Probabilistic Counterexamples

Ralf Wimmer
Albert-Ludwigs-Universität Freiburg, Germany
Saarland University, Saarbrücken, Germany

2nd AVACS Autumn School, Oldenburg, October 2, 2015
Acknowledgements

Most of what I present is joint work with

- Nils Jansen,
- Erika Ábrahám,
- Joost-Pieter Katoen, and
- Bernd Becker.
Introduction
Dave Parker did a good job yesterday, motivating the relevance of probabilistic systems and laying the foundations for counterexamples!

▶ Here: only a short reminder of the central notions.
Definition: DTMCs

Let $AP$ be a finite set of atomic propositions. A **discrete-time Markov chain** $M$ is a tuple $M = (S, s_{\text{init}}, P, L)$ such that

- $S$ is a finite set of **states**,
- $s_{\text{init}} \in S$ the **initial state**,
- $P : S \times S \to [0, 1]$ the **transition probability matrix** with $\sum_{s' \in S} P(s, s') \leq 1$ for all $s \in S$, and
- $L : S \to 2^{AP}$ a **labeling function**, assigning the set of true propositions to each state.
Definition: DTMCs

Let $AP$ be a finite set of atomic propositions. A **discrete-time Markov chain** $M$ is a tuple $M = (S, s_{\text{init}}, P, L)$ such that

- $S$ is a finite set of **states**, 
- $s_{\text{init}} \in S$ the **initial state**, 
- $P : S \times S \to [0, 1]$ the **transition probability matrix** with $\sum_{s' \in S} P(s, s') \leq 1$ for all $s \in S$, and 
- $L : S \to 2^AP$ a **labeling function**, assigning the set of true propositions to each state.
Definition: MDPs

Let $AP$ be a set of atomic propositions. A **discrete-time Markov decision process** $M$ is a tuple $M = (S, s_{\text{init}}, A, P, L)$ such that

- $S$, $s_{\text{init}}$, and $L$ are as for DTMCs,
- $A$ is a finite set of actions, and
- $P : S \times A \times S \to [0, 1]$ is a **transition probability matrix** such that $\sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\}$ for all $s \in S$ and $\alpha \in A$. 
Definition: MDPs

Let \( AP \) be a set of atomic propositions. A **discrete-time Markov decision process** \( M \) is a tuple \( M = (S, s_{\text{init}}, A, P, L) \) such that

- \( S \), \( s_{\text{init}} \), and \( L \) are as for DTMCs,
- \( A \) is a finite set of actions, and
- \( P : S \times A \times S \to [0, 1] \) is a transition probability matrix such that \( \sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\} \) for all \( s \in S \) and \( \alpha \in A \).
The non-determinism is resolved by a **scheduler**. It assigns to each finite path a distribution over the actions possible in the last state.
The non-determinism is resolved by a scheduler. It assigns to each finite path a distribution over the actions possible in the last state. A deterministic scheduler puts the whole probability into a unique action-distribution pair. The decisions made by a memoryless scheduler depend only on the last state of the path.
The non-determinism is resolved by a scheduler. It assigns to each finite path a distribution over the actions possible in the last state. A deterministic scheduler puts the whole probability into a unique action-distribution pair. The decisions made by a memoryless scheduler depend only on the last state of the path.

Each scheduler for an MDP/PA induces a DTMC.
Probabilistic Safety

Safety of DTMCs

Is the probability to eventually enter an unsafe state (labeled with “unsafe”) at most $\lambda$?

$$\mathcal{P}_{\leq \lambda}(\mathcal{F}_{\text{unsafe}})$$

Probability computation

Solve the following linear equation system:

$$x_s = \begin{cases} 
1 & \text{if } s \models \text{unsafe}, \\
0 & \text{if all unsafe states are unreachable from } s, \\
\sum_{s' \in S} P(s, s') \cdot x_{s'} & \text{otherwise}. 
\end{cases}$$
Safety of MDPs

Is the maximal probability to reach an unsafe state at most $\lambda$?

Probability computation

Solve the following linear program:

$$\text{minimize} \sum_{s \in S} x_s$$

such that

for $s \in T$:

$$x_s = 1$$

for $s$ with $T$ unreachable:

$$x_s = 0$$

otherwise, for $s \in S, a \in A$:

$$x_s \geq \sum_{s' \in S} P(s, a, s') \cdot x_{s'}$$

The equation system can be rewritten into a linear program, which can be solved, e.g., using the Simplex algorithm.
Reminder: Probabilistic Model Checking
Safety of MDPs

Safety of MDPs

Is the maximal probability to reach an unsafe state at most $\lambda$?

Probability computation

Solve the following linear program:

$$\text{minimize} \quad \sum_{s \in S} x_s$$

such that

for $s \in T$: $x_s = 1$

for $s$ with $T$ unreachable: $x_s = 0$

otherwise, for $s \in S, a \in A$: $x_s \geq \sum_{s' \in S} P(s, \alpha, s') \cdot x_{s'}$
Non-probabilistic Systems

Examples:
- digital circuits
- software
- hybrid systems

Safety: The system will never enter an unsafe state.

Counterexample: Trace (sequence of inputs and successor states) leading from the initial state to an unsafe state.
By-product of model checking:

- **bounded model checking (BMC):** Satisfying assignment of the BMC-formula corresponds to a counterexample.
- **state space traversal (explicit):** Store the current path during depth-first search.
- **state space traversal (symbolic):** Store intermediate state sets during forward traversal and extract a cex by walking backward.
- **LTL model checking:** Accepting run of the Büchi automaton.
“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 Winner 2007

Applications of cex:
- System debugging (fault reproduction / diagnosis)
- Counterexample-guided abstraction refinement (CEGAR)
Challenges:

- Algorithms only yield probabilities, but no counterexamples.
- A single trace to an error state typically does not suffice.
Aspects of probabilistic counterexamples:

- Counter-examples
  - Representation
    - Symbolic
    - Explicit
  - Executions
  - Level
  - Description

- Heuristic
  - Optimal
  - Optimality

- Property
  - Safety
  - LTL/ω-reg.
  - (nested) PCTL

- System Type
  - DTMCs
  - MDPs

- Reward

Description of the diagram:

- **Counter-examples**
  - Heuristic
  - Optimal
  - Optimality

- Property
  - Safety
  - LTL/ω-reg.
  - (nested) PCTL

- System Type
  - DTMCs
  - MDPs

- Reward

- Description
  - Symbolic
  - Explicit
  - Executions
  - Level
Overview

Introduction
  Probabilistic Model Checking
  Non-Probabilistic Counterexamples

Path-based Counterexamples

Computation of Minimal Critical Subsystems

Symbolic Computation of Critical Subsystems

High-level counterexamples
Path-based Counterexamples
Aspects of probabilistic counterexamples:

- Counter-examples
- System Type
  - DTMCs
  - MDPs
- Property
  - Safety
  - LTL/ω-reg.
  - (nested) PCTL
- Reward
- Level
- Executions
- States
- Symbolic
- Explicit
- Representation
- Optimal
- Optimality
- Heuristic
Non-prob. cex: 1 trace

Prob. cex: set of traces with enough probability
Adaptation of Non-Probabilistic Cex

- **Non-prob. cex**: 1 trace
- **Prob. cex**: set of traces with enough probability $\mathcal{P}_{\leq 0.5}(\mathcal{F}_{\text{unsafe}})$

$$\mathcal{P}_{\leq 0.5}(\mathcal{F}_{\text{unsafe}})$$

![Diagram with transitions and probabilities](attachment:image.png)
Adaptation of Non-Probabilistic Cex

- **Non-prob. cex:** 1 trace
- **Prob. cex:** set of traces with enough probability

\[ P_{\leq 0.5}(\mathcal{F}_{unsafe}) \]

Counterexample:
- \( s \rightarrow s_1 \rightarrow t_1 \)
- \( s \rightarrow s_1 \rightarrow s_2 \rightarrow t_1 \)
- \( s \rightarrow s_1 \rightarrow s_2 \rightarrow t_2 \)
  
  Prob: 0.52
Consider a violated safety property $P_{\leq \lambda}(\mathcal{F} \text{ unsafe})$.

- **Evidence:** Any finite path $\pi$ starting in $s_{\text{init}}$ and ending upon the first visit of an unsafe state.

- **Strongest evidence:** evidence $\pi^*$ such that $\Pr(\pi^*) \geq \Pr(\pi)$ for all evidences $\pi$.

- **Counterexample:** Set $C$ of evidences such that $\Pr(C) > \lambda$

- **Minimal counterexample:** Counterexample $C^*$ such that $|C^*| \leq |C|$ for all cex $C$.

- **Smallest counterexample:** Counterexample $C^*$ such that $\Pr(C^*) \geq \Pr(C)$ for all minimal cex $C$. 
Example

Evidences:
- $s \rightarrow s_1 \rightarrow t_1$, prob = 0.2
- $s \rightarrow s_1 \rightarrow s_2 \rightarrow t_1$, prob = 0.2
- $s \rightarrow s_2 \rightarrow t_1$, prob = 0.15
- $s \rightarrow s_1 \rightarrow s_2 \rightarrow t_2$, prob = 0.12
- $s \rightarrow s_2 \rightarrow t_2$, prob = 0.09

No evidences:
- $s_1 \rightarrow s_2 \rightarrow t_1$
- $s \rightarrow s_1 \rightarrow t_1 \rightarrow t_2$

Strongest evidences:
- $s \rightarrow s_1 \rightarrow t_1$
- $s \rightarrow s_1 \rightarrow s_2 \rightarrow t_1$
Example

\[ P_{\leq 0.5}(\mathcal{F}_{\text{unsafe}}) \]

Counterexamples:

- \( s \rightarrow s_1 \rightarrow t_1 \)
- \( s \rightarrow s_1 \rightarrow s_2 \rightarrow t_1 \)
- \( s \rightarrow s_1 \rightarrow t_1 \)

Prob: 0.55

- \( s \rightarrow s_1 \rightarrow t_1 \)
- \( s \rightarrow s_1 \rightarrow s_2 \rightarrow t_1 \)
- \( s \rightarrow s_1 \rightarrow s_2 \rightarrow t_2 \)

Prob: 0.52

- \( s \rightarrow s_1 \rightarrow s_2 \rightarrow t_1 \)
- \( s \rightarrow s_1 \rightarrow s_2 \rightarrow t_2 \)
- \( s \rightarrow s_2 \rightarrow t_1 \)
- \( s \rightarrow s_2 \rightarrow t_2 \)

Prob: 0.56
Example

\[ \mathcal{P}_{\leq 0.5}(\mathcal{F}_{\text{unsafe}}) \]

**Minimal Counterexamples:**

- \( s \rightarrow s_1 \rightarrow t_1 \)
  - \( s \rightarrow s_1 \rightarrow s_2 \rightarrow t_1 \)
  - \( s \rightarrow s_1 \rightarrow t_1 \)
  - Prob: 0.55

- \( s \rightarrow s_1 \rightarrow t_1 \)
  - \( s \rightarrow s_1 \rightarrow s_2 \rightarrow t_1 \)
  - \( s \rightarrow s_1 \rightarrow s_2 \rightarrow t_2 \)
  - Prob: 0.52
Example

\[ \mathcal{P}_{\leq 0.5}(\mathcal{F}_{\text{unsafe}}) \]

**Smallest Counterexamples:**

- \( s \rightarrow s_1 \rightarrow t_1 \)
- \( s \rightarrow s_1 \rightarrow s_2 \rightarrow t_1 \)
- \( s \rightarrow s_1 \rightarrow t_1 \)
- Prob: 0.55
Computation of Smallest Cex

Transformation into a shortest-paths problem:

1. Add a single deadlock target state \( t \); redirect all out-going transitions from unsafe states to \( t \)

2. Define weighted digraph \( G = (S, E, w) \):

\[(s, s') \in E \iff P(s, s') > 0 \quad \text{and} \quad w(s, s') = -\log P(s, s')\]
Lemma

The $k$ shortest path from $s_{\text{init}}$ to $t$ in the weighted digraph corresponds to the $k$-most probable evidence in the DTMC.
**Shortest Paths**

**Lemma**

The $k$ shortest path from $s_{\text{init}}$ to $t$ in the weighted digraph corresponds to the $k$-most probable evidence in the DTMC.

The computation of a smallest cex is a $k$-shortest paths problem in a weighted digraph with non-negative weights.

**Available Algorithms:**

- Jiménez/Marzal (Proc. of WAE, 1999)
- K* by Aljazzar/Leue (Artif. Intell., 2011)
Challenges

Counterexample = \( k \) shortest paths

Does this solve the counterexample problem?

Clearly: **NO!**

**Limiting factors:**

- size of the DTMC
- size of the path set
- models with non-determinism (MDPs)
Challenges

**Counterexample** = \( k \) shortest paths

Does this solve the counterexample problem?

Clearly: **NO**!

**Limiting factors:**

- *size of the DTMC*
  - sometimes millions or billions of states
- size of the path set
- models with non-determinism (MDPs)
Counterexample = $k$ shortest paths

Does this solve the counterexample problem?

Clearly: **NO**!

**Limiting factors:**

- *size of the DTMC*
  - sometimes millions or billions of states

- *size of the path set*
  - number of paths often larger than the number of states

- models with non-determinism (MDPs)
Counterexample = $k$ shortest paths

Does this solve the counterexample problem?

Clearly: **NO**!

**Limiting factors:**

- *size of the DTMC*
  - sometimes millions or billions of states
- *size of the path set*
  - number of paths often larger than the number of states
- *models with non-determinism (MDPs)*
  - all paths must resolve the non-determinism in the same way
Property: \( \Pr_{\leq 0.15}(\mathcal{F}_{\text{unsafe}}) \)

- Probability of each path: \( 0.1 \cdot (0.5)^{n-1} \)
- Number of paths: \( 2^n \) \((n = \text{number of branchings})\)
- Number of paths needed: \( \frac{0.15}{0.2} \cdot 2^n + 1 \)

⇒ exponential in the number of states.
Counterexamples can be even infinite sets

Property: \( \mathcal{P}_{<0.5}(\mathcal{F}_{\text{unsafe}}) \)
Counterexamples can be even infinite sets

Property: \( P_{<0.5}(\mathcal{F} unsafe) \)

Consider set \( C \) of all paths leading to state \( s_2 \):

\[
C = \{ (s_0) \rightarrow s_2, \ (s_0)^2 \rightarrow s_2, \ (s_0)^3 \rightarrow s_2, \ldots \}
\]

Probability of \( C \): \( \sum_{i=0}^{\infty} (0.5)^i \cdot 0.25 \)
Counterexamples can be even infinite sets

\[ S_1 \xrightarrow{0.25} S_0 \xrightarrow{0.25} S_2 \]

\[ \text{Property: } P_{<0.5}(F \text{ unsafe}) \]

Consider set \( C \) of all paths leading to state \( s_2 \):

\[ C = \{(s_0) \rightarrow s_2, \ (s_0)^2 \rightarrow s_2, \ (s_0)^3 \rightarrow s_2, \ldots\} \]

Probability of \( C \):

\[ \sum_{i=0}^{\infty} (0.5)^i \cdot 0.25 \quad \text{geom. ser.} \quad \frac{1}{1-0.5} \cdot 0.25 \]
Counterexamples can be even infinite sets

Property: $P_{<0.5}(\mathcal{F}_{\text{unsafe}})$

Consider set $C$ of all paths leading to state $s_2$:

$$C = \{(s_0) \to s_2, \ (s_0)^2 \to s_2, \ (s_0)^3 \to s_2, \ldots\}$$

Probability of $C$: $\sum_{i=0}^{\infty} (0.5)^i \cdot 0.25$ \text{geom. ser.} = $\frac{1}{1-0.5} \cdot 0.25 = 0.5$
Counterexamples can be represented
- by enumeration of the paths,
- by regular expressions, trees, ...  
- critical subsystems [Aljazzar/Leue, 2009; Jansen et al., 2011].

Critical subsystem

Subset $S'$ of the states such that the probability of reaching an unsafe-state visiting only states from $S'$ is already beyond $\lambda$. 
Computation of Minimal Critical Subsystems
Aspects of probabilistic counterexamples:

- Executions
- States
- Level
- Symbolic
- Explicit
- Representation
- Heuristic
- Optimal
- Optimality
- Reward
- Property
- Safety
- (nested) PCTL
- LTL/ω-reg.
- System Type
- DTMCs
- MDPs
- Property

Counterexamples
$P_{\leq 0.25}(\mathcal{F}_{\text{unsafe}})$
Critical subsystems for DTMCs: Example

\[ P_{\leq 0.25}(F_{\text{unsafe}}) \]

![Diagram with states and transition probabilities]
Formulate minimal critical subsystems as an optimization problem:

- $\lambda$: probability bound
- $x_s \in \{0, 1\} \subseteq \mathbb{Z}$ with $x_s = 1$ iff $s$ belongs to the subsystem
- $p_s \in [0, 1] \subseteq \mathbb{R}$: probability of state $s$ within the subsystem
Formulate minimal critical subsystems as an optimization problem:

- \( \lambda \): probability bound
- \( x_s \in \{0, 1\} \subseteq \mathbb{Z} \) with \( x_s = 1 \) iff \( s \) belongs to the subsystem
- \( p_s \in [0, 1] \subseteq \mathbb{R} \): probability of state \( s \) within the subsystem

**Mixed-integer linear program**

\[
\text{minimize} \quad \left( -\frac{1}{2}p_{s_{\text{init}}} + \sum_{s \in S} x_s \right)
\]

such that

- \( p_{s_{\text{init}}} > \lambda \)
- \( \forall s \in T : x_s = p_s \)
- \( \forall s \in S \setminus T : p_s \leq x_s \)
- \( \forall s \in S \setminus T : p_s \leq \sum_{s' \in S} P(s, s') \cdot p_{s'} \)
The computation time can be reduced by adding redundant constraints:

- Each state (except $s_{\text{init}}$) has a predecessor state in the subsystem
- Each state (except unsafe states) has a successor state in the subsystem
- Generalize this to strongly connected components
- Require that each state in the subsystem is reachable from $s_{\text{init}}$
- Require that each state in the subsystem can reach an unsafe state

▶ **Trade-off between additional constraints and size of search space**
Some results for DTMCs

Benchmarks:
- Crowds protocol
  - Randomized protocol for anonymous surfing
- Synchronous leader election
  - Randomized protocol to select a unique leader in a symmetric ring of computers.

Experimental setup:
- Time limit: 2 hours
- Memory limit: 4 GB
- Solver: Gurobi 6
Some results for DTMCs

| Model      | |S| | |E_M| | |T| | |λ| | |S_MCS| | |E_MCS| | |Time |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| crowds2-3  | 183 | 243 | 26 | 0.09 | 22 | 27 | 0.06 (0.11) |
| crowds2-4  | 356 | 476 | 85 | 0.09 | 22 | 27 | 0.30 (0.24) |
| crowds2-5  | 612 | 822 | 196 | 0.09 | 22 | 27 | 0.56 (0.24) |
| crowds3-3  | 396 | 576 | 37 | 0.09 | 37 | 51 | 0.38 (0.30) |
| crowds3-4  | 901 | 1321 | 153 | 0.09 | 37 | 51 | 0.89 (0.58) |
| crowds3-5  | 1772 | 2612 | 425 | 0.09 | 37 | 51 | 1.51 (0.87) |
| crowds5-4  | 3515 | 6035 | 346 | 0.09 | 72 | 123 | 12.51 (4.89) |
| crowds5-5  | 18817 | 32677 | 3710 | 0.09 | 72 | 123 | 100.26 (23.52) |
| crowds5-6  | 68740 | 120220 | 19488 | 0.09 | 72 | 123 | 1000.79 (145.84) |
| leader3-2  | 22 | 29 | 1 | 0.5 | 15 | 18 | 0.21 (0.13) |
| leader3-3  | 61 | 87 | 1 | 0.5 | 33 | 45 | 0.02 (0.06) |
| leader3-4  | 135 | 198 | 1 | 0.5 | 70 | 101 | 0.07 (0.09) |
| leader4-2  | 55 | 70 | 1 | 0.5 | 34 | 41 | 0.24 (0.17) |
| leader4-3  | 256 | 336 | 1 | 0.5 | 132 | 171 | 0.49 (0.37) |
| leader4-4  | 782 | 1037 | 1 | 0.5 | 395 | 522 | 1.88 (1.21) |
| leader4-5  | 1889 | 2513 | 1 | 0.5 | 946 | 1257 | 4.06 (2.80) |
| leader4-6  | 3902 | 5197 | 1 | 0.5 | 1953 | 2600 | 8.70 (5.92) |
MILP formulation for MDPs

\[ \begin{align*}
\text{minimize} & \quad - \frac{1}{2} p_{s, \text{init}} + \sum_{s \in S} x_s \\
\text{such that} & \quad p_{s, \text{init}} > \lambda \\
& \quad x_s = p_{s, \text{non-target}} \\
& \quad p_{s, \text{action } a} \leq (1 - \sigma_{s, a}) + \sum_{s' \in S} \mathcal{P}(s, a, s') \cdot p_{s', \text{non-target}} 
\end{align*} \]
MILP formulation for MDPs

- $\sigma_{s,a} \in [0, 1] \subseteq \mathbb{Z}$: encoding of the scheduler
MILP formulation for MDPs

- \( \sigma_{s,a} \in [0,1] \subseteq \mathbb{Z} \): encoding of the scheduler

<table>
<thead>
<tr>
<th><strong>minimize</strong></th>
<th>(-\frac{1}{2}p_{s_{\text{init}}} + \sum_{s \in S} x_s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>such that</strong></td>
<td>(p_{s_{\text{init}}} &gt; \lambda)</td>
</tr>
<tr>
<td><strong>targets</strong>:</td>
<td>(x_s = p_s)</td>
</tr>
<tr>
<td><strong>non-target s</strong>:</td>
<td>(p_s \leq x_s) where (x_s = \sum_{a \in A} \sigma_{s,a})</td>
</tr>
<tr>
<td><strong>non-target s, action a</strong>:</td>
<td>(p_s \leq (1 - \sigma_{s,a}) + \sum_{s' \in S} P(s,a,s') \cdot p_{s'})</td>
</tr>
</tbody>
</table>
MILP formulation for MDPs: Problematic states

- $\sigma_{s,a} \in [0, 1] \subseteq \mathbb{Z}$: encoding of the scheduler
\( \sigma_{s,a} \in [0, 1] \subseteq \mathbb{Z} \): encoding of the scheduler

- \( x_{s_0} = 1 \)
- \( p_{s_0} = 1 \)
- \( \sigma_{s_0,a} = 0 \)
- \( \sigma_{s_0,b} = 1 \)

- \( x_{s_1} = 1 \)
- \( p_{s_1} = 1 \)
- \( x_{s_2} = 1 \)
- \( p_{s_2} = 1 \)
MILP formulation for MDPs

- \( \sigma_{s,a} \in [0, 1] \subseteq \mathbb{Z} \): encoding of the scheduler

<table>
<thead>
<tr>
<th>MILP Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>minimize</strong> &amp; (-\frac{1}{2}p_{s_{\text{init}}} + \sum_{s \in S} x_s) &amp; (1)</td>
</tr>
<tr>
<td><strong>such that</strong> &amp; (p_{s_{\text{init}}} &gt; \lambda) &amp; (2)</td>
</tr>
<tr>
<td><strong>targets</strong> &amp; (x_s = p_s) &amp; (3)</td>
</tr>
<tr>
<td><strong>non-target s</strong> &amp; (p_s \leq x_s) &amp; (x_s = \sum_{a \in A} \sigma_{s,a}) &amp; (4)</td>
</tr>
<tr>
<td><strong>non-target s, action a</strong> &amp; (p_s \leq (1 - \sigma_{s,a}) + \sum_{s' \in S} P(s, a, s') \cdot p_{s'}) &amp; (5)</td>
</tr>
<tr>
<td><strong>probl. s, s' \in \text{succ}(s, a)</strong> &amp; (2t_{s,s'} \leq x_s + x_{s'}) &amp; (6)</td>
</tr>
<tr>
<td>&amp; (r_s &lt; r_{s'} + (1 - t_{s,s'})) &amp; (7)</td>
</tr>
<tr>
<td>&amp; ((1 - x_s) + (1 - \sigma_{s,a}) + \sum_{s' \in \text{succ}(s,a)} t_{s,s'} \geq 1) &amp; (8)</td>
</tr>
</tbody>
</table>
Some results for MDPs

| Model       | $|S|$ | $|E|$ | prob. | $\lambda$ | $|S_{\text{min}}|$ | basic     | best opt. |
|-------------|------|------|-------|--------|----------------|----------|-----------|
| consensus-2-2 | 272  | 400  | 1     | 0.1    | 15            | – TO – (≥ 8) | 2 167     |
| consensus-2-4 | 528  | 784  | 1     | 0.1    | ≤ 35          | – TO – (≥ 9) | – TO – (≥ 12) |
| csma-2-2     | 1 038 | 1 054 | 1     | 0.1    | 195          | – TO – (≥ 184) | 638       |
| csma-2-4     | 7 958 | 7 988 | 1     | 0.1    | 410          | – TO – (≥ 408) | 1 342     |
| csma-2-6     | 66 718 | 66 788 | 1     | 0.1    | 415          | 2 364     | 2 364     |
| aleader-3    | 364  | 573  | 1     | 0.5    | ≤ 66         | – TO – (≥ 18) | – TO – (≥ 27) |
| aleader-4    | 3 172 | 6 252 | 1     | 0.5    | ≤ 215        | – TO – (≥ 10) | – TO – (≥ 10) |
Extensions of the MILP approach

- LTL properties both for DTMCs and MDPs
  - LTL $\rightarrow$ deterministic Rabin automaton (DRA)
  - DRA $\otimes$ DTMC/MDP $\rightarrow$ DTMC/MDP
  - Minimize projection onto the original state space
Extensions of the MILP approach

- LTL properties both for DTMCs and MDPs
  - LTL $\rightarrow$ deterministic Rabin automaton (DRA)
  - DRA $\otimes$ DTMC/MDP $\rightarrow$ DTMC/MDP
  - Minimize projection onto the original state space

- Expected reward properties
Extensions of the MILP approach

- LTL properties both for DTMCs and MDPs
  - LTL $\rightarrow$ deterministic Rabin automaton (DRA)
  - DRA $\otimes$ DTMC/MDP $\rightarrow$ DTMC/MDP
  - Minimize projection onto the original state space

- Expected reward properties

- High-level counterexamples (see last chapter)
Other approaches for computing small critical subsystems

Approaches:
- heuristic search (variant of A*) (Aljazzar/Leue)
- hierarchical abstraction of SCCs (Jansen et al.)
  - *symbolic methods using MTBDDs*
Symbolic Computation of Critical Subsystems
Aspects of probabilistic counterexamples:

- Counter-examples
- Representation
- Symbolic
- Explicit
- Executions
- Level
- Heuristic
- Optimal
- Optimality
- Reward
- Property
- (nested) PCTL
- LTL/ω-reg.
- Safety
- System Type
- DTMCs
- MDPs
- States
- Description
- Level
- Executions
Symbolic representation of DTMCs

Multi-terminal binary decision diagrams (MTBDDs):

- directed acyclic graphs with a root node
- terminal nodes: labeled with a real number
- internal nodes: two successors, high and low, labeled with a boolean variable

Each assignment of the variables induces a path in the MTBDD to a terminal node, whose label is the function value.

▶ functions $f : \{0, 1\}^n \rightarrow \mathbb{R}$
Example: DTMC

Encoding of the states:

<table>
<thead>
<tr>
<th>s_0</th>
<th>s_1</th>
<th>s_2</th>
<th>s_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>001</td>
<td>010</td>
<td>011</td>
</tr>
</tbody>
</table>
Example: BDD-encoding
MTBDD-based representation

- often (not always) much smaller than explicit representations
- efficient algorithms for (point-wise) addition, multiplication, matrix-multiplication … available
- in practise MTBDDs allow for representing very large systems
Counterexample computation using MTBDDs

Idea

- Start with the states of a most probable path from the initial to a target state
- Extend the system with further paths / path fragments until it becomes a counterexample

- **Global search**: all paths go from initial to target states
- **Fragment search**: paths start and end at an arbitrary state of the subsystem and contain at least one new state
Example

Global search:
Example

Global search:

\[ \xrightarrow{0.5} s_0 \xrightarrow{0.5} s_1 \xrightarrow{0.5} s_3 \]
Example

Global search:

- $s_0 \xrightarrow{0.5} s_1 \xrightarrow{0.5} s_3 \xrightarrow{1} 1$
- $s_0 \xrightarrow{0.5} s_1 \xrightarrow{0.5} s_2 \xrightarrow{0.5} s_1 \xrightarrow{0.5} s_3 \xrightarrow{1} 1$
Example

Local search:
Example

Local search:

\[ s_0 \xrightarrow{0.5} s_1 \xrightarrow{0.5} s_3 \xrightarrow{1} \]
Example

Local search:

\[
\begin{align*}
& s_0 \xrightarrow{0.5} s_1 \xrightarrow{0.5} s_3 \xrightarrow{1} s_3 \xleftarrow{1} s_1 \\
& s_1 \xrightarrow{0.5} s_2 \xrightarrow{0.5} s_1
\end{align*}
\]
Example: Result

Resulting subsystem:
The basic algorithm

OBDD states, newStates := ∅
MTBDD subsys := ∅

while modelCheck(subsys, T) ≤ λ do
    newStates := findNextPath(dtmc, Subsys);
    Subsys := Subsys ∪ newStates
end while
return Subsys
Finding paths

Use a symbolic version of **Dijkstra’s shortest path algorithm** to find a most probable path to a target state (Siegle et al.).

- FloodingDijkstra(transitions, start set, target set)

![Diagram showing the symbolic Dijkstra's algorithm with states and transition probabilities.](Diagram.png)
Global search

Extend the subsystem with paths from the initial to a target state

- FloodingDijkstra(transitions, init, targets)

How to exclude already found paths?
Example: Global search

First path:

\[s_0 \xrightarrow{0.5} s_1 \xrightarrow{0.5} s_3 \xleftarrow{1}\]
Example: Global search

Exclude all found transitions by doubling the DTMC:

Shortest path in the new graph is shortest path in the old graph containing at least one new state.
procedure LocalSearch(MTBDD trans, BDD init, BDD targets, BDD subsys)
    if subsystem = ∅ then
        return FloodingDijkstra(trans, init, targets);
    else
        subsystemStates = toStateBDD(subsystem);
        return FloodingDijkstra(trans \ subsystem, subsystemStates, subsystemStates);
    end if
end procedure
Results

- Largest instance: crowds-20-30 with $\approx 10^{16}$ states
  - $\approx 3000$ seconds
  - 873 MB memory
  - subsystem with 76 007 states.
- Subsystem size typically not far from minimum.
- Global search slightly faster, fragment search yields slightly smaller subsystems.
- currently restricted to safety and expected reward properties of DTMCs.
High-level counterexamples
Aspects of probabilistic counterexamples:

- Counter-examples
- Representation
  - Symbolic
  - Explicit
- Executions
- Level
- Optimality
  - Heuristic
  - Optimal
- System Type
  - DTMCs
  - MDPs
- Property
  - Safety
  - (nested) PCTL
  - LTL/ω-reg.
- Reward
- Optimal
- Heuristic
- Level
- Executions
- States
- Description
module coin
    f: bool init 0;
    c: bool init 0;
    [flip] ¬f → 0.5 : (f' = 1) & (c' = 1) + 0.5 : (f' = 1) & (c' = 0);
    [reset] f ∧ ¬c → 1 : (f' = 0);
    [proc] f → 0.99 : (f' = 1) + 0.01 : (c' = 1);
endmodule

module processor
    p: bool init 0;
    [proc] ¬p → 1 : (p' = 1);
    [loop] p → 1 : (p' = 1);
    [reset] true → 1 : (p' = 0)
endmodule
The induced MDP

\[ \mathcal{M} \not\models \mathcal{P}_{\leq 0.5} (\Diamond (f = 1 \land c = 1 \land p = 1)) \]
Counterexamples for PRISM models

Goal:

- Compute a minimal subset of the commands such that the induced system is already erroneous (minimal critical command set)
Counterexamples for PRISM models

Goal:

- Compute a minimal subset of the commands such that the induced system is already erroneous (minimal critical command set)

```
module coin
    f: bool init 0;
    c: bool init 0;
    [flip] ¬f → 0.5 : (f' = 1) & (c' = 1) + 0.5 : (f' = 1) & (c' = 0);
    [reset] f \land ¬c → 1 : (f' = 0);
    [proc] f → 0.99 : (f' = 1) + 0.01 : (c' = 1);
endmodule

module processor
    p: bool init 0;
    [proc] ¬p → 1 : (p' = 1);
    [loop] p → 1 : (p' = 1);
    [reset] true → 1 : (p' = 0)
endmodule
```

\[ \mathcal{M} \not\equiv \mathcal{P}_{\leq 0.5}(\Diamond (f = 1 \land c = 1 \land p = 1)) \]
The induced MDP

\[ M \not\models P_{\leq 0.5}(\Diamond (f = 1 \land c = 1 \land p = 1)) \]
Compose the modules of the PRISM program
Generate the corresponding MDP
Label all transitions with the command(s) they are created from
Compute a minimal critical labeling:
  - SMT + binary search
  - Mixed integer linear programming (QEST’13)
  - MAXSAT
Composition and state space generation

module coin
  f: bool init 0;
  c: bool init 0;
  c₁:  [flip] ¬f → 0.5 : (f' = 1) & (c' = 1) + 0.5 : (f' = 1) & (c' = 0);
  c₂:  [reset] f ∧ ¬c → 1 : (f' = 0);
  c₃:  [proc] f → 0.99 : (f' = 1) + 0.01 : (c' = 1);
endmodule

module processor
  p: bool init 0;
  c₄:  [proc] ¬p → 1 : (p' = 1);
  c₅:  [loop] p → 1 : (p' = 1);
  c₆:  [reset] true → 1 : (p' = 0)
endmodule

↓

module coin||processor
  f: bool init 0;
  c: bool init 0;
  p: bool init 0;
  c₁:  [flip] ¬f → 0.5 : (f' = 1) & (c' = 1) + 0.5 : (f' = 1) & (c' = 0);
  c₂, c₆:  [reset] f ∧ ¬c → 1 : (f' = 0) & (p' = 0);
  c₃, c₄:  [proc] f ∧ ¬p → 0.99 : (f' = 1) & (p' = 1) + 0.01 : (c' = 1) & (p' = 1);
  c₅:  [loop] p → 1 : (p' = 1);
endmodule
Composition and state space generation

module coin || processor
    f: bool init 0;
    c: bool init 0;
    p: bool init 0;

    c_1:
    [flip] ¬f → 0.5 : (f' = 1) & (c' = 1) + 0.5 : (f' = 1) & (c' = 0);

    c_2, c_6:
    [reset] f ∧ ¬c → 1 : (f' = 0) & (p' = 0);

    c_3, c_4:
    [proc] f ∧ ¬p → 0.99 : (f' = 1) & (p' = 1) + 0.01 : (c' = 1) & (p' = 1);

    c_5:
    [loop] p → 1 : (p' = 1);

endmodule
Idea: MAXSAT approach

\[ C = \text{MinSat}(\Phi_C, \Phi_P) \]

compute \( C^* \) > \( \lambda \)

\[ \text{Pr}_{\mathcal{A}|_C}(\diamond T) \leq \lambda \]

add constraints to \( \Phi_P \)

model \( \langle \Psi \rangle \)

analysis of \( \mathcal{A}|_C \)
MAXSAT

Definition: MAXSAT

Given two sets of clauses:

- \( \varphi_h \) (hard constraints)
- \( \varphi_s \) (soft constraints)

find an assignment which satisfies all hard constraints and as many soft constraints as possible.

Several solvers available: MaxAntom, Z3, …
Initial constraint system

- **Guaranteed commands:**
  Commands occurring on each path from $s_{\text{init}}$ to $T$ are contained in $C^*$.

- **Proper synchronization:**
  Each synchronizing command $c \in C^*$ needs a matching partner from each module synchronizing with $c$.

- **Predecessors and successors:**
  At least one state $s \in S \setminus T$, in which $c \in C^*$ is enabled needs a successor state with an activated command.

  At least one state $s \in S \setminus \{s_{\text{init}}\}$, in which $c \in C^*$ is enabled needs a predecessor state with an activated command leading to $s$. 
Example: $T$ unreachable from $s_{\text{init}}$

- Some command appearing on an arbitrary cut between $A$ and $B$
  must be contained in the subsystem
## Evaluation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>coin(2, 2)</td>
<td>272</td>
<td>492</td>
<td>0.4 / 0.56</td>
<td>10 (4)</td>
<td>9</td>
<td>0.08</td>
<td>0.02</td>
<td>54%</td>
</tr>
<tr>
<td>coin(4, 4)</td>
<td>43136</td>
<td>144352</td>
<td>0.4 / 0.54</td>
<td>20 (8)</td>
<td>17</td>
<td>1876</td>
<td>0.07</td>
<td>50%</td>
</tr>
<tr>
<td>coin(4, 6)</td>
<td>63616</td>
<td>213472</td>
<td>0.4 / 0.53</td>
<td>20 (8)</td>
<td>17</td>
<td>6231</td>
<td>0.09</td>
<td>50%</td>
</tr>
<tr>
<td>coin(6, 2)</td>
<td>1258240</td>
<td>6236736</td>
<td>0.4 / 0.59</td>
<td>30 (12)</td>
<td>–</td>
<td>TO</td>
<td>&gt; 1.54</td>
<td>–</td>
</tr>
<tr>
<td>csma(2, 4)</td>
<td>7958</td>
<td>10594</td>
<td>0.5 / 0.999</td>
<td>38 (21)</td>
<td>36</td>
<td>2.26</td>
<td>0.04</td>
<td>0.09%</td>
</tr>
<tr>
<td>csma(4, 2)</td>
<td>761962</td>
<td>1327068</td>
<td>0.4 / 0.78</td>
<td>68 (22)</td>
<td>53</td>
<td>18272</td>
<td>0.92</td>
<td>3.9E-9%</td>
</tr>
<tr>
<td>fw(1)</td>
<td>1743</td>
<td>2199</td>
<td>0.5 / 1</td>
<td>64 (6)</td>
<td>24</td>
<td>16.14</td>
<td>0.05</td>
<td>1.4E-10%</td>
</tr>
<tr>
<td>fw(10)</td>
<td>17190</td>
<td>29366</td>
<td>0.5 / 1</td>
<td>64 (6)</td>
<td>24</td>
<td>90.47</td>
<td>0.07</td>
<td>1.4E-10%</td>
</tr>
<tr>
<td>fw(36)</td>
<td>212268</td>
<td>481792</td>
<td>0.5 / 1</td>
<td>64 (6)</td>
<td>24</td>
<td>1542</td>
<td>0.34</td>
<td>1.4E-10%</td>
</tr>
<tr>
<td>wlan(0, 2)</td>
<td>6063</td>
<td>10619</td>
<td>0.1 / 0.184</td>
<td>42 (22)</td>
<td>33</td>
<td>1.6</td>
<td>0.03</td>
<td>0.02%</td>
</tr>
<tr>
<td>wlan(2, 4)</td>
<td>59416</td>
<td>119957</td>
<td>4E-4 / 7.9E-4</td>
<td>48 (26)</td>
<td>39</td>
<td>50.27</td>
<td>0.07</td>
<td>0.01%</td>
</tr>
<tr>
<td>wlan(6, 6)</td>
<td>5007670</td>
<td>11475920</td>
<td>1E-7 / 2.2E-7</td>
<td>52 (30)</td>
<td>43</td>
<td>5035</td>
<td>3.86</td>
<td>0.01%</td>
</tr>
</tbody>
</table>
Different kinds of counterexamples available
- path-based counterexamples
- critical subsystems
- critical command sets

Both optimal and heuristic computation methods

Symbolic methods scale relatively well to large DTMCs
Open Research Questions

So far, there are few concrete applications of probabilistic cex:

- Probabilistic CEGAR (Hermanns et al., CAV’08; Chadha/Viswanathan, TOCL 2010)
- Fault trees from counterexamples (Fischer-Leitner/Leue, IJCCBS 2013)

Open challenges:

- Demonstrate usefulness for debugging
- Application of subsystems and high-level cex in abstraction refinement
- Counterexamples for continuous-time probabilistic models
- Application for model repair.
Some References

Overview paper on cex:


Research papers:


Aspects of probabilistic counterexamples:

- Executive
- Symbolic
- Explicit
- States
- Description
- Level
- Optimality
- Heuristic
- Optimal
- Reward
- Property
- (nested) PCTL
- LTL/ω-reg.
- Safety
- System Type
- DTMCs
- MDPs
- LTL/ω-reg.
- PCTL
- Safety