_1 \mid \mathtt{C}_2 \mid \ \ldots\ \mathtt{C}_k$, where $\mathtt{C}_1, \ldots, \mathtt{C}_k$ are constructors without arguments: this ensures that the state-space is finite. The function then takes care of computing the list of all possible paths, then computes the corresponding joint probability as `hmm_joint_distribution` does, and, in the end, returns the state path which has the best such probability (note that the notion of "best probability" is also defined inside HOL4 by using the axiomatic definition of the order on real numbers). This function is currently very slow (with a 3-state path, it will take around one second to obtain the best path; for a 5-state path, it takes around one minute) due to the computation of the set of all possible state paths, but there is a lot of room for improvement, in particular by filtering paths which have trivially a null transition probability or null emission probability. This is a first step, which is not as fast as other statistical tools, on developing a tool to formally analyze HMMs.

We now show how to apply these theorems and functions in practice, by providing the formal analysis of a HMM of DNA model in the next section.

5 Application: Formal Analysis of DNA Sequence

DNA sequence analysis plays a vital role in constructing gene mapping, discovering new species and investigating disease-manifestations in genetic linkage, parental testing and criminal investigation. Statistical methods are mainly applied for analyzing DNA sequence. In particular, obtaining the probability of a state path underlying the DNA fragment is the most critical step in identifying a particular DNA sequence.

A DNA fragment is a sequence of nucleotides called A (Adenine), T (Thymine), G (Guanine) and C (Cytosine). However, nucleotide composition of DNA is in

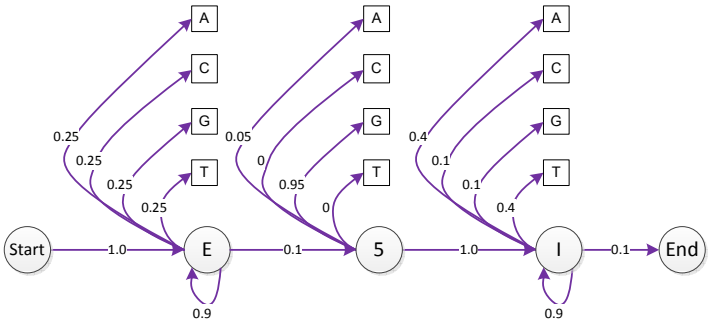

Fig. 1. 5' splice site recognition model

general not uniformly distributed (because every DNA sequence can be synthesised): some regularities can be found among the possible sequences. For instance, it might be that all four nucleotides can appear with equal probability at the beginning of the sequence, but, after a particular point, only A and G can appear, and then all four can appear again but with higher probabilities for A and T. In this application, there are thus three different "states" of the DNA, characterized by the probabilities of occurrence of each base. In this DNA model, the first state is called *exon* (E), the second one *5' splice site* (5), and the third one *intron* (I) [8]. This model is described and studied very naturally using HMMs [8]: a DTMC over the states E, 5, and I is used in order to know in which state the nucleotides are, then another random process is defined which characterizes the emission of A, G, T or C according to the state which the proteins are in. This is summarized in Fig. 1.

In order to formalize this HMM, we first define types representing the states and the bases:

Definition 7. (HOL4 Data Types)

```
⊢ dna = A | G | T | C
⊢ state = START | E | I | FIVE | END
```

Note that, in order to characterize the sequence, it is a common practice to add some fake start and end states, which have no connection with the observable sequence and thus no emission probability is required. Hence `START` and `END` are contained in the definition of `state` in Definition 7. As examples, we define the following state and DNA sequences:

Definition 8. (State Path and DNA Sequence)

```
⊢ state_seq = [START; E; E; E; E; E; E; E; E; E; E; E; E; E; E; E; E; E; E; FIVE; I; I; I; I; I; I; I; END]
⊢ dna_seq = [C; T; T; C; A; T; G; T; G; A; A; A; G; C; A; G; A; C; G; T; A; A; G; T; C; A]
```

So as to model the HMM represented in Fig. 1, we need an initial distribution, the transition probabilities, and the emission probabilities, which we define as follows:

Definition 9. (DNA Model Parameters)

```
⊢ ini_distr i = if (i = START) then 1 else 0

⊢ e_distr a i =
    case (i, a) of
      (E, _)       → 0.25
    ‖ (FIVE, A)    → 0.05
    ‖ (FIVE, G)    → 0.95
    ‖ (I, A)       → 0.4
    ‖ (I, T)       → 0.4
    ‖ (I, G)       → 0.1
    ‖ (I, C)       → 0.1
    ‖ _            → 0

⊢ trans_distr t i j =
    case (i, j) of
      (START, E)   → 1
    ‖ (E, E)       → 0.9
    ‖ (E, FIVE)    → 0.1
    ‖ (FIVE, I)    → 1
    ‖ (I, I)       → 0.9
    ‖ (I, END)     → 0.1
    ‖ _            → 0
```

Then, in order to work with random variables X and Y denoting the states and the observations, respectively, on a probability space p, it is sufficient to have the following predicate:

$$\texttt{thmm X Y p } s_X\ s_Y \texttt{ ini\_distr trans\_distr e\_distr} \wedge \texttt{ space } s_X = \texttt{univ(: state)} \wedge \texttt{ space } s_Y = \texttt{univ(: dna)}$$

where univ(:t) is the set of all possible values of type t, e.g., univ(:dna) = {A; G; T; C}.

Now, for instance, we can prove the theorem which gives the probability of obtaining the sequence dna_seq if the underlying state path is state_seq:

Theorem 6. (Joint Probability of a DNA Segment)

$$\begin{aligned}&\vdash \forall\ \texttt{X Y p } s_X\ s_Y.\\ &\texttt{thmm X Y p } s_X\ s_Y \texttt{ ini\_distr trans\_distr e\_distr} \wedge\\ &\texttt{space } s_X = \texttt{univ(: state)} \wedge \texttt{space } s_Y = \texttt{univ(: dna)} \Rightarrow\\ &\mathbb{P}(\bigcap_{k=0}^{|\texttt{state\_seq}|-1}\{\texttt{x | X k x = EL k state\_seq}\} \cap\\ &\quad \bigcap_{k=0}^{|\texttt{dna\_seq}|-1}\{\texttt{x | Y k x = EL k dna\_seq}\}) = 0.25^{18} * 0.9^{23} * 0.1^4 * 0.95 * 0.4^5\end{aligned}$$

To verify this theorem, a lemma of Theorem 4 is proved firstly:

Lemma 1.

$$\begin{aligned}&\vdash \forall\ \texttt{X Y p t } s_X\ s_Y\ p_0\ p_{ij}\ p_{XY}\ \texttt{sts}_X\ \texttt{sts}_Y.\\ &\texttt{thmm X Y p } s_X\ s_Y\ p_0\ p_{ij}\ p_{XY} \wedge (|\texttt{sts}_x|\texttt{=t + 3}) \wedge (|\texttt{sts}_y|\texttt{=t + 1}) \Rightarrow\\ &(\mathbb{P}(\bigcap_{k=0}^{t+2}\{\texttt{x | X k x = EL k sts}_X\} \cap \bigcap_{k=0}^{t}\{\texttt{x | Y k x = EL k sts}_Y\}) =\\ &\mathbb{P}\{\texttt{x | X 0 x = EL 0 sts}_X\}\\ &\mathbb{P}(\{\texttt{x | X (k + 2) x = EL (k + 2) sts}_X\} |\\ &\quad \{\texttt{x | X (k + 1) x = EL (k + 1) sts}_X\})\\ &(\prod_{k=0}^{t}\mathbb{P}(\{\texttt{x | X (k + 1) x = EL (k + 1) sts}_X\} | \{\texttt{x | X k x = EL k sts}_X\})\\ &\qquad \mathbb{P}(\{\texttt{x | Y (k + 1) x = EL k sts}_Y\} | \{\texttt{x | X k x = EL (k + 1) sts}_X\}))\end{aligned}$$

Actually, a more interesting information than the above number is to find which among all possible state paths has the highest probability to occur given a particular DNA sequence. This state path is called the *best path* in our case. In our particular context, this problem is called *5' splice site recognition*. This is verified as follows:

Theorem 7. (Best State Path)

$$
\begin{aligned}
\vdash\ & \forall\ \mathtt{X}\ \mathtt{Y}\ \mathtt{p}\ \mathtt{s_X}\ \mathtt{s_Y}. \\
& \mathtt{thmm}\ \mathtt{X}\ \mathtt{Y}\ \mathtt{p}\ \mathtt{s_X}\ \mathtt{s_Y}\ \mathtt{ini\_distr}\ \mathtt{trans\_distr}\ \mathtt{e\_distr}\ \wedge \\
& \mathtt{space}\ \mathtt{s_X} = \mathtt{univ}(:\mathtt{state})\ \wedge\ \mathtt{space}\ \mathtt{s_Y} = \mathtt{univ}(:\mathtt{dna})\ \Rightarrow \\
& \mathtt{REAL\_MAXIMIZE\_SET} \\
& \qquad [\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{E};\mathtt{FIVE};\mathtt{I};\mathtt{I};\mathtt{I};\mathtt{I};\mathtt{I};\mathtt{I};\mathtt{I}] \\
& \qquad (\lambda \mathtt{sts}.\ \mathbb{P}(\textstyle\bigcap_{\mathtt{k}=0}^{|\mathtt{sts}|-1}\{\mathtt{x} \mid \mathtt{X}\ \mathtt{k}\ \mathtt{x}\ =\ \mathtt{EL}\ \mathtt{k}\ \mathtt{state\_seq}\}\ \bigcap \\
& \qquad\qquad \textstyle\bigcap_{\mathtt{k}=0}^{|\mathtt{dna\_seq}|-1}\{\mathtt{x} \mid \mathtt{Y}\ \mathtt{k}\ \mathtt{x}\ =\ \mathtt{EL}\ \mathtt{k}\ \mathtt{dna\_seq}\}))\ \{\mathtt{sts} \mid |\mathtt{sts}| = 26\}
\end{aligned}
$$

where `REAL_MAXIMIZE_SET m f s` is a predicate which is true only if `f m` is the maximum element of $\{\mathtt{f\ x} \mid \mathtt{x} \in \mathtt{s}\}$ (this is defined as a predicate because there can be several elements of `s` having this property). Note once again that this theorem is proved in a purely formal way, i.e., even the comparisons between probabilities are proved deductively from the axiomatic definition of real numbers. Consequently, the confidence that we can have in the result is maximal.

While Theorems 6 and 7 have been proved in the classical theorem proving way, i.e., interactively, there are rare chances that a biologist has the required knowledge of higher-order logic and HOL4 so as to conduct such a study. However, we can, by using SML functions that we presented in the previous section, get the same result in a purely automated way. In order to call the functions `hmm_joint_distribution` and `best_path`, we need to define their arguments as SML values:

```
> val dna_seq =
"[C;T;T;C;A;T;G;T;G;A;A;A;G;C;A;G;A;C;G;T;A;A;G;T;C;A]";

> val state_seq =
"[START;E;E;E;E;E;E;E;E;E;E;E;E;E;E;E;E;E;E;FIVE;I;I;I;I;I;I;I;END]";
 > val ini_distr = "λ i. if (i = START) then 1 else 0";

 > val trans_distr = "λ t i j. case (i,a) of
     (START,E) → 1 || (E,E) → 0.9 || (E,FIVE) → 0.1 || (FIVE,I) → 1 ||
     (I,FIVE) → 0.9 || (I,END) → 0.1 || _ → 0"

 > val e_distr a i = "λ t a i. case (i,a) of
     (E,_) → 0.25 || (FIVE,A) → 0.05 || (FIVE,G) → 0.95 || (I,A) → 0.4 ||
     (I,T) → 0.4 || (I,G) → 0.1 || (I,C) →; 0.1 || _ → 0"
```

Note that, contrarily to the previous definitions, `dna_seq`, `state_seq`, `ini_distr`, `trans_distr` and `e_distr` are *SML values*, whereas the values with the same names presented in Definitions 8 and 9 are *HOL4 values*. Of course, in practice, these need to be defined only once (in SML if using the automated way, or in HOL4 if using the interactive way). We can then call the SML function

`hmm_joint_distribution` as follows:

```
> hmm_joint_distribution ini_distr trans_distr e_distr dna_seq state_seq;
```

which gives the following output:

```
Exact value with the corresponding assumptions (obtained by HOL4):

∀ X Y p sX sY.
  thmm X Y p sX sY
     (λ i. if i = START then 1 else 0)
     (λ t i j. case (i,j) of
        (E, _) → 0.25 ‖ (FIVE, A) → 0.05 ‖ (FIVE, G) → 0.95 ‖ (I, A) → 0.4 ‖
        (I, T) → 0.4 ‖ (I, G) → 0.1 ‖ (I, C) → 0.1 ‖ _ → 0.1
     (λ t a i. case (i,a) of
        (START, E) → 1 ‖ (E, E) → 0.9 ‖ (E, FIVE) → 0.1 ‖
        (FIVE, I) → 1 ‖ (I, I) → 0.9 ‖ (I, END) → 0.1 ‖ _ → 0 ∧
  (space sX = univ(:state)) ∧ (space sY = univ(:dna)) ⇒
  ℙ(⋂_{k=0}^{27} {x | X k x =
     EL k [START;E;E;E;E;E;E;E;E;E;E;E;E;E;E;E;E;E;E;FIVE;I;I;
           I;I;I;I;I;END]} ∩
     ⋂_{k=0}^{25} {x | Y k x =
     EL k [C;T;T;C;A;T;G;T;G;A;A;A;G;C;A;G;A;C;G;T;A;A;G;T;C;A]})
     = 168395824273397520822651 / 13421772800000000000000000000000000000000000
```

Thus, as we can see, the SML function is able to return a HOL4 theorem giving the exact value of the desired probability in a purely automated way. For convenience, the approximated value can also be computed by SML from the HOL4 exact value. Similarly, a result corresponding to Theorem 7 can be obtained automatically by using `best_path`. In [8], the probability of the best path is $e^{-41.22}$ and that of the second best path is $e^{-41.71}$. It is quite likely that the path chosen by numerical algorithm in the simulation tools is not the best one due to the numerical approximations. On the other hand, theorem proving based approach provides the best path with unrivaled accuracy.

This concludes our analysis of the 5' splice site DNA problem. It is, to the best of our knowledge, the first such *formal* analysis. In addition, we demonstrated how useful are our automation functions, since they allow to reduce the interaction with the user to a minimum, especially in reducing interactive guide when computing concrete numerical values in applications. All the proof scripts corresponding to this work are available at [14].

6 Conclusions

HMMs, which are used to model an observable stochastic process with an underlying Markov process, are mainly applied to model and analyze time series data

in various engineering and scientific systems. This paper presents a formalization of HMMs based on an enhanced definition of discrete-time Markov chain with finite state-space in a higher-order logic theorem prover. In particular, we present a formal definition of time homogeneous DTMC and formally verify some of their classical properties, such as *joint probabilities*, *Chapman-Kolmogorov Equation* and *absolute probabilities*, using the HOL4 theorem prover. Furthermore, some properties of HMMs are verified in HOL4. This work facilitates the formal analysis of HMMs and provides the foundations for formalizing more advanced concepts of Markov chain theory, like classified DTMCs and useful properties of HMMs. In addition, we automatized some of the most common tasks related to HMMs, thus demonstrating the practical usability of our approach. Due to the inherent soundness of theorem proving, it is guaranteed to provide accurate results, which is a very useful feature while analyzing HMMs associated with safety or mission-critical systems. In order to illustrate the usefulness of the proposed approach, we analyzed an HMM for 5' splice site DNA recognition using our formalization and automation. Our results exactly matched the corresponding paper-and-pencil based analysis [8], which ascertains the precise nature of the proposed approach. Note that our approach is quite general and it can be applied in DNA models, which usually consist of many states.

As the formal analysis of HMMs cannot be achieved in *PRISM*, the presented work opens the door to a new and very promising research direction, i.e., integrating HOL theorem proving in the domain of analyzing HMMs. We are currently working on extending the set of formally verified properties regarding DTMCs and extending our work to time-inhomogeneous discrete-time Markov chains, which will enable us to target a wider set of systems. We also plan to formally verify the *Forward-Backward*, *Viterbi* and *Baum-Welch* algorithms [16], which are widely applied in statistical biology analysis. By improving the efficiency of automation functions and by making their scope broader, we could also consider the development of a purely automated but formal tool to analyse HMMs.

References

1. Affeldt, R., Hagiwara, M.: Formalization of shannon's theorems in sSReflect-coq. In: Beringer, L., Felty, A. (eds.) ITP 2012. LNCS, vol. 7406, pp. 233–249. Springer, Heidelberg (2012)
2. C. Baier and J. Katoen. Principles of Model Checking. MIT Press (2008)
3. Bhattacharya, R.N., Waymire, E.C.: Stochastic Processes with Applications. John Wiley & Sons (1990)
4. ChIP-Seq Tool Set (2012), `http://havoc.genomecenter.ucdavis.edu/cgi-bin/chipseq.cgi`
5. Chung, K.L.: Markov chains with stationary transition probabilities. Springer, Heidelberg (1960)
6. Coq (2014), `http://coq.inria.fr/`
7. Daniel, N.: Electrocardiogram Signal Processing using Hidden Markov Models. Ph.D. Thesis, Czech Technical University, Czech Republic (2003)

8. Eddy, S.R.: What is a Hidden Markov Model? Nature Biotechnology 22(10), 1315–1316 (2004)
9. Frédéric, S., Delorenzi, M.: MAMOT: Hidden Markov Modeling Tool. Bioinformatics 24(11), 1399–1400 (2008)
10. 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)
11. HMMER (2013), http://hmmer.janelia.org/
12. HMMTool (2013), http://iri.columbia.edu/climate/forecast/stochastictools/
13. Hölzl, J., Heller, A.: Three Chapters of Measure Theory in Isabelle/HOL. In: van Eekelen, M., Geuvers, H., Schmaltz, J., Wiedijk, F. (eds.) ITP 2011. LNCS, vol. 6898, pp. 135–151. Springer, Heidelberg (2011)
14. L. Liu (2013), http://hvg.ece.concordia.ca/projects/prob-it/dtmc_hmm.html
15. Liu, L., Hasan, O., Tahar, S.: Formalization of finite-state discrete-time markov chains in HOL. In: Bultan, T., Hsiung, P.-A. (eds.) ATVA 2011. LNCS, vol. 6996, pp. 90–104. Springer, Heidelberg (2011)
16. MacDonald, I.L., Zucchini, W.: Hidden Markov and Other Models for Discrete-valued Time Series. Chapman & Hall, London (1997)
17. Mhamdi, T., Hasan, O., Tahar, S.: On the Formalization of the Lebesgue Integration Theory in HOL. In: Kaufmann, M., Paulson, L.C. (eds.) ITP 2010. LNCS, vol. 6172, pp. 387–402. Springer, Heidelberg (2010)
18. MRMC (2013), http://www.mrmc-tool.org/trac/
19. Norris, J.R.: Markov Chains. Cambridge University Press (1999)
20. PRISM (2013), http://www.prismmodelchecker.org
21. Robertson, A.W., Kirshner, S., Smyth, P.: Downscaling of Daily Rainfall Occurrence over Northeast Brazil using a Hidden Markov Model. Journal of Climate 17, 4407–4424 (2004)
22. Rutten, J., Kwaiatkowska, M., Norman, G., Parker, D.: Mathematical Techniques for Analyzing Concurrent and Probabilisitc Systems. CRM Monograph Series, vol. 23. American Mathematical Society (2004)
23. Sen, K., Viswanathan, M., Agha, G.: VESTA: A Statistical Model-Checker and Analyzer for Probabilistic Systems. In: IEEE International Conference on the Quantitative Evaluation of Systems, pp. 251–252 (2005)
24. YMER (2013), http://www.tempastic.org/ymer/
25. Zhang, L., Hermanns, H., Jansen, D.N.: Logic and Model Checking for Hidden Markov Models. In: Wang, F. (ed.) FORTE 2005. LNCS, vol. 3731, pp. 98–112. Springer, Heidelberg (2005)
26. Zhang, L.J.: Logic and Model Checking for Hidden Markov Models. Master Thesis, Universität des Saarlandes, Germany (2004)
27. Zoubin, G.: An Introduction to Hidden Markov Models and Bayesian Networks. International Journal of Pattern Recognition and Artificial Intelligence 15(1), 9–42 (2001)