Hierarchical Counterexamples for DTMCs

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

Software Modeling and Verification, Theory of Hybrid Systems
RWTH Aachen University, Germany

Computer Architecture
University of Freiburg, Germany

March 27, 2011

ROCKS Workshop
Counterexamples

Model Checking
- Shows correctness of a system
- Reveals defectiveness of a system

Counterexamples for LTL properties
- Are delivered by Model Checking for defective systems
- Consist of single traces through a system

Counterexamples in the probabilistic setting
- Are not computed during Model Checking
- Consist of (large or infinite) sets of paths
Some state-of-the-art methods

- Search for paths in order of their probability \((\text{Damman, Han, and Katoen 2008})\)
- Find minimal counterexamples
- Use the abstraction of SCCs \((\text{Andrés, D’Argenio and van Rossum, 2008})\)

Counterexamples are represented

- By enumeration of the paths
- By regular expressions
Hierarchical Counterexample Generation

Method

- SCC-based Model Checking [QEST’10]
- If property was falsified:
  - Search on abstract system
  - Hierarchical concretization
  - Two different search approaches

Representation

- Critical subsystem

Advantages

- Compact representation $\rightsquigarrow$ Usability
- Abstract counterexamples $\rightsquigarrow$ Treatment of large systems
- Hierarchical approach $\rightsquigarrow$ Omission of system parts
Contents

1 Motivation

2 SCC-based Model Checking

3 Counterexample Generation

4 Implementation and Case Studies

5 Conclusion & Future Work
Problem
- Model Checking DTMCs against unbounded reachability properties
- Target nodes are absorbing.

Idea
- Reduce each (nested) SCC to an abstract node whose outgoing edges carry the whole probability mass.

Recursive algorithm
- Bottom-Up computing starting with “minimal SCCs”
- Exploiting specific properties of Markov Chains
SCC-based Model Checking - Example

Abstraction of SCC 1.2.1
SCC-based Model Checking - Example

Abstraction of SCC 1.2
SCC-based Model Checking - Example

Abstraction of SCC 1.1
Abstraction of SCC 1
SCC-based Model Checking - Example

Path reduction
DTMC $M = (S, I, P, L)$

$S' \subseteq S$, $Out^M(S')$
Formalisms

DTMC $M = (S, I, P, L)$

$S' \subseteq S$, $Out^M(S')$

*Induced DTMC:*

$M' = DTMC(S', M)$
Formalisms

DTMC $M = (S, I, P, L)$

$S' \subseteq S$, $\text{Out}^M(S')$

Induced DTMC:
$M' = \text{DTMC}(S', M)$

Pairs $(M', M_{\text{abs}})$ are saved during procedure.
Formalisms

DTMC $M = (S, I, P, L)$

$S' \subseteq S$, $Out^M(S')$

**Induced DTMC:**

$M' = DTMC(S', M)$

**Abstraction of $M'$:**

$Abs(M') = M_{abs}$
DTMC $M = (S, I, P, L)$

$S' \subseteq S$, $Out^M(S')$

*Induced DTMC:*

$M' = DTMC(S', M)$

*Abstraction of $M'$:*

$Abs(M') = M_{abs}$
DTMC $M = (S, I, P, L)$

$S' \subseteq S$, $Out^M(S')$

**Induced DTMC:**

$M' = DTMC(S', M)$

**Abstraction of $M'$:**

$Abs(M') = M_{abs}$

**Substitution:**

$M[M_{abs}/M']$
Formalisms

DTMC $M = (S, I, P, L)$

$S' \subseteq S$, $Out^M(S')$

**Induced DTMC:**

$M' = DTMC(S', M)$

**Abstraction of $M'$:**

$Abs(M') = M_{abs}$

**Substitution:**

$M[M_{abs}/M']$
Formalisms

DTMC $M = (S, I, P, L)$

$S' \subseteq S$, $Out^M(S')$

**Induced DTMC:**

$M' = DTMC(S', M)$

**Abstraction of $M'$:**

$Abs(M') = M_{abs}$

**Substitution:**

$M[M_{abs}/M']$

**Concretization:**

$M[M'/M_{abs}]$
DTMC $M = (S, I, P, L)$

$S' \subseteq S$, $Out^M(S')$

*Induced DTMC:*

$M' = DTMC(S', M)$

*Abstraction of $M'$:*

$Abs(M') = M_{abs}$

*Substitution:*

$M[M_{abs}/M']$

*Concretization:*

$M[M'/M_{abs}]$
Formalisms

DTMC $M = (S, I, P, L)$

$S' \subseteq S$, $Out^M(S')$

*Induced DTMC:* $M' = DTMC(S', M)$

*Abstraction of $M'$:* $\text{Abs}(M') = M_{abs}$

*Substitution:* $M[M_{abs}/M']$

*Concretization:* $M[M'/M_{abs}]$

Pairs $(M', M_{abs})$ are saved during procedure.
Counterexample Generation - Overview

- **Hierarchical concretization** of counterexamples
- Search for paths $\leadsto$ selection of edges
  - Global Search
  - Local Search
- Representation by a critical subsystem
Critical Subsystem

- Subsystem of a DTMC induced by a set of selected edges.
- Set of all paths shall violate a certain PCTL property.

DTMC $M = (S, I, P, L)$,
selection of edges $m \subseteq S \times S$
Critical Subsystem

- Subsystem of a DTMC induced by a set of selected edges.
- Set of all paths shall violate a certain PCTL property.

DTMC $M = (S, I, P, L)$,

selection of edges $m \subseteq S \times S$

$\text{closure}^M(m)$
Hierarchical Algorithm - Overview

- **Input**: Abstract DTMC $M$
- **Concretize** one or more states of $M$ (using heuristics)
- Find a critical subsystem $M_{ce}$

- Global Search

- Local Search
Hierarchical Algorithm - Overview

- **Input:** Abstract DTMC $M$
- **Concretize** one or more states of $M$ (using heuristics)
- **Find a critical subsystem** $M_{ce}$
  - Find a certain path $\pi$
  - Edges along $\pi$ are selected $\sim m$
  - Compute $\text{closure}_M(m)$
  - Search for more paths until $\text{closure}_M(m)$ has enough probability mass

- **Global Search**

- **Local Search**
Hierarchical Algorithm - Overview

- **Input:** Abstract DTMC $M$
- **Concretize** one or more states of $M$ (using heuristics)
- Find a critical subsystem $M_{ce}$
  - Find a certain path $\pi$
  - Edges along $\pi$ are selected $\sim m$
  - Compute $\text{closure}_M(m)$
  - Search for more paths until $\text{closure}_M(m)$ has enough probability mass

- **Global Search**
  - Searches for the most probable paths through the current critical subsystem

- **Local Search**
Hierarchical Algorithm - Overview

- **Input**: Abstract DTMC $M$
- **Concretize** one or more states of $M$ (using heuristics)
- Find a **critical subsystem** $M_{ce}$
  - Find a certain path $\pi$
  - Edges along $\pi$ are selected $\sim m$
  - Compute $\text{closure}_M(m)$
  - Search for more paths until $\text{closure}_M(m)$ has enough probability mass

- **Global Search**
  - Searches for the **most probable paths** through the current critical subsystem

- **Local Search**
  - Searches for **most probable path fragments** that connect already selected paths
  - Search is restricted to the recently concretized parts
During one search iteration

- Selection $m_{\text{max}}$ induces the maximum probability mass
- Selection $m_{\text{min}}$ induces the minimum probability mass

- $m_{\text{min}}$ is extended by edges of $m_{\text{max}}$
Global Search - Example

Input:
Abstract DTMC, Target state 5, Bound $\mathbb{P}_{<0.3}(\Box 5)$
Initial most probable path
Global Search - Example

Concretize 0
Global Search - Example

Selection $m_{\text{max}}$

Selection $m_{\text{min}}$
Global Search - Example

Search for most probable paths

Current critical subsystem
Global Search - Example

Concretize 1

Current critical subsystem

Graph with nodes 0, 1, 5, and 9 showing transitions and labels.
Global Search - Example

Selection $m_{\text{max}}$

Selection $m_{\text{min}}$

Nils Jansen, Jens Katelaan - Hierarchical Counterexamples
Global Search - Example

Search for most probable paths

Current critical subsystem
Global Search - Example

Concretize 2

Current critical subsystem
Global Search - Example

Selection $m_{\text{max}}$

Selection $m_{\text{min}}$
Global Search - Example

Search for most probable paths

Probability mass: 0.125
Global Search - Example

Search for most probable paths

Probability mass: 0.2
Search for most probable paths

Probability mass: $\frac{1}{3}$
- Search for most probable path fragments
- "Blow up" paths

Local Search - Intuition

Connect red states along most probable paths

Critical subsystem induces probability mass of 0.7
- Search for most probable path fragments
- "Blow up" paths

Connect red states along most probable paths
- Search for most probable path fragments
- "Blow up" paths

Connect red states along most probable paths
Search for most probable path fragments

"Blow up" paths

Connect red states along most probable paths
- Search for most probable path fragments
- "Blow up" paths

Connect red states along most probable paths

Critical subsystem induces probability mass of 0.7
Contents

1 Motivation

2 SCC-based Model Checking

3 Counterexample Generation

4 Implementation and Case Studies

5 Conclusion & Future Work
Prototype implementation in C++

Two general approaches:
- Hierarchical concretization of counterexamples
- Counterexample search on the concrete DTMC

Two different search algorithms:
- Global Search
- Local Search

Closure computations on top of a search for the most probable paths

Result is a (possibly abstract) subsystem that induces violation of the given property
...are boring
Counterexample generation for case studies

- Synchronous leader election protocol (Itai, Rodeh 1990)
- Crowds protocol (Reiter, Rubin 1998)
The Crowds Protocol

- Protocol for **anonymous communication in networks**
- **n** users divided into good and bad members
- Random forwarding of messages to destination or other member
- A run of **r** message deliveries is modelled
- Our models are parameterized by **n** and **r**
- Fixed good-to-bad ratio, fixed forwarding probabilities
- Model checking: Probability that a member is identified (i.e. not anonymous)?
Show applicability and advantages of
- Representation as critical subsystem
- Both search approaches
- The hierarchical approach
- Heuristics
Critical subsystems

States: 3515, Transitions: 6035 (Crowds 5 / 4)

Model checking result $\mathbb{P}(M) : 0.2346$

<table>
<thead>
<tr>
<th>Probability bound $p$</th>
<th>0.15</th>
<th>0.15</th>
<th>0.15</th>
</tr>
</thead>
<tbody>
<tr>
<td>counterexample as</td>
<td>set of paths</td>
<td>subsystem</td>
<td>subsystem</td>
</tr>
<tr>
<td>algorithm</td>
<td>path search</td>
<td>global</td>
<td>local</td>
</tr>
<tr>
<td># computed paths</td>
<td>488644</td>
<td>958</td>
<td>98</td>
</tr>
<tr>
<td># computed closures</td>
<td>-</td>
<td>182</td>
<td>98</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>1071</td>
<td>632</td>
<td>171</td>
</tr>
</tbody>
</table>

- For $p \rightarrow \mathbb{P}(M)$ the size of counterexample path sets increases rapidly
- The critical subsystem automatically includes all loop iterations etc $\implies$ many improbable paths are omitted
Critical subsystems

States: 3515, Transitions: 6035 (Crowds 5 / 4)

Model checking result $\mathbb{P}(M) : 0.2346$

<table>
<thead>
<tr>
<th>Probability bound $p$</th>
<th>0.15</th>
<th>0.15</th>
<th>0.15</th>
</tr>
</thead>
<tbody>
<tr>
<td>counterexample as</td>
<td>set of paths</td>
<td>subsystem</td>
<td>subsystem</td>
</tr>
<tr>
<td>algorithm</td>
<td>path search</td>
<td>global</td>
<td>local</td>
</tr>
<tr>
<td># computed paths</td>
<td>488644</td>
<td>958</td>
<td>98</td>
</tr>
<tr>
<td># computed closures</td>
<td>-</td>
<td>182</td>
<td>98</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>1071</td>
<td>632</td>
<td>171</td>
</tr>
</tbody>
</table>

- For $p \rightarrow \mathbb{P}(M)$ the size of counterexample path sets increases rapidly
- The critical subsystem automatically includes all loop iterations etc
  $\Rightarrow$ many improbable paths are omitted
Critical subsystems

States: 3515, Transitions: 6035 (Crowds 5 / 4)

Model checking result $\mathbb{P}(M) : 0.2346$

<table>
<thead>
<tr>
<th>Probability bound $p$</th>
<th>0.15</th>
<th>0.15</th>
<th>0.15</th>
</tr>
</thead>
<tbody>
<tr>
<td>counterexample as</td>
<td>set of paths</td>
<td>subsystem</td>
<td>subsystem</td>
</tr>
<tr>
<td>algorithm</td>
<td>path search</td>
<td>global</td>
<td>local</td>
</tr>
<tr>
<td># computed paths</td>
<td>488644</td>
<td>958</td>
<td>98</td>
</tr>
<tr>
<td># computed closures</td>
<td>-</td>
<td>182</td>
<td>98</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>1071</td>
<td>632</td>
<td>171</td>
</tr>
</tbody>
</table>

- For $p \rightarrow \mathbb{P}(M)$ the size of counterexample path sets increases rapidly
- The critical subsystem automatically includes all loop iterations etc $\Rightarrow$ many improbable paths are omitted
Critical subsystems

States: 3515, Transitions: 6035 (Crowds 5 / 4)

Model checking result $\mathbb{P}(M) : 0.2346$

<table>
<thead>
<tr>
<th>Probability bound $p$</th>
<th>0.15</th>
<th>0.15</th>
<th>0.15</th>
<th>0.23</th>
</tr>
</thead>
<tbody>
<tr>
<td>counterexample as</td>
<td>set of paths</td>
<td>subsystem</td>
<td>subsystem</td>
<td>subsystem</td>
</tr>
<tr>
<td>algorithm</td>
<td>path search</td>
<td>global</td>
<td>local</td>
<td>global</td>
</tr>
<tr>
<td># computed paths</td>
<td>488644</td>
<td>958</td>
<td>98</td>
<td>151639</td>
</tr>
<tr>
<td># computed closures</td>
<td>-</td>
<td>182</td>
<td>98</td>
<td>623</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>1071</td>
<td>632</td>
<td>171</td>
<td>1071</td>
</tr>
</tbody>
</table>

- For $p \rightarrow \mathbb{P}(M)$ the size of counterexample path sets increases rapidly
- The critical subsystem automatically includes all loop iterations etc \[\implies\] many improbable paths are omitted
Critical subsystems

States: 3515, Transitions: 6035 (Crowds 5 / 4)

Model checking result $P(M) : 0.2346$

<table>
<thead>
<tr>
<th>Probability bound $p$</th>
<th>0.15</th>
<th>0.15</th>
<th>0.15</th>
<th>0.23</th>
</tr>
</thead>
<tbody>
<tr>
<td>counterexample as</td>
<td>set of paths</td>
<td>subsystem</td>
<td>subsystem</td>
<td>subsystem</td>
</tr>
<tr>
<td>algorithm</td>
<td>path search</td>
<td>global</td>
<td>local</td>
<td>global</td>
</tr>
<tr>
<td># computed paths</td>
<td>488644</td>
<td>958</td>
<td>98</td>
<td>151639</td>
</tr>
<tr>
<td># computed closures</td>
<td>-</td>
<td>182</td>
<td>98</td>
<td>623</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>1071</td>
<td>632</td>
<td>171</td>
<td>1071</td>
</tr>
</tbody>
</table>

- For $p \rightarrow P(M)$ the size of counterexample path sets increases rapidly
- The critical subsystem automatically includes all loop iterations etc
  $\implies$ many improbable paths are omitted
(Closure-based) Global and local search

States: 18817, Transitions: 32677 (Crowds 5 / 6)

Model checking result $\mathbb{P}(M) : 0.4270$

<table>
<thead>
<tr>
<th>Probability bound $p$</th>
<th>0.2</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td># paths global</td>
<td>3007</td>
<td>56657</td>
</tr>
<tr>
<td># closures global</td>
<td>302</td>
<td>767</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>663</td>
<td>2047</td>
</tr>
<tr>
<td>$\mathbb{P}(M_{ce})$</td>
<td>0.2002</td>
<td>0.2500</td>
</tr>
<tr>
<td># paths/clos. local</td>
<td>202</td>
<td>798</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>326</td>
<td>1439</td>
</tr>
<tr>
<td>$\mathbb{P}(M_{ce})$</td>
<td>0.2001</td>
<td>0.2508</td>
</tr>
</tbody>
</table>

- **Global search** finds many unnecessary paths (loops etc.)
- **Local search** finds only significant path fragments, but is expensive
(Closure-based) Global and local search

States: 18817, Transitions: 32677 (Crowds 5 / 6)

Model checking result $P(M) : 0.4270$

<table>
<thead>
<tr>
<th>Probability bound $p$</th>
<th>0.2</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td># paths global</td>
<td>3007</td>
<td>56657</td>
</tr>
<tr>
<td># closures global</td>
<td>302</td>
<td>767</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>663</td>
<td>2047</td>
</tr>
<tr>
<td>$P(M_{ce})$</td>
<td>0.2002</td>
<td>0.2500</td>
</tr>
<tr>
<td># paths/clos. local</td>
<td>202</td>
<td>798</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>326</td>
<td>1439</td>
</tr>
<tr>
<td>$P(M_{ce})$</td>
<td>0.2001</td>
<td>0.2508</td>
</tr>
</tbody>
</table>

- **Global search** finds many unnecessary paths (loops etc.)
- **Local search** finds only significant path fragments, but is expensive
(Closure-based) Global and local search

States: 18817, Transitions: 32677 (Crowds 5 / 6)

Model checking result $\mathbb{P}(M) : 0.4270$

<table>
<thead>
<tr>
<th>Probability bound $p$</th>
<th>0.2</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td># paths global</td>
<td>3007</td>
<td>56657</td>
</tr>
<tr>
<td># closures global</td>
<td>302</td>
<td>767</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>663</td>
<td>2047</td>
</tr>
<tr>
<td>$\mathbb{P}(M_{ce})$</td>
<td>0.2002</td>
<td>0.2500</td>
</tr>
<tr>
<td># paths/clos. local</td>
<td>202</td>
<td>798</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>326</td>
<td>1439</td>
</tr>
<tr>
<td>$\mathbb{P}(M_{ce})$</td>
<td>0.2001</td>
<td>0.2508</td>
</tr>
</tbody>
</table>

- **Global search** finds many unnecessary paths (loops etc.)
- **Local search** finds only significant path fragments, but is expensive
The hierarchical approach

States: 18817, Transitions: 32677 (Crowds 5 / 6)
Model checking result $\mathbb{P}(M) : 0.4270$
(Hierarchical) SCCs: 756

<table>
<thead>
<tr>
<th>Probability bound $p$</th>
<th>0.2</th>
<th>0.25</th>
<th>0.3</th>
<th>0.35</th>
</tr>
</thead>
<tbody>
<tr>
<td># refinement steps</td>
<td>13</td>
<td>21</td>
<td>23</td>
<td>35</td>
</tr>
<tr>
<td># paths global</td>
<td>13525</td>
<td>55770</td>
<td>205362</td>
<td>3641675</td>
</tr>
<tr>
<td># closures global</td>
<td>728</td>
<td>1729</td>
<td>2197</td>
<td>4944</td>
</tr>
<tr>
<td># concretized SCCs</td>
<td>37</td>
<td>88</td>
<td>105</td>
<td>224</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>457</td>
<td>1109</td>
<td>1363</td>
<td>3036</td>
</tr>
</tbody>
</table>

- Idea: Identify whole components of the system that form a counterexample
- The hierarchical approach does this by selecting / discarding SCCs
The hierarchical approach

States: 18817, Transitions: 32677 (Crowds 5 / 6)
Model checking result $\mathbb{P}(M) : 0.4270$
(Hierarchical) SCCs: 756

<table>
<thead>
<tr>
<th>Probability bound $p$</th>
<th>0.2</th>
<th>0.25</th>
<th>0.3</th>
<th>0.35</th>
</tr>
</thead>
<tbody>
<tr>
<td># refinement steps</td>
<td>13</td>
<td>21</td>
<td>23</td>
<td>35</td>
</tr>
<tr>
<td># paths global</td>
<td>13525</td>
<td>55770</td>
<td>205362</td>
<td>3641675</td>
</tr>
<tr>
<td># closures global</td>
<td>728</td>
<td>1729</td>
<td>2197</td>
<td>4944</td>
</tr>
<tr>
<td># concretized SCCs</td>
<td>37</td>
<td>88</td>
<td>105</td>
<td>224</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>457</td>
<td>1109</td>
<td>1363</td>
<td>3036</td>
</tr>
</tbody>
</table>

- Idea: Identify whole components of the system that form a counterexample
- The hierarchical approach does this by selecting / discarding SCCs
The hierarchical approach

States: 18817, Transitions: 32677 (Crowds 5 / 6)
Model checking result $\mathbb{P}(M) : 0.4270$
(Hierarchical) SCCs: 756

<table>
<thead>
<tr>
<th>Probabilitiy bound $p$</th>
<th>0.2</th>
<th>0.25</th>
<th>0.3</th>
<th>0.35</th>
</tr>
</thead>
<tbody>
<tr>
<td># refinement steps</td>
<td>13</td>
<td>21</td>
<td>23</td>
<td>35</td>
</tr>
<tr>
<td># paths global</td>
<td>13525</td>
<td>55770</td>
<td>205362</td>
<td>3641675</td>
</tr>
<tr>
<td># closures global</td>
<td>728</td>
<td>1729</td>
<td>2197</td>
<td>4944</td>
</tr>
<tr>
<td># concretized SCCs</td>
<td>37</td>
<td>88</td>
<td>105</td>
<td>224</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>457</td>
<td>1109</td>
<td>1363</td>
<td>3036</td>
</tr>
</tbody>
</table>

- Idea: Identify whole components of the system that form a counterexample
- The hierarchical approach does this by selecting / discarding SCCs
The hierarchical approach

States: 18817, Transitions: 32677 (Crowds 5 / 6)
Model checking result $\mathbb{P}(M) : 0.4270$
(Hierarchical) SCCs: 756

<table>
<thead>
<tr>
<th>Probability bound $p$</th>
<th>0.2</th>
<th>0.25</th>
<th>0.3</th>
<th>0.35</th>
</tr>
</thead>
<tbody>
<tr>
<td># refinement steps</td>
<td>13</td>
<td>21</td>
<td>23</td>
<td>35</td>
</tr>
<tr>
<td># paths global</td>
<td>13525</td>
<td>55770</td>
<td>205362</td>
<td>3641675</td>
</tr>
<tr>
<td># closures global</td>
<td>728</td>
<td>1729</td>
<td>2197</td>
<td>4944</td>
</tr>
<tr>
<td># concretized SCCs</td>
<td>37</td>
<td>88</td>
<td>105</td>
<td>224</td>
</tr>
<tr>
<td># states $M_{ce}$</td>
<td>457</td>
<td>1109</td>
<td>1363</td>
<td>3036</td>
</tr>
</tbody>
</table>

- Idea: Identify whole components of the system that form a counterexample
- The hierarchical approach does this by selecting / discarding SCCs
Heuristics

- How many abstract states should be concretized per step?
- In which order should abstract states be concretized?

⇝ **Heuristics:**

- Select abstract states with most probable edges
- Number of concretized states parametrized by the number of available states
Heuristics

States: 18817, Transitions: 32677, Bound p: 0.2 (Crowds 5 / 6)

<table>
<thead>
<tr>
<th>Search Type</th>
<th>global</th>
<th>global</th>
</tr>
</thead>
<tbody>
<tr>
<td>conc-Heur</td>
<td>single</td>
<td>single</td>
</tr>
<tr>
<td>choose-Heur</td>
<td>prob</td>
<td>None</td>
</tr>
<tr>
<td>#Nodes</td>
<td>458</td>
<td>457</td>
</tr>
<tr>
<td>#Refinement</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>#Paths</td>
<td>38379</td>
<td>594881</td>
</tr>
<tr>
<td>#Closures</td>
<td>728</td>
<td>729</td>
</tr>
</tbody>
</table>

Heuristics:
- $\sqrt{k}$: $\sqrt{k}$ out of $k$ possible SCCs are concretized in 1 step
- single: Only 1 SCC is concretized per refinement step.
- prob: SCCs are concretized in order w.r.t. their average output probabilities
Heuristics:

- $\sqrt{k}$: $\sqrt{k}$ out of $k$ possible SCCs are concretized in 1 step
- single: Only 1 SCC is concretized per refinement step.
- prob: SCCs are concretized in order w.r.t. their average output probabilities
Heuristics

States: 18817, Transitions: 32677, Bound p: 0.2 (Crowds 5 / 6)

<table>
<thead>
<tr>
<th>Search Type</th>
<th>global</th>
<th>global</th>
<th>global</th>
<th>global</th>
<th>local</th>
<th>local</th>
</tr>
</thead>
<tbody>
<tr>
<td>conc-Heur</td>
<td>single</td>
<td>single</td>
<td>√</td>
<td>√</td>
<td>single</td>
<td>√</td>
</tr>
<tr>
<td>choose-Heur</td>
<td>prob</td>
<td>None</td>
<td>prob</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>#Nodes</td>
<td>458</td>
<td>457</td>
<td>457</td>
<td>457</td>
<td>347</td>
<td>319</td>
</tr>
<tr>
<td>#Refinement</td>
<td>37</td>
<td>37</td>
<td>13</td>
<td>10</td>
<td>28</td>
<td>9</td>
</tr>
<tr>
<td>#Paths</td>
<td>38379</td>
<td>594881</td>
<td>13525</td>
<td>912455</td>
<td>545</td>
<td>496</td>
</tr>
<tr>
<td>#Closures</td>
<td>728</td>
<td>729</td>
<td>728</td>
<td>730</td>
<td>545</td>
<td>496</td>
</tr>
</tbody>
</table>

Heuristics:

- √: √\(k\) out of \(k\) possible SCCs are concretized in 1 step
- single: Only 1 SCC is concretized per refinement step.
- prob: SCCs are concretized in order w.r.t. their average output probabilities
GUI - In Progress
Contents

1 Motivation

2 SCC-based Model Checking

3 Counterexample Generation

4 Implementation and Case Studies

5 Conclusion & Future Work
Conclusion & Future Work

We did

We are working on

We will work on
Conclusion & Future Work

We did

- Model Checking for DTMCs with a resulting abstract system
- Counterexample Generation
  - both on abstract and concrete systems
  - with a very compact representation
  - with promising test results

We are working on

We will work on
Conclusion & Future Work

We did

- Model Checking for DTMCs with a resulting abstract system
- Counterexample Generation
  - both on abstract and concrete systems
  - with a very compact representation
  - with promising test results

We are working on

- a user interface
- an optimized implementation
- further heuristics

We will work on
Conclusion & Future Work

We did

- **Model Checking for DTMCs** with a resulting abstract system
- **Counterexample Generation**
  - both on abstract and concrete systems
  - with a very compact representation
  - with promising test results

We are working on

- a **user interface**
- an optimized implementation
- further **heuristics**

We will work on

- **Case studies:**
  - Retain information
  - Reduce size
- **Complexity analysis**

Thank you!