DTMC Model Checking by SCC Reduction

Erika Ábrahám, Nils Jansen, Ralf Wimmer, Joost-Pieter Katoen, Bernd Becker

Theory of Hybrid Systems
RWTH Aachen

June 7, 2010
Probabilistic Model Checking
Counterexamples for Probabilistic Systems

“It is impossible to overestimate the importance of the counterexample feature. The counterexamples are invaluable in debugging complex systems. Some people use model checking just for this feature.”

Edmund Clarke, Turing Award 2007, in “The birth of model checking”, 2008
Model Checking in general provides diagnostic information (CTL, LTL).

Erroneous run is called counterexample.

Classical Probabilistic Model Checking provides no diagnostic information.

State-of-the-art counterexample generation for probabilistic systems needs the Model Checking result.
Model Checking in general provides diagnostic information (CTL, LTL).

Erroneous run is called counterexample.

Classical Probabilistic Model Checking provides no diagnostic information.

State-of-the-art counterexample generation for probabilistic systems needs the Model Checking result.

Goal: Development of a Model Checking Algorithm with simultaneous Counterexample Generation

- Counterexamples may consist of a big number of paths.
- \(\Rightarrow\) Abstract but refinable counterexamples
- \(\Rightarrow\) Save information during model checking and abstraction.
Definition

A discrete-time Markov chain (DTMC) is a tuple $M = (S, s_{init}, \mathbb{P}, L)$ with finite state space $S$, initial state $s_{init}$, state labeling $L : S \to 2^{AP}$ and transition probability matrix $\mathbb{P} : S \times S \to [0, 1]$. 
Markov Chain

Definition
A *discrete-time Markov chain (DTMC)* is a tuple $M = (S, s_{init}, P, L)$ with finite state space $S$, initial state $s_{init}$, state labeling $L : S \rightarrow 2^{AP}$ and transition probability matrix $P : S \times S \rightarrow [0, 1]$.

Andrey Andreyevich Markov (1856-1922)
Markov Chain

Definition

A **discrete-time Markov chain (DTMC)** is a tuple $M = (S, s_{init}, P, L)$ with finite state space $S$, initial state $s_{init}$, state labeling $L : S \rightarrow 2^{AP}$ and transition probability matrix $P : S \times S \rightarrow [0, 1]$. 

![Diagram of a Markov Chain]
Probabilistic CTL (Hansson & Jonsson, 1994)

**Definition**
The syntax of **Probabilistic Computation Tree Logic** is given by

\[ \varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid P_{\sim \lambda}(\varphi U \varphi) \]

with \( p \in AP \), \( \lambda \in [0, 1] \subseteq \mathbb{R} \), and \( \sim \in \{<, \leq, \geq, >\} \).
Probabilistic CTL (Hansson & Jonsson, 1994)

Definition
The syntax of Probabilistic Computation Tree Logic is given by

\[ \varphi ::= p | \neg \varphi | \varphi \land \varphi | \mathbb{P}_{\sim \lambda}(\varphi \ U \ \varphi) \]

with \( p \in AP, \lambda \in [0, 1] \subseteq \mathbb{R}, \) and \( \sim \in \{<, \leq, \geq, >\} \).

Property:

\[ \mathbb{P}_{\leq 0.02}(true \ U \ s_5) \]
Counterexamples

Property:

\[ \mathbb{P}_{\leq 0.02}(\text{true } U \ s_5) \]
Counterexamples

Property: \( \mathbb{P}_{\leq 0.02}(\text{true } U s_5) \)

\[ \pi_1 : s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_5 \text{ with } Pr_{\text{fin}}(\pi_1) = 0.02 \sim \sim \text{Property is true?} \]
Counterexamples

Property:

\[ P_{\leq 0.02}(\text{true } U s_5) \]

- \( \pi_1 : s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_5 \) with \( Pr_{\text{fin}}(\pi_1) = 0.02 \) \( \rightsquigarrow \) Property is true?
- \( \pi_2 : s_1 \rightarrow s_2 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_5 \) with \( Pr_{\text{fin}}(\pi_2) = 0.004 \)
Counterexamples

Property:

$$P_{\leq 0.02}(\text{true } U \ s_5)$$

- $$\pi_1 : s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_5 \text{ with } Pr_{\text{fin}}(\pi_1) = 0.02 \leadsto \text{Property is true?}$$
- $$\pi_2 : s_1 \rightarrow s_2 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_5 \text{ with } Pr_{\text{fin}}(\pi_2) = 0.004$$
- $$Pr_{\text{fin}}(\pi_1) + Pr_{\text{fin}}(\pi_2) = 0.024 > 0.02 \leadsto \text{Property is false!}$$
Counterexamples

Property: \( \mathbb{P}_{\leq 0.02}(\text{true } U \ s_5) \)

- \( \pi_1 : s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_5 \) with \( Pr_{\text{fin}}(\pi_1) = 0.02 \) \( \sim \) Property is true?
- \( \pi_2 : s_1 \rightarrow s_2 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_5 \) with \( Pr_{\text{fin}}(\pi_2) = 0.004 \)
- \( Pr_{\text{fin}}(\pi_1) + Pr_{\text{fin}}(\pi_2) = 0.024 > 0.02 \) \( \sim \) Property is false!

\( \sim \) Set \( C = (\pi_1, \pi_2) \) is a counterexample for the property.

(With well-defined probability mass on (in-)finite paths, set of paths...)

AlgoSyn - Nils Jansen - SCC-based Model Checking
Strongly Connected Components

Digraph

- **SCC**: Set of nodes where all nodes are reachable from each other.
- **Bottom-SCC**: SCC has no outgoing edges.

DTMC

- Every run of a DTMC will eventually reach a **Bottom-SCC** (and stay there).
- **Non-Bottom-SCC** will be left at some point in time with probability 1.
Strongly Connected Components

Digraph

- **SCC**: Set of nodes where all nodes are reachable from each other
- **Bottom-SCC**: SCC has no outgoing edges.

DTMC

- Every run of a DTMC will eventually reach a **Bottom-SCC** (and stay there).
- **Non-Bottom-SCC** will be left at some point in time with probability 1

Example:

```
\begin{align*}
S_1 & \xrightarrow{0.99} S_1 \\
S_1 & \xrightarrow{0.01} S_2 \\
S_2 & \xrightarrow{} S_2
\end{align*}
```

- The path that stays in $s_1$ has probability 0
Example:

- The path that stays in $s_1$ has probability 0
- All paths that eventually leave $s_1$:

$$(s_1, s_2), (s_1, s_1, s_2), \ldots, (s_1, \ldots, s_1, s_2)$$
Example:

- The path that stays in $s_1$ has probability 0
- All paths that eventually leave $s_1$:
  
  $$(s_1, s_2), (s_1, s_1, s_2), \ldots, (s_1, \ldots, s_1, s_2)$$

- Probability of all these paths:
  
  $0.01 + 0.99 \cdot 0.01 + 0.99 \cdot 0.99 \cdot 0.01 \ldots$
Example:

- The path that stays in $s_1$ has probability 0.
- All paths that eventually leave $s_1$:
  \[(s_1, s_2), (s_1, s_1, s_2), \ldots, (s_1, \ldots, s_1, s_2)\]

- Probability of all these paths:
  \[0.01 + 0.99 \cdot 0.01 + 0.99 \cdot 0.99 \cdot 0.01 \ldots\]
  \[= \left( \sum_{i \in \mathbb{N}} 0.99^i \right) \cdot 0.01\]
Example:

- The path that stays in $s_1$ has probability 0.
- All paths that eventually leave $s_1$:
  
  $(s_1, s_2), (s_1, s_1, s_2), \ldots, (s_1, \ldots, s_1, s_2)$

- Probability of all these paths:
  
  $0.01 + 0.99 \cdot 0.01 + 0.99 \cdot 0.99 \cdot 0.01 \ldots$

  $= \left( \sum_{i \in \mathbb{N}} 0.99^i \right) \cdot 0.01$

  $= \frac{1}{1 - 0.99} \cdot 0.01 = 1$
Contents

1 Motivation

2 Preliminaries

3 SCC-based Model Checking

4 Interactive Counterexample Refinement

5 Implementation and Case Studies

6 Conclusion and Future Work
Observation

- Paths through a non-bottom-SCC will leave it with probability 1.
- For unbounded reachability we look at the whole probability mass provided by SCCs.

Idea

- Reduce SCC to abstract states and edges carrying the whole probability mass.

Approaches

- State elimination
  - Finite Automaton $\rightarrow$ Regular Expression (Kleene)
  - Markov Chain: Compute Probabilities (Daws 2004, Damman, Han and Katoen 2008 and Hahn, Hermanns and Zhang, 2009)
- Our approach: Recursive Algorithm
  - Bottom-Up computing
  - Use specific properties of Markov Chains
State Elimination

\[
\text{Probability of reaching } 2 \text{ (from } 1) = p_1 + (p_2 \cdot p_3) + (p_2 \cdot p_4 \cdot p_3) + \ldots
\]

\[
= p_1 + \sum_{i \in N} p_2 \cdot p_i \cdot p_3
\]

\[
= p_1 + p_2 \cdot 1 - p_4 \cdot p_3
\]
State Elimination

Probability of reaching 2 (from 1):

\[ p_1 + (p_2 \cdot p_3) + (p_2 \cdot p_4 \cdot p_3) + (p_2 \cdot p_4^2 \cdot p_3) \ldots = p_1 + \sum_{i \in \mathbb{N}} p_2 \cdot p_4^i \cdot p_3 \]

\[ = p_1 + p_2 \cdot \sum_{i \in \mathbb{N}} p_4^i \cdot p_3 = p_1 + p_2 \cdot \frac{1}{1 - p_4} \cdot p_3 \]
**State Elimination**

Probability of reaching 2 (from 1):

\[ p_1 + (p_2 \cdot p_3) + (p_2 \cdot p_4 \cdot p_3) + (p_2 \cdot p_4^2 \cdot p_3) \ldots = p_1 + \sum_{i \in \mathbb{N}} p_2 \cdot p_4^i \cdot p_3 \]

\[ = p_1 + p_2 \cdot \sum_{i \in \mathbb{N}} p_4^i \cdot p_3 = p_1 + p_2 \cdot \frac{1}{1 - p_4} \cdot p_3 \]
State Elimination

Probability of reaching 2 (from 1):

\[ p_1 + (p_2 \cdot p_3) + (p_2 \cdot p_4 \cdot p_3) + (p_2 \cdot p_4^2 \cdot p_3) \ldots = p_1 + \sum_{i \in \mathbb{N}} p_2 \cdot p_4^i \cdot p_3 \]

\[ = p_1 + p_2 \cdot \sum_{i \in \mathbb{N}} p_4^i \cdot p_3 = p_1 + p_2 \cdot \frac{1}{1 - p_4} \cdot p_3 \]
SCC with one input node

Example SCC

- \( s_1 \)
- \( s_2 \)
- \( s_3 \)

Probability of coming back to input node can be computed without considering paths that lead back to it!
SCC with one input node

Example SCC and abstract view

Probability of coming back to input node can be computed without considering paths that lead back to it!
SCC with one input node

Example SCC and more abstract view

The probability of coming back to the input node can be computed without considering paths that lead back to it!
SCC with one input node

Example SCC: Paths leading back to input state are ignored

![Diagram showing a strongly connected component with one input node]

1. The probability of coming back to the input node can be computed without considering paths that lead back to it!
SCC with one input node

Probability mass of all paths entering and eventually leaving SCC

\[ \sum_{i \in \mathbb{N}} p_1^i \cdot (p_2 + p_3) = \frac{1}{1 - p_1} \cdot (p_2 + p_3) \]

Prob. for leaving SCC is 1

\[ \frac{1}{1 - p_1} \cdot (p_2 + p_3) = 1 \]

\( \Leftrightarrow p_1 = 1 - p_2 - p_3 \)
SCC with one input node

Probability mass of all paths entering and eventually leaving SCC

\[ \sum_{i \in \mathbb{N}} p_i^1 \cdot (p_2 + p_3) = \frac{1}{1 - p_1} \cdot (p_2 + p_3) \]

Prob. for leaving SCC is 1

\[ \frac{1}{1 - p_1} \cdot (p_2 + p_3) = 1 \]

\[ \Leftrightarrow p_1 = 1 - p_2 - p_3 \]

Probability of coming back to input node can be computed without considering paths that lead back to it!
SCC with one input node

Probability of reaching $s_2$:

$$\frac{1}{1 - p_1 \cdot p_2}$$
Probability of reaching $s_2$:

$$\frac{1}{1 - p_1 \cdot p_2}$$

Discard $p_1$ and scale $p_2$ and $p_3$:

$$p_2' = \frac{p_2}{p_2 + p_3}$$

$$p_3' = \frac{p_3}{p_2 + p_3}$$
SCC with one input node

Probability of reaching $s_2$:

\[
\frac{1}{1 - p_1} \cdot p_2
\]

Discard $p_1$ and scale $p_2$ and $p_3$:

\[
p_2' = \frac{p_2}{p_2 + p_3}
\]

\[
p_3' = \frac{p_3}{p_2 + p_3}
\]

Scaling preserves reachability probabilities:

(As before: $p_3 = 1 - p_2 - p_1$)

\[
p_2' = \frac{p_2}{p_2 + p_3} = \frac{p_2}{p_2 + (1 - p_2 - p_1)} = \frac{p_2}{1 - p_1}
\]
Example

Whole graph:
- Search for SCCs
Example

SCC 1

Graph showing states $S_1$ to $S_9$ with labeled transitions.
Example

SCC 1:
- Ignore
- Search for SCCs

SCC 1:
- Ignore
- Search for SCCs
Example

SCC 1

SCC 1.1

SCC 1.2

AlgoSyn
Nils Jansen - SCC-based Model Checking
Example

SCC 1

SCC 1.1

SCC 1.2

SCC 1.1:
- Ignore
- Input-Node

Search for SCCs

SCC 1.2:
- Ignore
- Input-Node

Search for SCCs
Example

SCC 1

SCC 1.1

SCC 1.2

SCC 1.2.1

SCC 1.1:

- No SCCs found!
Example

SCC 1

SCC 1.1

SCC 1.2

SCC 1.2.1

SCC 1.2:

- Ignore Input-Node
- Search for SCCs
SCC 1

SCC 1.1

SCC 1.2

SCC 1.2.1:

- No SCCs found!
- Compute probabilities for reaching output nodes from input node.
Example

SCC 1

SCC 1.1

SCC 1.2

SCC 1.2.1:

- Scale probabilities.
Example

SCC 1

SCC 1.1

SCC 1.2

SCC 1.2.1
Example

SCC 1

SCC 1.1

SCC 1.2

SCC 1.2.1

Reduced to abstract node.

SCC 1.2.1:
SCC 1

SCC 1.1

SCC 1.2

SCC 1.2.1

SCC 1.1

SCC 1.2

SCC 1.2.1

Example

SCC 1:

SCC 1.1

SCC 1.2

SCC 1.2.1

SCC 1.2:

- Compute probabilities for reaching output nodes from input node.
Example

SCC 1

SCC 1.1

SCC 1.2

SCC 1.2.1

SCC 1.2:

- Scale probabilities.
Example

SCC 1

SCC 1.1

SCC 1.2

SCC 1.2.1

AlgoSyn Nils Jansen - SCC-based Model Checking
Example

SCC 1

SCC 1.1

SCC 1.2

SCC 1.2.1

SCC 1.2:
- Reduced to abstract node.
Example

SCC 1

SCC 1.1

SCC 1.1:
- Compute probabilities for reaching output nodes from input node.
Example

SCC 1

SCC 1.1

SCC 1.2

SCC 1.2.1

SCC 1.1:
- Scale probabilities.

Nils Jansen - SCC-based Model Checking
Example

SCC 1

SCC 1.1

SCC 1.1:
- Reduced to abstract node.

SCC 1.2

SCC 1.2.1
Example

SCC 1: Compute probabilities for reaching output nodes from input node.
Example

SCC 1:
- Scale probabilities.

SCC 1.1

SCC 1.2

SCC 1.2.1

SCC 1

S₁

S₅

S₉

0.3886

0.3393

1

1
Example

SCC 1

SCC 1.1

SCC 1.2

SCC 1.2.1

S₁

0.4661

0.5339

S₅

S₉

1

1
Example

SCC 1:
- Reduced to abstract node.
- Model Checking result for reachability
Contents

1 Motivation

2 Preliminaries

3 SCC-based Model Checking

4 Interactive Counterexample Refinement

5 Implementation and Case Studies

6 Conclusion and Future Work
Towards Counterexamples

- Information of original Markov Chain is saved.
- SCCs are represented by abstract nodes.
- Abstract paths may contain both abstract and concrete nodes.