Linear Inequality LTL (*iLTL*): A Model Checker for Discrete Time Markov Chains ⋆

YoungMin Kwon and Gul Agha

Open Systems Laboratory
Department of Computer Science
University of Illinois at Urbana Champaign
{ykwon4,agha}@cs.uiuc.edu

Abstract. We develop a way of analyzing the behavior of systems modeled using Discrete Time Markov Chains (DTMC). Specifically, we define *iLTL*, an LTL with linear inequalities on the pmf vectors as atomic propositions. *iLTL* allows us to express not only properties such as the expected number of jobs or the expected energy consumption of a protocol during a time interval, but also inequalities over such values. We present an algorithm for model checking properties of DTMCs expressed in *iLTL*. Our model checker differs from existing probabilistic ones in that the latter do not check properties of the transitions on the probability mass function (pmf) itself. Thus, *iLTLChecker* can check, given an interval estimate of current pmf, whether future pmfs will always satisfy a specification. We believe such properties often arise in distributed systems and networks and may, in particular, be useful in specifying requirements for routing or load balancing protocols. Our algorithm has been implemented in a tool called *iLTLChecker* and we illustrate the use of the tool by means of some examples.

1 Introduction

Many aspects of the behavior of distributed and embedded systems are stochastic and memoryless in nature; *Markov chains* provide a good model to analyze such behavior. In particular, queueing systems and network protocols are often analyzed using Markov chains [14, 4]. We develop a simple and efficient algorithm for model checking the behavior of such systems. An advantage of using Markov chains is that one can directly obtain the model from a system by estimating the *probability mass functions (pmf)* over time. For example, in a sensor network, we can obtain a sequence of pmfs of sensor states by taking successive snapshots.

We are interested in the temporal behavior of a system that can be expressed by *linear inequalities* over the pmfs. For example, such inequalities may be used to compose the expected queue length of a queueing system or the expected energy consumption of a network protocol. We define a *Linear Temporal Logic*, *iLTL*, which expresses properties of a Markov chain; the atomic propositions of *iLTL* include linear inequalities over pmfs. We develop a method for model checking formulae in this logic given the

⋆ This research has been supported by the DARPA IXO NEST Award F33615-01-C-1907 and the DARPA/AFOSR MURI Award F49620-02-1-0325.

Markov chain representation of a system and implement the method in a tool called *iLTLChecker*.

iLTLChecker may also be used as a run-time safety checker. For example, by analyzing a snapshot, we can get an interval estimate of the probability distribution of current states. Since such an interval estimate may be expressed as a set of linear inequalities about the pmf, we can check with certain confidence level whether some (possibly future) safety property may be violated.

A number of useful probabilistic logics and associated model checkers have been developed based on logics. In particular, model checking logics pCTL and pCTL* are probabilistic extensions of the Computation Tree Logic (CTL). They express quantitative bounds on the probabilities of correct behavior [1]. Continuous-time Stochastic Logic (CSL) is a logic for Continuous Time Markov Chains (CTMC) which has an operator to reason about the steady-state probabilities of a system [13].

PRISM is a tool that can model check DTMC, CTMC and Markov Decision Processes for specifications written in pCTL or CSL [16]. Alfaro introduces nondeterminism, or *strategy*, in pCTL and pCTL* [2]. A strategy is expressed as a conditional probability and his model checker checks the worst case and the best case probabilities. Recently, Andova *et al.* [19] extended pCTL to model check a Markov reward model. With their logic, we can specify such characteristics as the expected value. An LTL model checker for Markov chains has been developed by Vardi which checks whether the probability measure of the paths that do not satisfy a specification is zero [17]. However, all the above model checkers use logics that cannot express dynamic behaviors of Markov processes: using these logics, one cannot compare one probability directly with another probability. For example, in pCTL we can express the property "The probability that a process p will eventually terminate is larger than 0.9." Similarly, in CSL we can express "The probability that in the long run the queue will be full is 0.5" [16]. But in neither logic can we express the property "Eventually the probability of p in a busy state will be at least 10% larger than the probability of p in a terminating state." Although extended logic in Andova *et al.* [19] can express these properties through a Markov reward model, it cannot express the reward in terms of a current pmf. In particular, it cannot model check a property such as "Given an interval estimate of the current pmf, the energy consumption of the network will be within [30 mW, 50 mW]." Such properties are often of interest in practical systems.

The essential idea behind our approach is quite simple. Inequalities combined by logical connectives are expressive enough to specify a finite union or complement of polytopes in the pmf space. Using temporal operators, we can also specify the time in which the sequence pmf's of a Markov chain should remain in a given polytope. Since the pmf sequence of a Markov chain is a function of the initial pmf, model checking is about determining whether there is an initial pmf that may violate the specification. In our model checking algorithm, we find the monotonic bounding functions that bound the left-hand side of inequalities. Together with the boundedness of the initial pmf, these bounding functions make the model checking a finite procedure. The model checking is done by feasibility checking for the sets of inequalities.

2 The Discrete Time Markov Chain Model

A Markov process is a stochastic process whose past has no influence on the future if its present is specified and a Markov chain is a Markov process having a countable number of states [5]. In this paper we represent a Discrete Time Markov Chain (DTMC) X as a tuple $(S, \mathbf{M})$ where S is a finite set of states $S = \{s_1, s_2, \ldots s_n\}$ that X can take and $\mathbf{M}$ is a Markov transition matrix that governs the transitions of X's probability mass function (pmf). We also use a column vector $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^T$ to represent the pmf of X at time t such that $x_i(t) = P\{X(t) = s_i\}$. Thus,

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

Since $\mathbf{M}$ is a Markov matrix, the sum of each column is one. This guarantees that $\mathbf{M}$ has an eigenvalue $\lambda_1 = 1$ [11]. We consider only the Markov matrices which are diagonalizable, and their eigenvalues λ_i have absolute values strictly less than 1 except $\lambda_1 = 1$ ($|\lambda_i| < 1$ for $i \in [2, n]$). Since all the eigenvalues other than λ_1 have absolute values less than 1, regardless of the initial pmf $\mathbf{x}(0)$, there is a unique final pmf $\mathbf{x}(\infty)$ which is a multiple of the eigenvector $\boldsymbol{\zeta}^1 = [\zeta_1^1, \zeta_2^1, \ldots, \zeta_n^1]^T$ corresponding to the eigenvalue λ_1. That is, $\mathbf{M} \cdot \boldsymbol{\zeta}^1 = 1 \cdot \boldsymbol{\zeta}^1$ and $\mathbf{x}(\infty) = \boldsymbol{\zeta}^1 / \sum_{i=1}^{n} \zeta_i^1$ [11]. The constraint about the eigenvalues is a necessary and sufficient condition for the existence of a unique steady state pmf of X. Note that an ergodic Markov chain satisfies this eigenvalue constraint and the irreducibility and aperiodicity are the necessary and sufficient conditions for the ergodicity of X since $|S|$ is finite [15]. This condition is generally true in practical systems [14]. The diagonalizability constraint can be easily enforced even if the original Markov matrix is not diagonalizable by adding and subtracting small random numbers from its elements [11]. Observe that such very small perturbation to the probability values do not change the behavior significantly.

A Markov process is a deterministic system in the sense that once the initial pmf $\mathbf{x}(0)$ is given, the rest of the transitions $\mathbf{x}(t)$, $t \geq 1$ are uniquely determined by the Markov matrix $\mathbf{M}$: $\mathbf{x}(t) = \mathbf{M}^t \cdot \mathbf{x}(0)$.

3 Specification Logic

Because we are interested in the temporal behavior of the pmf, the specification logic should be able to express properties of the transitions of the pmf. The sort of properties we are interested in compare a probability that a DTMC will be a particular state with a constant or with another such probability possibly at a different time. We use linear inequalities about the pmf vectors as atomic propositions of our specification logic.

3.1 Syntax

We use LTL as the specification logic where the atomic propositions of the LTL are linear inequalities about the pmf $\mathbf{x}(t)$. The syntax of the specification logic is:

$$\begin{aligned} \psi &::= T \mid F \mid ineq \mid \\ &\quad \neg\psi \mid \psi \vee \phi \mid \psi \wedge \phi \mid \\ &\quad \mathtt{X}\,\psi \mid \psi\ \mathtt{U}\ \phi \mid \psi\ \mathtt{R}\ \phi \\ ineq &::= \textstyle\sum_{i=1}^{n} a_i \cdot P\{X = s_i\} < b, \end{aligned}$$

where $X = (\{s_1, \ldots, s_n\}, \mathbf{M})$, $a_i \in \mathbb{R}$ and $b \in \mathbb{R}$. As usual, $\rightarrow$, $\Box$ and $\Diamond$ are defined as follows:

$$\begin{aligned} \psi \rightarrow \phi &\equiv \neg\psi \vee \phi, \\ \Box\psi &\equiv F\,\mathtt{R}\,\psi, \\ \Diamond\psi &\equiv T\,\mathtt{U}\,\psi. \end{aligned}$$

Observe that the comparison between two probabilities at different times can be expressed by the linear inequalities of the form *ineq*. For example, given the DTMC $X = (\{s_1, \ldots, s_n\}, \mathbf{M})$, the probability that X is in state s_i at time $t + k$ is given by

$$P\{X(t+k) = s_i\} = x_i(t+k) = \mathbf{M}_i^k \cdot \mathbf{x}(t),$$

where $\mathbf{M}_i^k$ is the i^{th} row of $\mathbf{M}^k$ and $\mathbf{x}(t)$ is the pmf at time t.

Predicates about a Markov reward process [10] can also be expressed by linear inequalities. We consider only a constant reward function $\rho : S \rightarrow \mathbb{R}$ for each state. A performance metric is an accumulated reward over time. A predicate about an expected accumulated reward can be expressed as follows:

$$\begin{aligned} \textstyle\sum_{k=0}^{T} \sum_{s_i \in S} \rho(s_i) \cdot P\{X(t+k) = s_i\} &= \mathbf{r} \cdot \left(\textstyle\sum_{k=0}^{T} \mathbf{M}^k\right) \cdot \mathbf{x}(t) \\ &= \mathbf{r} \cdot \mathbf{S} \cdot \left(\textstyle\sum_{k=0}^{T} \mathbf{\Lambda}^k\right) \cdot \mathbf{S}^{-1} \cdot \mathbf{x}(t) \end{aligned}$$

where $\rho(s_i)$ is a reward function associated with a state s_i, $\mathbf{r}$ is a row vector $[\rho(s_1), \ldots, \rho(s_n)]$, $\mathbf{M} = \mathbf{S} \cdot \mathbf{\Lambda} \cdot \mathbf{S}^{-1}$ with $\mathbf{\Lambda}$ a diagonal matrix of eigenvalues of $\mathbf{M}$ and the T on the summation is an upper bound of the accumulation interval. It can be ∞ if the reward vector $\mathbf{r}$ is orthogonal to the steady state pmf vector.

3.2 Semantics

A ternary satisfaction relation $\models$ over tuples consisting of a Markov matrix, an initial pmf vector and an LTL formula is recursively defined as:

$\mathbf{M}, \mathbf{x} \models T$
$\mathbf{M}, \mathbf{x} \not\models F$
$\mathbf{M}, \mathbf{x} \models \Sigma a_i \cdot P\{X = s_i\} < b$ iff $\Sigma a_i \cdot x_i < b$
$\mathbf{M}, \mathbf{x} \models \neg\phi$ iff $\mathbf{M}, \mathbf{x} \not\models \phi$
$\mathbf{M}, \mathbf{x} \models \phi \vee \psi$ iff $\mathbf{M}, \mathbf{x} \models \phi$ *or* $\mathbf{M}, \mathbf{x} \models \psi$
$\mathbf{M}, \mathbf{x} \models \phi \wedge \psi$ iff $\mathbf{M}, \mathbf{x} \models \phi$ *and* $\mathbf{M}, \mathbf{x} \models \psi$
$\mathbf{M}, \mathbf{x} \models \mathtt{X}\,\phi$ iff $\mathbf{M}, \mathbf{M} \cdot \mathbf{x} \models \phi$
$\mathbf{M}, \mathbf{x} \models \phi\,\mathtt{U}\,\psi$ iff there exists an $i \geq 0$ such that $\mathbf{M}, \mathbf{M}^i \cdot \mathbf{x} \models \psi$ and for all $0 \leq j < i$, $\mathbf{M}, \mathbf{M}^j \cdot \mathbf{x} \models \phi$
$\mathbf{M}, \mathbf{x} \models \phi\,\mathtt{R}\,\psi$ iff for all $j \geq 0$, if for every $i < j$ $\mathbf{M}, \mathbf{M}^i \cdot \mathbf{x} \not\models \phi$ then. $\mathbf{M}, \mathbf{M}^j \cdot \mathbf{x} \models \psi$

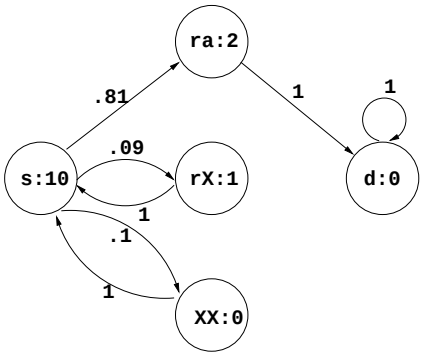
A binary satisfaction relation $\models$, over tuples of a Markov chain and an LTL formula is define as follows:

$$X \models \psi \text{ iff for all initial pmf } \mathbf{x}(0),\ \ \mathbf{M}, \mathbf{x}(0) \models \psi,$$

where $\mathbf{M}$ is the Markov matrix of X. The model checking problem to find out whether a given tuple (X, ψ) is in the binary relation $\models$ above.

3.3 Example

We give a simple example of a send/ack protocol to illustrate the notation. The protocol is represented by a state transition diagram where the transition labels are probabilities and the labels of the state include the reward (energy consumption in this example) in that state.



$$\mathbf{M} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ .81 & 0 & 0 & 0 & 0 \\ .09 & 0 & 0 & 0 & 0 \\ .1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$E = [10\ 2\ 1\ 0\ 0] \cdot \left(\sum_{t=0}^{T} \mathbf{M}^t\right) \cdot \mathbf{x}(0)$$

Fig. 1. A simple send/ack protocol (left) and its Markov matrix (**M**) and expected energy consumption (E) during the interval 0 and T (right).

The LHS of Figure 1 shows a simple send/ack network protocol. In this protocol, a sender sends a packet and a receiver, on receiving the packet, replies an ack. If the sender does not receive the ack during a certain period of time it sends the packet again. The states `s`, `ra`, `rX`, `XX` and `d` represent 'sent', 'received and acknowledged', 'received but not acknowledged', 'not received' and 'done'. The Markov matrix of this process and the expected energy consumption during the interval 0 and T with initial pmf $\mathbf{x}(0)$ are on the RHS of Figure 1. Because the reward vector [10 2 1 0 0] is orthogonal to the steady state pmf vector [0 0 0 0 1], the expected energy consumption is bounded even if T is infinite.

4 Model Checking

Let $s_X(\mathbf{x}(0))$ be a string whose alphabet is $\Sigma = 2^{AP}$ and its i^{th} alphabet is $\{ineq \in AP : ineq(\mathbf{M}^i \cdot \mathbf{x}(0))\}$ where X is a DTMC, $\mathbf{x}(0)$ is an initial pmf and AP is a set of inequalities. Let $L_X \subseteq \Sigma^*$ be a set of strings $s_X(\mathbf{x}(0))$ for all $\mathbf{x}(0)$. Then our model checker checks whether $L_X \subseteq L_\psi$ where L_ψ is a language accepted by the Büchi automata built from an LTL formula ψ. More specifically, for a given specification ψ, it checks whether any $s_X \in L_X$ is in $L_{\neg\psi}$.

Our model checking algorithm has two steps. First, we build a Büchi automaton for the negated normal form of a given LTL specification ψ using the *expand* algorithm [18]. Second, we check the feasibility of the initial pmf $\mathbf{x}(0)$ against the set of inequalities collected along finite paths obtained from the automaton. From the set of inequalities, if

there is a feasible solution, then a counterexample that does not satisfy the specification ψ is found. Otherwise, the DTMC X satisfies the given specification. Note that given the linear inequalities of an LTL formula ψ and a Markov matrix $\mathbf{M}$, we can compute an upper bound N on the number of time steps after which the atomic propositions of ψ become constant. The following section derives the upper bound.

4.1 Computation of Search Depth

Since we consider only Markov matrices $\mathbf{M}$ that are diagonalizable and whose eigenvalues have absolute values strictly less than 1 (except for $\lambda_1 = 1$), there is a stationary final pmf $\mathbf{x}(\infty)$. That is, $\mathbf{x}(\infty) = \mathbf{S} \cdot \mathbf{\Lambda}^{\infty} \cdot \mathbf{S}^{-1} \cdot \mathbf{x}(0) = \boldsymbol{\zeta}^1 / \sum_{i=1}^{n} \zeta_i^1$, where $\mathbf{S} = [\boldsymbol{\zeta}^1, \ldots, \boldsymbol{\zeta}^n]$, $\boldsymbol{\zeta}^i$ is the eigenvector corresponding to λ_i, $\mathbf{\Lambda}$ is a diagonal matrix whose diagonal is $[\lambda_1, \ldots \lambda_n]$ [11]. Furthermore, since $\mathbf{x}(t)$ is a pmf, $0 \leq x_i(t) \leq 1$ for all t. This constraint on the initial pmf and the existence of the final stationary pmf leads to bounding functions within which left-hand side of the inequalities, $\mathbf{a} \cdot \mathbf{x}(t) < b$, will remain for $t \geq 0$. So, for the set of inequalities of a given LTL specification, there is a number N after which the truth values of the inequalities become constant for every initial pmf $\mathbf{x}(0)$. This guarantees the termination of the model checking procedure.

Theorem 1. *Let $\mathbf{M} \in [0,1]^{n \times n}$ be a Markov matrix which is diagonalizable with eigenvalues $|\lambda_i| < 1$ for $i = 2 \ldots n$ and $\lambda_1 = 1$, and let $\mathbf{x}(t) \in [0,1]^{n \times 1}$ be the pmf at* t *transited by $\mathbf{M}$. Then for all inequalities $\sum_{j=1}^{n} a_{ij} \cdot x_j(t) < b_i$ of a given LTL formula ψ, if $\sum_{j=1}^{n} a_{ij} \cdot x_j(\infty) \neq b_i$ then there is an integer N such that for any integer $N' \geq N$,*

$$\Sigma_{j=1}^{n} a_{ij} \cdot x_j(N') < b_i \text{ iff } \Sigma_{j=1}^{n} a_{ij} \cdot x_j(N) < b_i.$$

Proof. For an inequality $\sum_{i=1}^{n} a_i \cdot x_i(t) < b$, let $\mathbf{a}$ be the row vector $[a_1, \ldots, a_n]$ and let b^{∞} be the inner product of $\mathbf{a}$ and the final pmf of $\mathbf{M}$ such that $b^{\infty} = \mathbf{a} \cdot \mathbf{M}^{\infty} \cdot \mathbf{x}(0)$ where $\mathbf{x}(0)$ is any initial pmf. From both sides of the inequality at step t, $\mathbf{a} \cdot \mathbf{x}(t) < b$ we subtract the equality $\mathbf{a} \cdot \mathbf{x}(\infty) = b^{\infty}$. Thus

$$\mathbf{a} \cdot (\mathbf{M}^t \cdot \mathbf{x}(0) - \mathbf{x}(\infty)) < b - b^{\infty}.$$

Since the left hand side of the inequality tends to 0 as $t \to \infty$ and $b \neq b^{\infty}$, for a given $\mathbf{x}(0)$ there is a bound n' after which the truth value of the inequality becomes a constant (true if $b^{\infty} < b$, false otherwise).

Now, we show that for all initial pmf $\mathbf{x}(0)$ there is a bound N after which the inequality become a constant. The bound N can be a value of t such that $|\mathbf{a} \cdot (\mathbf{M}^t \cdot \mathbf{x}(0) - \mathbf{x}(\infty))| < |b - b^{\infty}|$ even though this bound may not be minimal. The fact that $\mathbf{x}(0)$ is a pmf ($0 \leq x_i(0) \leq 1$) leads to a monotonically decreasing function which is larger than the left side of the previous inequality regardless the choice of $\mathbf{x}(0)$.

$$\begin{aligned}
\left|\mathbf{a} \cdot (\mathbf{M}^t \cdot \mathbf{x}(0) - \mathbf{x}(\infty))\right| &= \left|\mathbf{a} \cdot \left(\mathbf{S} \cdot \mathbf{\Lambda}^t \cdot \mathbf{S}^{-1} \cdot \mathbf{x}(0) - \mathbf{x}(\infty)\right)\right| \\
&= \left|\textstyle\sum_{i=2}^{n} c_i \cdot \lambda_i^t \cdot \sum_{j=1}^{n} S_{ij}^{-1} \cdot x_j(0)\right| \\
&\leq \textstyle\sum_{i=2}^{n} |c_i| \cdot |\lambda_i|^t \cdot \sum_{j=1}^{n} |S_{ij}^{-1}| \cdot x_j(0) \\
&\leq \textstyle\sum_{i=2}^{n} |c_i| \cdot |\lambda_i|^t \cdot \sum_{j=1}^{n} |S_{ij}^{-1}| \cdot 1,
\end{aligned}$$

where $\mathbf{M} = \mathbf{S} \cdot \mathbf{\Lambda} \cdot \mathbf{S}^{-1}$, $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues λ_i and $\mathbf{S}$ is the eigenvector matrix $[\zeta^1, \zeta^2, \ldots, \zeta^n]$ and $\mathbf{c} = [c_1, c_2, \ldots, c_n] = \mathbf{a} \cdot \mathbf{S}$. Since $\sum_{i=2}^{n} |c_i| \cdot |\lambda_i|^t \cdot \sum_{j=1}^{n} |S_{ij}^{-1}|$ is a monotonically decreasing function of t which is larger than $|\mathbf{a} \cdot (\mathbf{M}^t \cdot \mathbf{x}(0) - \mathbf{x}(\infty))|$, the integer N is the maximum N_i of all inequalities in ψ such that N_i is the minimum t that satisfies $\sum_{i=2}^{n} |c_i| \cdot |\lambda_i|^t \cdot \sum_{j=1}^{n} |S_{ij}^{-1}| < |b - b^{\infty}|$.

■

Given a DTMC $X = (S, \mathbf{M})$, an initial pmf $\mathbf{x}(0)$ and an LTL formula, because we can compute the bound after which the truth value of the inequalities in the LTL formula become constants, after a finite expansion of the LTL formula, we can evaluate it. Recall that the 'until' and 'release' operators may be rewritten as

$$\phi \,\mathtt{U}\, \psi \equiv \psi \wedge (\phi \vee \mathtt{X}\,(\phi \,\mathtt{U}\, \psi))$$
$$\phi \,\mathtt{R}\, \psi \equiv (\phi \wedge \psi) \vee (\phi \wedge \mathtt{X}\,(\phi \,\mathtt{R}\, \psi)).$$

However, because *iLTL* uses a Büchi automaton for a given LTL formula, we do not explicitly expand the formula as above.

4.2 Model checking as feasibility checking

In order to check the model X against a specification ψ, we first build a Labeled Generalized Büchi Automata (LGBA) by the expansion algorithm in [18]. The LGBA is a 6-tuple $\mathcal{A} = (S, R, S_0, F, D, L)$ where S is a set of states, $R \subseteq S \times S$ is a transition relation, $S_0 \subset S$ is a set of initial states, $F \in 2^{2^S}$ is a set of sets of final states, D is a set of inequalities and $L : S \to 2^D$ is a labeling function that returns a set of inequalities that a state should satisfy. Note that $\neg(\mathbf{a} \cdot \mathbf{x} < b)$ is converted to the inequality $-\mathbf{a} \cdot \mathbf{x} < -b$ while finding the negation normal form of ψ.

We call a (possibly infinite) sequence of states $q = q_0 q_1 \cdots q_\omega$ a *sequence of* $\mathcal{A}$ if $(q_i, q_{i+1}) \in R$ for $i \geq 0$. For each final set $F_i \in F$, if some elements of the set appear infinitely often in q then we call q an *accepting sequence of* $\mathcal{A}$. A sequence q of $\mathcal{A}$ is an *execution of* $\mathcal{A}$ if $q_0 \in S_0$. We say that a *sequence* q *of* $\mathcal{A}$ *accepts a pmf* $\mathbf{x}(t)$ if $\mathbf{x}(t + \tau)$ satisfies all inequalities of $L(q_\tau)$ for $\tau \geq 0$. An LGBA $\mathcal{A}$ *accepts a pmf* $\mathbf{x}(0)$ if there is an accepting execution of $\mathcal{A}$ that accepts $x(0)$.

Let $F' = \{q_i$: an accepting sequence of $\mathcal{A}$ $(q_1, \ldots, q_i, \ldots, q_k)^*$ accepts the pmf $\mathbf{x}(\infty)\}$ where q^* means an infinite concatenation of q. And, let $F^+ = \{q_i$: a sequence of $\mathcal{A}$ $q_1, \ldots, q_i, \ldots, q_k$ accepts the pmf $\mathbf{x}(\infty)$, and $q_k \in F'\}$. Note that any state in F^+ is reachable to an accepting sequence of $\mathcal{A}$. So, we check only those search paths of length N that ends with a state in F^+.

Theorem 2. *Let* $\mathcal{A}$ *be an LGBA* (S, R, S_0, F, D, L) *for a specification* ψ*, let* M *be the Markov matrix of* X *and let a set of inequalities* $I(q)$ *for an execution* q *be* $I(q) = \{\mathbf{a} \cdot \mathbf{M}^i \cdot \mathbf{x} < b : \mathbf{a} \cdot \mathbf{x} < b \in L(q_i), i \in [0, N]\}$ *where the* N *is the bound computed in theorem 1.*

A pmf $\mathbf{x}(0)$ *is accepted by* $\mathcal{A}$ *iff there is an execution* $q = q_0 q_1 \cdots q_\omega$ *of* $\mathcal{A}$ *with* $q_N \in F^+$ *and* $\mathbf{x}(0)$ *is a feasible point of* $I(q)$*.*

Proof. $\rightarrow$: Since $\mathcal{A}$ accepts $\mathbf{x}(0)$ there is an accepting execution q of $\mathcal{A}$ that accepts $\mathbf{x}(0)$. That is, $\mathbf{x}(t)$ satisfies all inequalities in $L(q)$ for $t \geq 0$. Since $\mathbf{a}\cdot\mathbf{x}(t) < b \equiv \mathbf{a}\cdot\mathbf{M}^t\cdot\mathbf{x}(0) < b$, $\mathbf{x}(0)$ is a feasible point of $I(q)$

Let q^N be the suffix of q after first N elements. Since q^N is an accepting sequence of $\mathcal{A}$ with infinite length over a finite set of states, there is a loop in the sequence that includes at least one elements of F_i for all $F_i \in F$. Since some states of the loop are in F', and since q_N is reachable to that states $q_N \in F^+$.

$\leftarrow$: Since $\mathbf{x}(0)$ satisfies all inequalities of $I(q)$, q accepts $\mathbf{x}(0)$ and since $q_N \in F^+$, q is an accepting execution of $\mathcal{A}$. Hence, $\mathcal{A}$ accepts $\mathbf{x}(0)$.

■

Since $\mathcal{A}$ accepts exactly those executions that satisfy ψ [18] if there is a feasible solution for $I(q)$ of an execution q of $\mathcal{A}$ with $q_n \in F^+$ then $\mathbf{M}, \mathbf{x}(0) \models \psi$. Hence, $\mathbf{x}(0)$ is a counterexample of the specification $\neg\psi$ the original specification. The feasibility can be checked by solving a linear programming problem with artificial variables [7]. Since we check the feasibility by linear programming we have a problem dealing with inequality. For example $\neg(a > b)$ will be translated as $-a < -b$ instead of $-a \leq -b$. However, because we are dealing with continuous values, precise equality is not meaningful. Instead, we can check properties such as, $|a - b| < \epsilon$ or $|a - b| > \epsilon$.

Figure 2 shows a Büchi automaton for the negation of `[]a`. Let us assume that $\neg a$ is true for the final pmf. Then all three states are in $F^+ = \{\{T\}, \{\neg a\}, \{\}\}$ and after N transitions the model checker will decide `[]a` false. If $\neg a$ is false for the final pmf then the only state in F^+ is the final state with an empty set of atomic propositions ($F^+ = \{\{\}\}$). So, if F^+ is reachable by N transitions then `[]a` is false, otherwise it is true.

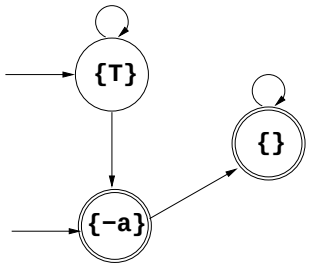


Fig. 2. A Büchi automaton for an LTL formula ¬`[]a`

One difficulty in the DTMC model checking is the size of the initial pmf space. Since $\mathbf{x}(0) \in [0, 1]^n$ with $\sum_{i=1}^{n} x_i(0) = 1$, we cannot build an automaton for the intersection of X and an LTL formula ψ as is done in [8]. Instead, we check the feasibility of the initial distribution against the transformed inequalities (i.e., consider the inequalities as linear constraints).

Let $\mathcal{A}$ be the Büchi automaton for an LTL formula $\neg\psi$ and let N be the bound for a DTMC X and the inequalities in ψ. Then the LTL DTMC model checking can be done

by a depth-first-search for a sequence $q = q_0 q_1 \cdots q_N$ of $\mathcal{A}$ with $q_0 \in S_0$ and $q_N \in F^+$ and has a feasible solution $\mathbf{x}(0)$ for the set of inequalities $I(q)$. Thus, $X \models \psi$ iff there is a such sequence q. If $X \not\models \psi$ then $\mathbf{x}(0)$ is a counterexample.

Since we depth-first search the sequences, if infeasibility is found early in the search tree by the set $\cup_{i=0}^{k}\{\mathbf{a} \cdot \mathbf{M}^{k-1} \cdot \mathbf{x} < b : \mathbf{a} \cdot \mathbf{x} < b \in L(q_i)\}$ with a small k then we can skip checking large sub-branches. One small but very efficient optimization is checking the newly added inequalities with the feasible vector from the previous state before doing the linear programming. In many cases we can skip the linear programming.

As an example of the model checking let us consider the the Büchi automaton of Figure 2. Assume that N is 3 and $\neg a$ is false by the final pmf $\mathbf{x}(\infty)$. Then F^+ is the states labeled by $\{\{\neg a\}, \{\}\}$. All paths of length 3 that end with the closure are:

$$(\neg a, \phi, \phi), (T, \neg a, \phi), (T, T, \neg a).$$

So the set of inequalities to check are (let $\neg a$ is $\mathbf{c} \cdot \mathbf{x} < b$):

$$\{\mathbf{c} \cdot \mathbf{x} < b\}, \{\mathbf{c} \cdot \mathbf{M} \cdot \mathbf{x} < b\}, \{\mathbf{c} \cdot \mathbf{M}^2 \cdot \mathbf{x} < b\}$$

If there is a feasible solution vector $\mathbf{x}(0)$ that satisfies any of the three sets of inequalities, then $\mathbf{x}(0)$ is a counterexample that satisfies the negated specification $\neg$`[]a`. That is, the initial pmf $\mathbf{x}(0)$ violates the specification `[]a`.

5 *iLTLChecker*: a Markov Chain Model Checker

The syntax of *iLTLChecker* is straight forward; it has two main blocks: corresponding to the model and to the specification. We input the states of a DTMC and its Markov transition matrix in the model block. In the specification block, we write optional inequality definitions followed by an *iLTL* formula. Figure 3 shows a snapshot of the execution of *iLTLChecker*. It is available from the web site *http://osl.cs.uiuc.edu/~ykwon4*.

5.1 Example: an M/M/1 queueing system

Consider a queuing system of Figure 4. It is an M/M/1 queueing system [12] with capacity 5, arrival rate $\lambda = 60/sec$ and service rate $\mu = 70/sec$. To simplify the discretization process, we over-sample the system (1000 sample/sec) and assume that on each sample a single new job arrives or a single job is finished with the probability of 0.06 (λ) and 0.07 (μ) each. If a new job arrives when the queue is full, it will be dropped.

Let $q = (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \mathbf{M})$ be the DTMC for the queueing system. Each state of q represents the number of jobs in the queue. The Markov transition matrix $\mathbf{M}$ can be obtained from the following probability equations:

$$\begin{aligned}
P\{q(t+1) = q_i\} &= \lambda \cdot P\{q(t) = q_{i-1}\} + \mu \cdot P\{q(t) = q_{i+1}\} \\
&\quad +(1 - \lambda - \mu) \cdot P\{q(t) = q_i\} \text{ where } i \in \{1, \ldots, 4\} \\
P\{q(t+1) = q_0\} &= \mu \cdot P\{q(t) = q_1\} + (1 - \mu) \cdot P\{q(t) = q_0\} \\
P\{q(t+1) = q_5\} &= \lambda \cdot P\{q(t) = q_4\} + (1 - \lambda) \cdot P\{q(t) = q_5\}.
\end{aligned}$$

The steady state pmf vector of q is [0.24 0.20 0.17 0.15 0.13 0.11] and the expected number of jobs in the queue at the steady state is 2.0569.

Our model checker description for the system is as follows:

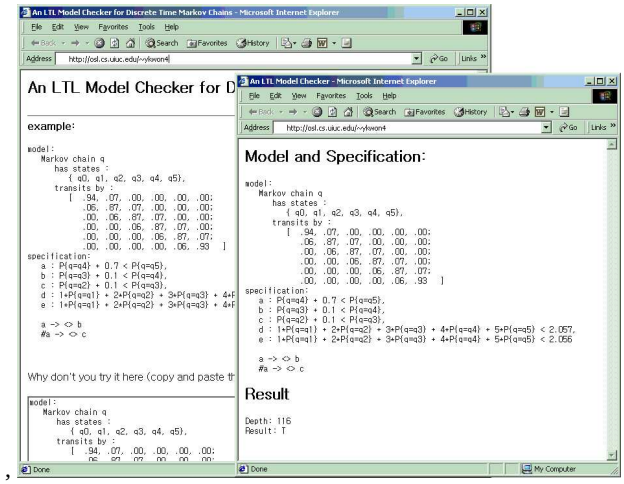


Fig. 3. *iLTLChecker*: a snapshot of an execution of *iLTLChecker*.

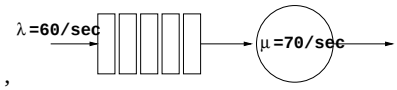


Fig. 4. A queueing system example: M/M/1 queue with capacity 5, $\lambda = 60/sec$ and $\mu = 70/sec$.

```
model:
   Markov chain q
      has states :
         { q0, q1, q2, q3, q4, q5},
      transits by :
         [  .94, .07, .00, .00, .00, .00;
            .06, .87, .07, .00, .00, .00;
            .00, .06, .87, .07, .00, .00;
            .00, .00, .06, .87, .07, .00;
            .00, .00, .00, .06, .87, .07;
            .00, .00, .00, .00, .06, .93  ]
specification:
   a : P{q=q4} + 0.7 < P{q=q5},
   b : P{q=q3} + 0.1 < P{q=q4},
   c : P{q=q2} + 0.1 < P{q=q3},
   d : 1*P{q=q1} + 2*P{q=q2} + 3*P{q=q3}
                 + 4*P{q=q4} + 5*P{q=q5} < 2.057,
   e : 1*P{q=q1} + 2*P{q=q2} + 3*P{q=q3}
                 + 4*P{q=q4} + 5*P{q=q5} < 2.056
   a -> <> b
```

The `model` block of the checker defines the DTMC for the diagram: the states and the Markov transition matrix. The `'has states'` block specifies an ordered set of the states of the Markov process q. The `'transits by'` block defines the Markov matrix that governs the transitions of the pmf of q. The rows and columns are ordered according to the order of the state set defined previously. That is, the $(i, j)^{th}$ element of the matrix m_{ij} is the probability $P[q(t+1) = s_i | q(t) = s_j]$.

The first part of the `specification` block contains optional inequality definitions. The time index of the random variable is omitted from the inequalities for convenience. The second part of the `specification` block is an LTL formula against which the model will be checked. Its atomic propositions are either `T`, `F` or the inequalities defined previously.

The LTL formula `a-><>b` checks whether once $P\{q = q5\}$ is at least 0.7 larger than $P\{q = q4\}$ then eventually $P\{q = q4\}$ will be at least 0.1 larger than $P\{q = q3\}$. The following is the result of the model checking.

```
Depth:  138
Result: T
```

The depth value 138 is the search depth. Since the depth appears before the actual model checking algorithm begins, a user can abort the model checking and adjust the specification if the bound is too large. The result of the model checking is *true* meaning the DTMC q satisfies the formula `a-><>b`.

Again we checked the formula `a-><>c`. The formula checks a similar condition with the previous example except $q4$ is replaced by $q3$ and $q3$ is replaced by $q2$. The model checking result is:

```
Depth:  95
Result: F
counterexample: [ 0.30 0.00 0.00 0.00 0.00 0.70 ]
```

The checker disproves the property by providing a counterexample above. Figure 5 shows the probability transitions. It shows that once the inequality a is satisfied, $P\{q = q3\}$ is never larger than $P\{q = q2\}$ by 0.1.

Finally, the specification `<>[](d /\ ~e)` checks whether the expected number of entities in the queue at the steady state is in between 2.057 and 2.056. Since the computed value of it is 2.0569, the checker should return *true*. However, the required depth to check the specification is 630 and it cannot be checked easily because the number of paths to be searched is exponential in N. In this example, the early checking strategy does not prune many branches. As a result the checking did not finish in 10 minutes with a 1.2GHz Pentium III machine. However, since the eigenvalues of the Markov matrix are { 1.0,0.98,0.93,0.87,0.80,0.76 } the Markov process has a unique steady state pmf. So, for the purpose of steady state expected queue length, we can check `<>[]~(d /\ ~e)` instead of `<>[](d /\ ~e)`. The result is as follows:

```
Depth:  630
Result: F
counterexample: [ 0.13 0.21 0.12 0.53 0.00 0.00 ]
```

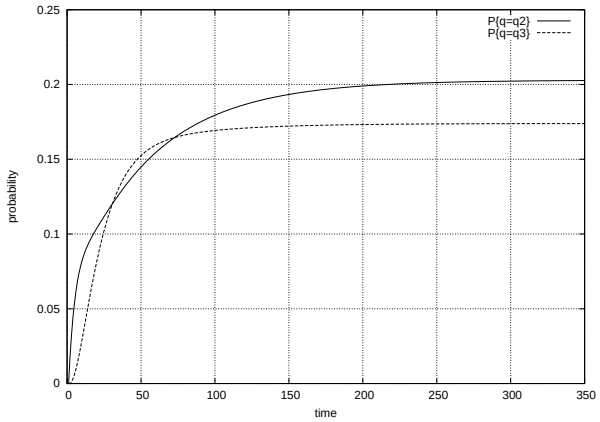


Fig. 5. The transition of the probabilities $P\{q = q2\}$ and $P\{q = q3\}$ beginning with the counter example discovered by the model checker.

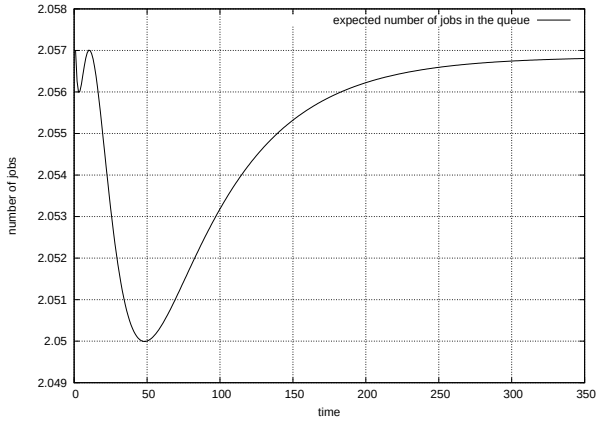


Fig. 6. The transition of the expected number of jobs in the queue beginning with the counter example discovered by the model checker.

Since there is a steady state expected value and the expected value is not outside of the interval (2.057,2.056), it must be in the interval. Figure 6 shows how the expected value changes from the initial pmf of the counterexample. The value always eventually remains in the interval. So the model does not satisfy the specification, and we can infer that the steady state expected value will remain in the interval.

6 Conclusions

We have developed an LTL model checking algorithm for discrete time Markov chain models. To specify the temporal behavior of Markov chains, we add linear inequalities about pmf vectors to the atomic propositions of LTL. Such inequalities combined with logical connectives are expressive enough to specify temporal behavior of a Markov reward model. Since the future rewards are expressed in terms of initial pmfs, we can use an interval estimate of the current pmf in the specification. For example, an expected energy consumption of a network protocol given current pmf about states can be specified and checked. Given a Markov transition matrix, we can find a monotonic bounding function within which the pmf must remain. With the bounding function, we can find a time bound after which the inequalities become constant. Hence the model checking is guaranteed to terminate.

In a practical system like wireless sensor network (WSN), modeling states of the network as they are will encounter a state-explosion problem. For example, a WSN of 100 nodes modeled as queueing systems of capacity 10 has 10^{100} states which cannot be represented directly. For this matter, state lumping techniques and space efficient structures like MDD (multi-valued decision diagram) are introduced [3]. There also is a probabilistic simulation based approach for this problem [6]. As a future work for verifying large scale system, we are looking for these directions.

Although model checking is expensive, some preliminary experiments with a tool based on our model checker suggests that it may be useful for a number of practical examples. Recall that the model checker does a depth-search for a counterexample over the executions of the Buchi automata of the specification. A bound for the length of executions is found; this bound is a function of the Markov matrix M and the inequalities used in the specification. The time complexity of our algorithm is O((maximum out degree of the nodes in Büchi automaton)^N). However, in practice the model checker skips many branches of the search tree by checking the feasibility early in the path. We are planning to introduce the partial order reduction technique used in the SPIN model checker which can reduce the search space substantially [9].

We describe a queueing system and model check several properties for this system as a way to illustrate the usefulness of our model checking tool. We are currently carrying out further work on the tool and applying it to other applications.

References

1. Adnan Aziz, Vigyan Singhal and Felice Balarin. It usually works: The temporal logic of stochastic systems. In *Proc. CAV'95, LNCS 939*, pages 155–165, 1995.

2. Andrea Bianco, Luca de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Proceedings of Conferenco on Foundations of Software Technology and Theoretical Computer Science, Lecturenotes in Computer Science*, volume 1026, pages 499–513, 1995.
3. Andrew S. Miner and Gianfranco Ciardo. Efficient reachability set generation and storage using decision diagrams. In *Int. Conf. on Applications and Theory of Petri Nets*, pages 6–25. LNCS 1639, Springer-Verlag, 1999.
4. Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall PTR, 4th edition, 2003.
5. Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 3rd edition, 1991.
6. Christel Baier, Joost-Pieter Katoen, Holger Hermanns and Boudewijn Haverkort. Simulation for continuous-time markov chains. In *Proc. CONCUR*, pages 338–354. Springer LNCS 2421, 2002.
7. David G. Luenberger. *Linear and Nonlinear Programming*. Addison Wesley, 2nd edition, 1989.
8. Edmund Clarke, Orna Grumberg, Doron Peled. *Model Checking*. MIT-Press, 2000.
9. Gerard J. Holzmann. The model checker spin. In *IEEE Transactions on Software Engineering*, volume 23, pages 279–295, May 1997.
10. Gianfranco Ciardo, Raymond A. Marie, Bruno Sericola and Kishor S. Trivedi. Performability analysis using semi-markov reward process. In *IEEE Transactions on Computers*, volume 39, pages 1251–1264, October 1990.
11. Gilbert Strang. *Linear Algebra and Its Applications*. Harcourt Brace Jovanovich, 3rd edition, 1988.
12. Henry Start, John W. Woods. *Probability and Random Processes with Applications to Signal Processing*. Prentice-Hall, 3rd edition, 2002.
13. Holger Hermanns, Joost-Pieter Katoen, Joachim Meyer-Kayser and Markus Siegle. A markov chain model checker. In *S. Graf and M. Schwartzbach, editors, TACAS'2000*, pages 347–362, 2000.
14. J.R. Norris. *Markov Chains*. Cambridge University Press, 1997.
15. Julian Keilson. *Markov Chain Models-Rarity and Exponentiality*. Springer-Verlag, 1979.
16. Marta Kwiatkowska, Gethin Norman and David Parker. Prism: Probabilistic symbolic model checker. In *Proc. TOOLS*, volume 2324, pages 200–204. LNCS, Springer-Verlag, April 2002.
17. Moshe Y. Vardi. Probabilistic linear-time model checking: an overview of the automata-theoretic approach. In *Proc. 5th Int. AMAST Workshop Formal Methods for Real-Time and Probabilistic Systems*, volume 1601, May 1999.
18. R. Gerth, D. Peled, M.Y. Vardi and P. Wolper. Simple on-the-fly automatic verification of linear temporal logic. In *IFIP/WG*, volume 6.1, pages 3–18, 1995.
19. Suzana Andova, Holger Hermanns and Joost-Pieter Katoen. Discrete-time rewards model-checked. In *Formal Modeling and Analysis of Timed Systems 2003*, pages 88–104. LNCS, Springer-Verlag, 2003.