iLTLChecker: A Probabilistic Model Checker for Multiple DTMCs *

YoungMin Kwon and Gul Agha
Department of Computer Science
University of Illinois at Urbana-Champaign
{ykwon4,agha}@cs.uiuc.edu

Abstract

iLTL is a probabilistic temporal logic that can specify properties of multiple discrete time Markov chains (DTMCs). In this paper, we describe two related tools: MarkovEstimator *a tool to estimate a Markov transition matrix, and* iLTLChecker*, a tool to model check iLTL properties of DTMCs. These tools work together to verify iLTL properties of DTMCs.*

1 iLTLChecker

iLTL [6] is a *Linear Temporal Logic* (LTL) [5] that can specify temporal changes of expected rewards of a Markov process. Unlike existing probabilistic temporal logics such as *Probabilistic Computation Tree Logic* (PCTL) [3] and *Continuous Stochastic Logic* (CSL) [1] which build a probability space on the paths of computation and reason about the probability space, iLTL specifies directly on the transitions of *Probability Mass Function* (pmf).

iLTL captures the frequency interpretation of probability in large scale systems. Consider the following simple example: suppose that there is a *Discrete Time Markov Chain* (DTMC) of two states, say $\{a, b\}$, and two transitions to other states with probability one. Suppose also that a predicate α is true in a but not in b. Because, this DTMC alternates states in every step, there are no path with consecutive sequences of a state. However, suppose that there are 100 nodes initially 50 in a state and 50 in b state. Then, there are always 50% of the nodes in a state or in b state and $\Box(\mathrm{P}[\alpha] \geq 0.5)$ is true in iLTL.

Model The model of iLTL is a set of DTMCs: $\bigcup_{i=1}^{I}\{X^{(i)}\}$, where $X^{(i)} = (S^{(i)}, \mathbf{M}^{(i)})$, $S^{(i)} = \{s_1, \ldots, s_{n_i}\}$ is a set of states, and $\mathbf{M}^{(i)} \in \mathbb{R}^{n \times n}$ is a Markov transition matrix that specifies the state transition probabilities. A state of a DTMC model X is a column vector $\mathbf{x} \in \mathbb{R}^{n \times 1}$ such that $x_i(t) = \mathrm{P}[X(t) = s_i]$. State transition is governed by the following equation:

$$\mathbf{x}(t+1) \quad = \quad \mathbf{M} \cdot \mathbf{x}(t).$$

The Markov transition matrix of a DTMC can be estimated from a periodic sampling. The estimated Markov transition matrix $\hat{\mathbf{M}}$ can be obtained by minimizing the discrepancy between the measured pmf $\bar{\mathbf{x}}(t+1)$ and the predicted pmf $\mathbf{x}(t+1|t) = \hat{\mathbf{M}} \cdot \bar{\mathbf{x}}(t)$:

$$\begin{aligned} \text{minimize}_{\hat{\mathbf{M}}} \quad & \Sigma_{t=1}^{m} |\, \bar{\mathbf{x}}(t) - \mathbf{x}(t|t-1)\,|^2 \\ \text{such that} \quad & \Sigma_{i=1}^{n} \hat{M}_{ij} = 1, \text{ for } j = 1, \ldots, n, \\ & \hat{M}_{ij} \geq 0, \text{ for } i, j = 1, \ldots, n. \end{aligned}$$

The constrained minimization problem above can be solved by Quadratic Programming (QP).

The first graph of Figure 1 shows pmf samples from 90 Mica2 [2] nodes. In this experiment each node is recording a microphone sample, performing a *Discrete Fourier Transform* (DFT) on the sample and sending the result back to a base station at random interval in order to avoid message

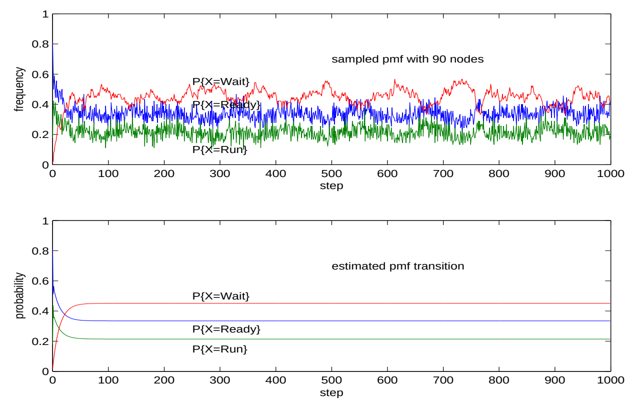

Figure 1. Pmf samples from 90 Mica2 nodes and responses from an estimated model.

*This research has been supported by the DARPA IXO NEST Award No. F33615-01-C-1907 and the ONR MURI Award No. N0014-02-0715.

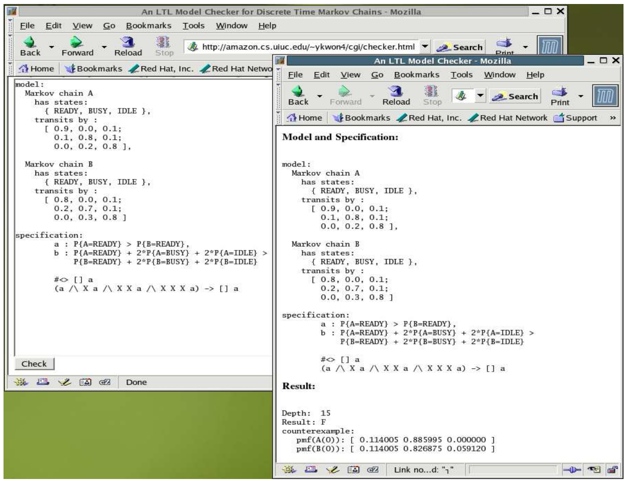

Figure 2. iLTLChecker

collision. The second graph of Figure 1 shows responses from the estimated model.

Specification iLTL can specify temporal changes of expected rewards of multiple DTMCs. A reward function ($S \to \mathbb{R}$) can be regarded as benefit or cost associated with states. Let us consider the following examples of iLTL specification. In this example we compare two DTMCs $A = (S, \mathbf{M})$, $B = (S, \mathbf{M}')$, where $S = \{ready, run, wait\}$.

- `◇□ P[A=ready] > P[B=ready] + 0.1:` in the steady state the availability, the probability that a process is in ready state, of DTMC A is at least 10% larger than that of DTMC B.

- `P[A=ready] > P[B=ready] ->`
  `□ P[A=ready]+2*P[A=run]+2*P[A=wait]`
  `> P[B=ready]+2*P[B=run]+2*P[B=wait]:`
  once the availability of A becomes larger than that of B, the expected energy consumption level of A is always larger than that of B.

2 Implementation

The Markov chain estimator is implemented in Java 1.4.2 and the goodness of fit test method uses Matlab functions, although we are migrating the tester to Java. iLTL checker is implemented mostly in C using LAPACK Fortran library for Matrix operations. Figure 2 is a snapshot of iLTLChecker [4].

Figure 2 shows an iLTLChecker description. An iLTLChecker description has two blocks: a model block which describes a list of DTMCs and a specification block which describes iLTL specifications. Note that the model block of Figure 2 defines two DTMCs `A` and `B` with the same set of states but different probability transition matrices. In the specification the inequality `a` is whether the availability of `A` is larger than that of `B`. The first commented specification describes whether in the steady state `a` is true. For which the model checker returns true. The second specification checks whether the witness of $a(t)$ being true for $t = 0, \ldots, 3$ is a sufficient condition for a a to be true always. The model checking result is false with a counter example $\mathbf{x}^{(A)}(0) = [0.114, 0.886, 0]^T$ and $\mathbf{x}^{(B)}(0) = [0.114, 0.8269, 0.0591]^T$.

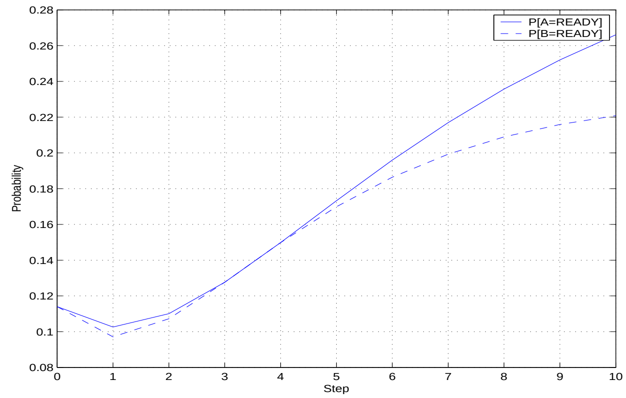

Figure 3. A counter example

Figure 3 shows probability transitions of the two DTMCs form the initial pmfs of the counter example. Note that for the first four steps the inequality `a` is true. But at the fifth step it becomes false and from then on it is true, as is proved by the first commented specification. If we add `a(4)` to the precondition of the specification then the model checker returns true. That is, `(a ∧ X a ∧ X X a ∧ X X X a ∧ X X X X a) -> [] a` is true. Thus, four consecutive witnesses of `a` is a sufficient condition that the expected availability of `A` is always larger than that of `B`.

References

[1] C. Baier, J.P. Katoen, and H. Hermanns. Approximate symbolic model checking of continuous-time markov chains. In *CONCUR, LNCS 1664*, pages 146–162, 1999.

[2] Crossbow Technology, Inc. http://www.xbow.com/.

[3] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. In *Formal Aspects of Computing*, volume 6, pages 512–535, 1994.

[4] iLTL Checker. http://osl.cs.uiuc.edu/∼ykwon.

[5] Orna Lichtenstein and Amir Pnueli. Checking that finite state concurrent programs satisfy their linear specification. 1984.

[6] YoungMin Kwon and Gul Agha. Linear Inequality LTL (iLTL): A model checker for discrete time markov chains. In *Int. Conf. on Formal engineering Methods*. LNCS 3308, Springer-Verlag, 2004.