Model Checking by Stochastic Comparison

Nihal Pekergin

LACL, University of Paris-Est (P12)

French Research Project ANR-06-SETIN-002 CheckBound

www.lamsade.dauphine.fr/checkbound

Univ.Paris1, Univ. Paris-Dauphine, Univ. Versailles, INT-Evry, INRIA Rhône Alpes (Mescal)

Outline

- Brief introduction for PCTL Model Checking
- Bounding Approach for Model Checking
- Stochastic Comparison
- Model Checking by Class $\mathcal{C}$ Markov chains
- Perspectives

Temporal logic for specification

Markovian models have been widely used as performance, reliability, and dependability models

- High-level specification methods to construct large and complex models
  Stochastic Petri Nets, Stochastic Process Algebra, etc.
- Computation of transient and the steady-state distributions
- Evaluation of underlying performability measures

Temporal logic to specify complex measures of interest in a compact and unambiguous way

Model checking

- Model Checking : an automated manner to check if the underlying formulas are satisfied or not
- Deterministic Model Checking has been successfully applied to validate qualitative properties
- Extension to stochastic models for the verification of probabilistic quantitative properties

Checking performance and reliability guarantees

Different Formalisms

Different stochastic models

- DTMC, Probabilistic Computation Tree Logic (PCTL)
- CTMC, Continuous Stochastic Logic (CSL)
- Markov Decision Processes
- Markov Reward Models

Specification of

- standard transient and steady-state state measures
- probabilistic measures over paths

Probabilistic Model Checking Formulas

$\mathcal{M}$: labelled Markov chain is a 3-tuple $(S, \mathbf{P}, L)$

- S : a finite set of *states*
- $\mathbf{P}$: the transition probability matrix
- $L : S \rightarrow 2^{AP}$ the *labelling* function

  $L(s)$: the set of atomic propositions $a \in AP$ that are valid in s

  AP: the finite set of atomic propositions

Syntax PCTL

Let α, β be integers, $p \in [0,1]$ be a probability, a be an atomic proposition, and $\triangleleft$ be a comparison operator $\in \{\leq, \geq\}$. ϕ : state formula

$$\phi ::= true \mid a \mid \phi \wedge \phi \mid \neg \phi \mid \mathcal{P}_{\triangleleft p}(\mathcal{X} \phi) \mid \mathcal{P}_{\triangleleft p}(\phi_1 \, \mathcal{U}^{[\alpha,\beta]} \phi_2) \mid \mathcal{S}_{\triangleleft p}(\phi)$$

- $\mathcal{X}\phi$: Path formula **Next**
- $\phi_1 \, \mathcal{U}^{[\alpha,\beta]} \phi_2)$: Path formula **Until**
- $\mathcal{S}_{\triangleleft p}(\phi)$: Steady-state formula

Semantic

- a path $\sigma \equiv s_0 s_1 \ldots$ is an infinite sequence of states of the Markov chain
- φ: path formula; $\sigma \models \varphi$: path σ satisfies φ
- $\sigma \models \mathcal{X}\phi$ iff $s_1 \models \phi$

  (next state satisfies ϕ)
- $\sigma \models \phi_1 \mathcal{U}^{[\alpha,\beta]} \phi_2$ iff $\exists i \; \alpha \leq i \leq \beta \wedge s_i \models \phi_2 \wedge \forall j < i \; s_j \models \phi_1$

  (passing through ϕ_1 states to reach a ϕ_2 state in $[\alpha, \beta]$)

$\mathcal{P}_{\triangleleft p}(\mathcal{X}\phi)$ and $\mathcal{P}_{\triangleleft p}(\phi_1 \, \mathcal{U}^{[\alpha,\beta]} \phi_2)$: state formulas

The true or false value will be assigned to the initial state

Semantic

- $s \models \mathcal{P}_{\lhd p}(\mathcal{X}\phi)$ iff $Prob^{\mathcal{M}}(s, \mathcal{X}\phi) \lhd p$

  the probability to reach a ϕ state from state s in one step

  $\sum_{s' \models \phi} \mathbf{P}[s, s'] \lhd p$

- $s \models \mathcal{P}_{\lhd p}(\phi_1 \, \mathcal{U}^{[\alpha,\beta]} \phi_2)$ iff $Prob^{\mathcal{M}}(s, \phi_1 \mathcal{U}^{[\alpha,\beta]} \phi_2) \lhd p$

  sum of probability measures of paths beginning from s passing through only ϕ_1 states to reach a ϕ_2 states in $[\alpha, \beta]$ steps

  transformation of $\mathcal{M}$ and transient analysis

- $s \models (\mathcal{M} \models) \mathcal{S}_{\lhd p}(\phi)$ iff $\sum_{s' \models \phi} \mathbf{\Pi}_s^{\mathcal{M}}(s')$

  steady-state analysis

Performability Guarantees

Consider a reliability model :

- **DOWN** states : not operational
- **UP** states : operational
- **SECURE** : secure for security issues

Standard Measures

- steady-state availability : $\mathcal{S}_{\lhd p}(UP)$
- instantaneous availability at step n : $\mathcal{P}_{\lhd p}(UP\,\mathcal{U}^{[n,n]}UP)$
- interval failure : $\mathcal{P}_{\lhd p}(UP\,\mathcal{U}^{[0,n]}DOWN)$
- secure execution : $\mathcal{P}_{\lhd p}(SECURE\,\mathcal{U}^{[0,\infty]}SECURE)$

Bounding Approach for Model Checking

Model checking : specification of a constraint (bound)

the exact values are not necessary, we must check if the constraints are satisfied or not

Bounding methods are useful for Model Checking

S_Σ : the set of states for which the probabilities must be summed to check the underlying formula Fr. Let denote this sum by $P_{Fr}(S_\Sigma)$

- Check to see if $P_{Fr}(S_\Sigma) \leq p$
- Compute lower and upper bounds $\mathcal{B}_{inf}$ and $\mathcal{B}_{sup}$ on $P_{Fr}(S_\Sigma)$
  - $B_{sup} \leq p$, then Fr is true
  - $B_{inf} > p$, then Fr is false
  - otherwise, it is not possible to conclude

Until Operator

$Fr = \mathcal{P}_{\leq p}(\phi_1 \mathcal{U}^{[0,k]} \phi_2)$

- **success** states : labelled with ϕ_2
- **failure** states : not labelled with ϕ_1 nor ϕ_2
- **inconclusive** states : labelled with ϕ_1 but not with ϕ_2

Transformation of $\mathcal{M} \rightarrow \mathcal{M}^T$

- Make success and failure states absorbing
- Compute transient distribution at time t starting from state s
- If the transient distribution to be in success state at step k is less or equal to p, state s satisfies the formula $s \models \mathcal{P}_{\leq p}(\phi_1 \mathcal{U}^{[0,k]} \phi_2)$

$S_\Sigma = S_{\phi_2}$

$P_{Fr}(S_\Sigma) = \sum_{S_\Sigma} \mathbf{\Pi}_s^{\mathcal{M}^T}(S_\Sigma, t)$

Bounding of Markov chains

Bounding techniques have been largely applied to overcome the state space explosion of Markov chains

Different methods according to the concepts that they are based on and to the type of obtained bounds

Stochastic Comparison for Model Checking

- Bounds on transient and the steady-state distributions
- Inequalities on the sum of probabilities

Stochastic Comparison

Computing bounding distributions by considering bounding chains having simpler numerical computations

- by reducing the state space size
- by imposing specific structures letting to apply specific numerical methods

Stochastic Order

Let X and Y be random variables taking values in a totally ordered state space E :

$$X \preceq_{st} Y \iff Ef(X) \leq Ef(Y), \quad \forall\, f \text{ increasing function}$$

State space $E = \{1, \ldots, n\}$

Let $\mathbf{\Pi}^X = (p_1, p_2, \cdots, p_n)$ and $\mathbf{\Pi}^Y = (q_1, q_2, \cdots, q_n)$ be probability distributions of X and Y

$$\mathbf{\Pi}^X \preceq_{st} \mathbf{\Pi}^Y \iff \sum_{k=i}^{n} p_k \leq \sum_{k=i}^{n} q_k, \quad \forall i \in \{1, \ldots, n\}$$

Comparison of Markov chains

Let $\{X(n),\ n \geq 0\}$ and $\{Y(n),\ n \geq 0\}$ two homogeneous discrete time Markov chains with probability transition matrices P and Q. If

- $X(0) \preceq_{st} Y(0)$
- **Comparison** $P \preceq_{st} Q \quad (\Longleftrightarrow P[i,*] \preceq_{\mathcal{F}} Q[i,*] \ \ \forall\ i)$
- **Monotonicity** $\quad P$ or Q is $\preceq_{st}$-monotone.

then

$$\{X(n)\} \ \preceq_{st} \ \{Y(n)\} \qquad\qquad (\mathbf{\Pi}^X(n) \preceq_{\mathcal{F}} \mathbf{\Pi}^Y(n)) \qquad \forall n$$

If steady state $\mathbf{\Pi}^X$ and $\mathbf{\Pi}^Y$ exist, then

$$\mathbf{\Pi}^X \preceq_{st} \mathbf{\Pi}^Y$$

Proposed Method

- Let $\mathcal{M}$ be the CTMC to check the underlying formula
- Construct by means of Stochastic Comparison a bounding chain $\mathcal{M}_{sup}$
- Check the formula through the bounding distributions

Motivations

1- Complexity reduction

2- Solution for intractable cases

- Infinite cases
- Partially defined models (Interval-valued Markov chains)

Model Checking by Special Structures

Model Checking by bounding Class $\mathcal{C}$ Chains (Ben Mamoun, Pekergin N., Younès QEST06)

- closed-form solutions for transient and the steady-state distributions, time to absorption
- simple characterizations for the stochastic monotonicity

Construction Algorithm based on stochastic monotonicity and comparison (Ben Mamoun and Pekergin N. PEIS 2000)

Worst-case complexity $\theta(N^2)$

Class $\mathcal{C}$ stochastic matrices

A stochastic matrix $P = (p_{i,j})_{1 \le i,j \le N}$ belongs to class $\mathcal{C}$ matrix, if for each column j, there is a real constant c_j such that

$$p_{i+1,j} = p_{i,j} + c_j, \qquad 1 \le i \le N-1$$

which is equivalent to

$$p_{i,j} = p_{1,j} + (i-1)\ c_j, \qquad 1 \le i,j \le N$$

Example :

$$P = \begin{pmatrix} 0.5 & 0.1 & 0.4 \\ 0.4 & 0.15 & 0.45 \\ 0.3 & 0.2 & 0.5 \end{pmatrix}$$

$$c_1 = -0.1,\ c_2 = 0.05\ , c_3 = 0.05 \qquad \sum_{j=1}^{n} c_j = 0$$

Class $\mathcal{C}$ stochastic matrices

A stochastic matrix P in class $\mathcal{C}$ can be represented by means of vectors:

$$\mathbf{P} = \mathbf{e}\ \mathbf{p} + \mathbf{d}\ \mathbf{c}$$

- $\mathbf{p}$ is the row vector representing the first row of $\mathbf{P}$
- $\mathbf{c}$ is the row vector for constants c_j
- $\mathbf{e}$, $\mathbf{d}$ are the column vectors such that

  $e_i = 1, \qquad d_i = (i-1), \qquad 1 \leq i \leq N$

$$\mathbf{P} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0.5 & 0.1 & 0.4 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \begin{pmatrix} -0.1 & 0.05 & 0.05 \end{pmatrix}$$

Two vectors to represent a class $\mathcal{C}$ chain

Properties of class $\mathcal{C}$ matrices

- **Power matrices :** $\mathbf{P} \in \mathcal{C} \rightarrow \mathbf{P}^n \in \mathcal{C}$:

$$\mathbf{P}^n = \mathbf{e}\ \mathbf{p} + [a^{n-1}\ \mathbf{d} + (b \sum_{k=0}^{n-2} a^k)\ \mathbf{e}]\ \mathbf{c}$$

$$a \ = \ \mathbf{c}\ \mathbf{d} = \sum_{k=1}^{N} (k-1) c_k, \quad b \ = \ \mathbf{p}\ \mathbf{d} = \ \sum_{k=1}^{N} (k-1) p_{1,k}$$

- **Closed-form solutions** to compute transient and the steady-state distributions for a DTMC with probability transition matrix $\mathbf{P} \in \mathcal{C}$

- **Steady-state distribution :**

$$\mathbf{\Pi} = \mathbf{p} + \frac{b}{1-a}\ \mathbf{c}$$

$a \neq 1$ if $\mathbf{P}$ is irreducible

Class $\mathcal{C}$ CTMC

By uniformization of the infinitesimal generator $\mathbf{Q}$:

$$\mathbf{P}_\lambda = \mathbf{I} + \frac{1}{\lambda}\mathbf{Q} \quad \text{where } \lambda \geq sup_i |q_{i,i}|$$

If the uniformized matrix $\mathbf{P}_\lambda \in \mathcal{C}$, **transient distribution :**

$$\mathbf{\Pi}(\mathbf{t}) = e^{-\lambda t}\mathbf{\Pi}(\mathbf{0}) + (1 - e^{-\lambda t})\mathbf{p} + \alpha(t)\ \mathbf{c}$$

$$\begin{aligned}\alpha(t) = & \quad e^{-\lambda t} \sum_{n=1}^{\infty} \frac{(\lambda t)^n}{n!} \alpha_n \\ = & \begin{cases} b\frac{1-e^{-\lambda t}}{1-a} + (\frac{g}{a} - \frac{b}{a(1-a)})e^{-\lambda t}(e^{\lambda t a} - 1), & \text{if } a \neq 1, a \neq 0 \\ e^{-\lambda t}(\lambda t g - \lambda t b - b) + b, & \text{if } a = 0 \\ b\lambda t + (g - b)(1 - e^{-\lambda t}), & \text{if } a = 1 \end{cases}\end{aligned}$$

Reduction of Complexities

- **Storage** of only vectors for class $\mathcal{C}$ instead of the matrix
- **Steady-state analysis:**

  Class $\mathcal{C}$: $\theta(N)$
- **Transient analysis :**

  Class $\mathcal{C}$: $\theta(N)$

Stochastic Monotonicity of Class $\mathcal{C}$ chains

Probability transition matrix P is $\preceq_{\mathcal{F}}$-monotone, if for all probability vectors p and q,

$$p \preceq_{\mathcal{F}} q \implies pP \preceq_{\mathcal{F}} qP$$

Simple characterization for class $\mathcal{C}$:

- $\leq_{st}$ monotone:

  $P \preceq_{st}$ -monotone $\iff \sum_{k=j}^{n} c_k \geq 0, \ \forall$ column j

- $\leq_{icx}$ monotone:

  $P \preceq_{icx}$ -monotone $\iff \sum_{k=j}^{n}(k-j+1) \ c_k \geq 0, \ \forall \ j$

Proposed Algorithm

Input: DTMC $\mathcal{M}$, initial state s,

formula $Fr \in \{\mathcal{P}_{\lhd p}(\phi_1 \, \mathcal{U}^I \phi_2), \; \mathcal{T}^{@t}_{\lhd p}(\phi), \; \mathcal{S}_{\lhd p}(\phi)\}$

Output: Three cases: $1.s \models Fr$, $2.s \not\models Fr$ 3. It is not possible to decide.

1. Transformation of the model if Fr is path-based (*for transient analysis)*
2. Determination of S_Σ states
3. State space organization

   Aggregation of S_Σ states if they are absorbing (*necessary for* $\leq_{icx}$)

   Aggregation of other absorbing states (*optional*)

   Reorder state space to put S_Σ states at the end (*stochastic ordering*)

$\mathcal{M}^{Ta}$ *DTMC that must be analyzed*

4. Construction of bounding chains : $\mathbf{P}^{\mathcal{M}^{Ta}_{sup}}$ is a class $\mathcal{C}$, monotone, bounding matrix in the sense of $\leq_{\mathcal{F}}$

6. Computing bounding distributions through closed-form solutions

$$\mathcal{F} = st : \qquad \mathbf{\Pi}(S_\Sigma) \leq \mathbf{\Pi}_{sup}(S_\Sigma)$$

7. Computing bounding probabilities $\mathcal{B}_{inf}$ and $\mathcal{B}_{sup}$ to $P_{Fr}(S_\Sigma)$ to check the formula:
   - $B_{sup} \leq p$, then Fr is true
   - $B_{inf} > p$, then Fr is false
   - otherwise, it can not be decided

Conclusions

- First step rapid model checking
- Including the proposed method in model checkers does not increase significantly the complexity but may decrease largely the overall complexities for some cases

Model Checking by simulation

Verification by statistical tests

Software for Perfect Simulation developed by project MESCAL, INRIA Rhône-Alpes

Model Checking by Perfect Simulation?

Perspectives

- Infinite model checking (AMSTA08)
- Interval-valued Markov chains
- Other Formalisms
- Software