Optimal Counterexamples for Discrete-Time Markov Models

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Ralf Wimmer

Albert-Ludwigs-Universität Freiburg, Germany

Joint work with Nils Jansen, Erika Ábrahám, Joost Pieter Katoen, Bernd Becker

Outline

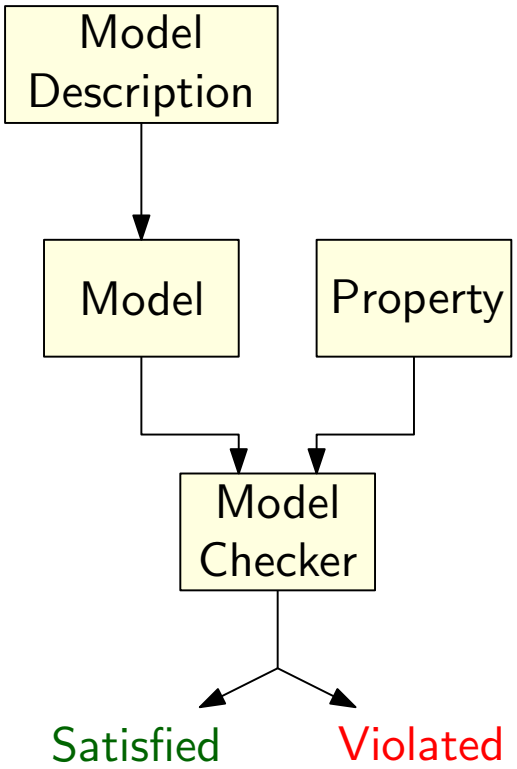
UNI FREIBURG

Overview on Probabilistic Model Checking

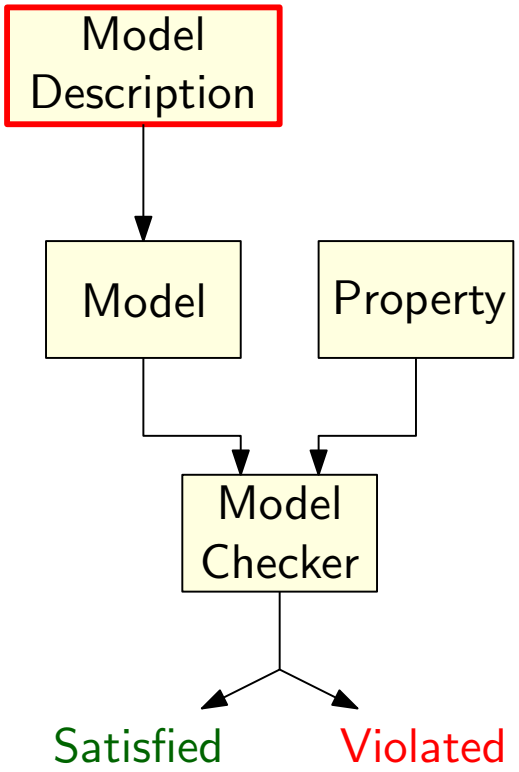
Counterexamples
- Path-based Counterexamples
- Minimal Critical Subsystems
- Minimal critical command sets

Conclusion and Future Work

Probabilistic Model Checking



Probabilistic Model Checking



Model Description:

Guarded command language

```
x,y: [0..5] init 0
module M1
```

$$[\alpha]\ (x+y \leq 2) \rightarrow 0.4 : x' = 4 \ + \ 0.6 : x' = y+1$$

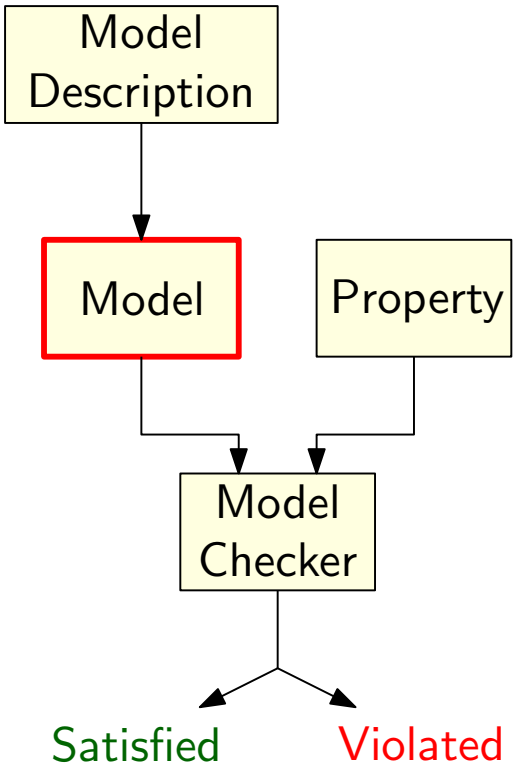
```
endmodule

module M2
```

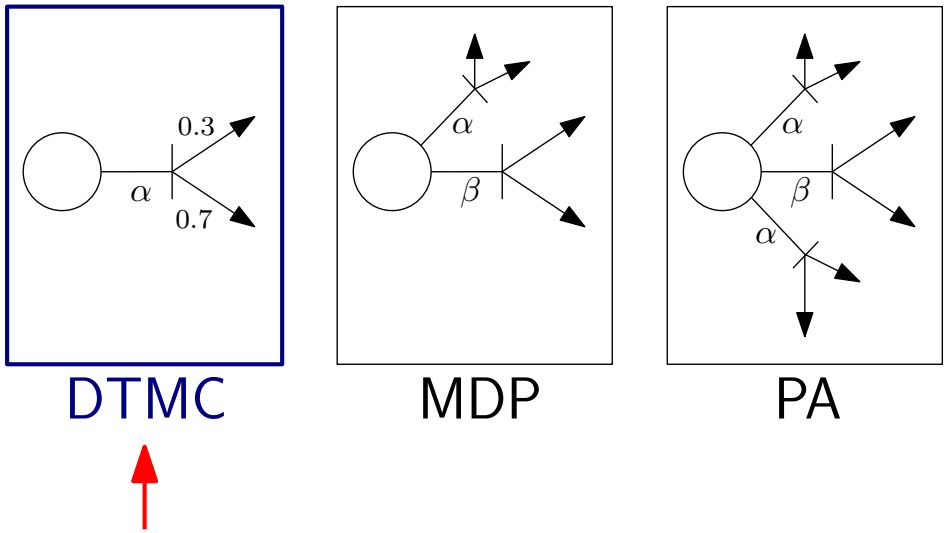
$$[\alpha]\ (x-y = 3) \rightarrow 0.1 : y' = x \ + \ 0.9 : y' = 2x$$

```
endmodule
```

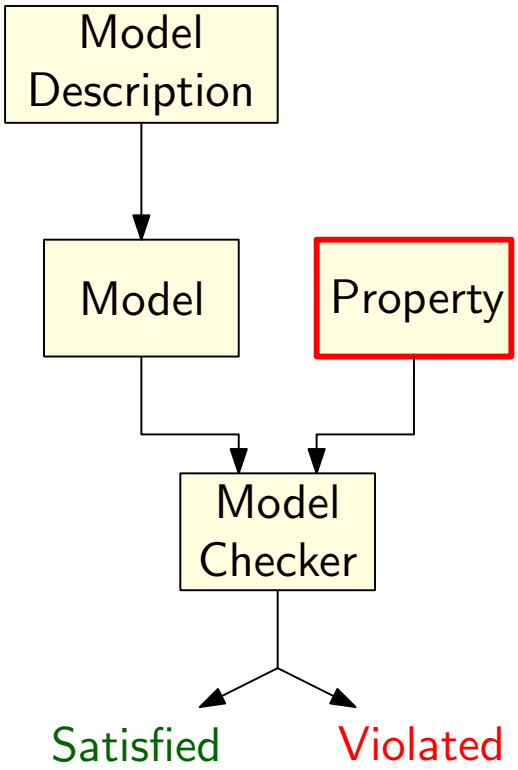
Probabilistic Model Checking



Models:



Probabilistic Model Checking



Probabilistic temporal logics

- Reachability:

$$\mathcal{P}_{\leq\lambda}(\mathcal{F}\,\neg\text{safe})$$

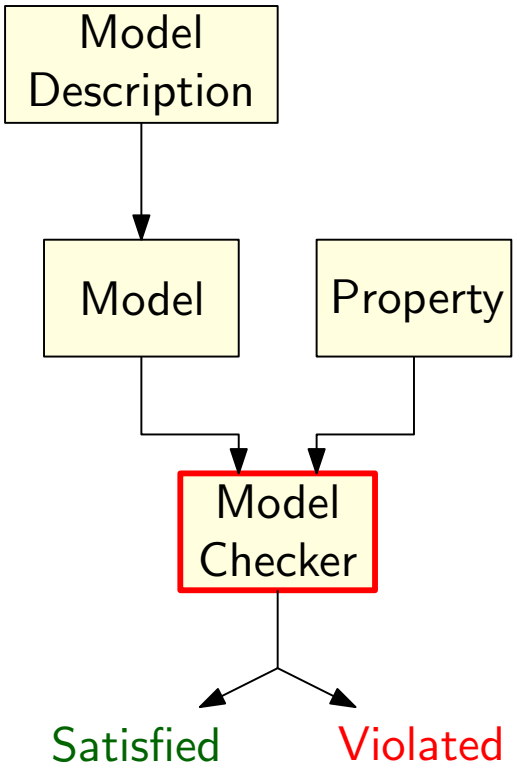
- LTL/ω-regular:

$$\mathcal{P}_{\leq\lambda}(\mathcal{F}\,\mathcal{G}\,\neg\text{safe})$$

- PCTL:

$$\mathcal{P}_{\leq\lambda}(\mathcal{F}(\mathcal{P}_{\geq\kappa}(\mathcal{G}\,\neg\text{safe})))$$

Probabilistic Model Checking



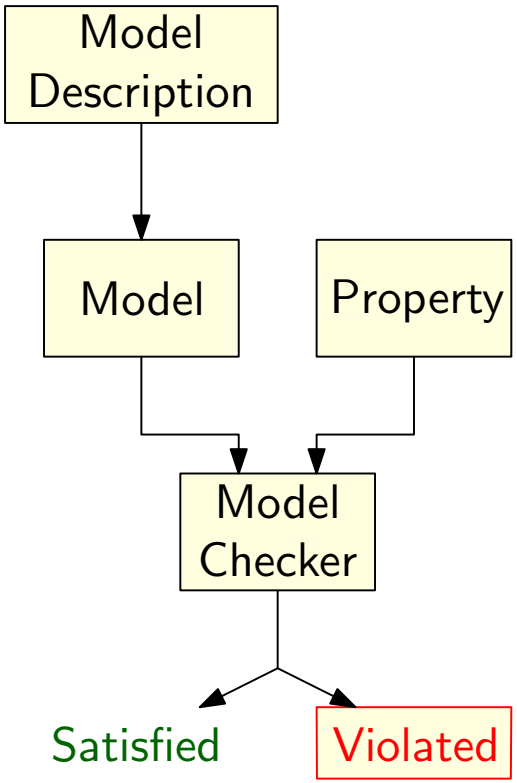
Model Checking (DTMCs):

- matrix-vector multiplication
- (linear) equation systems

E. g.: Unbounded reachability of states T:

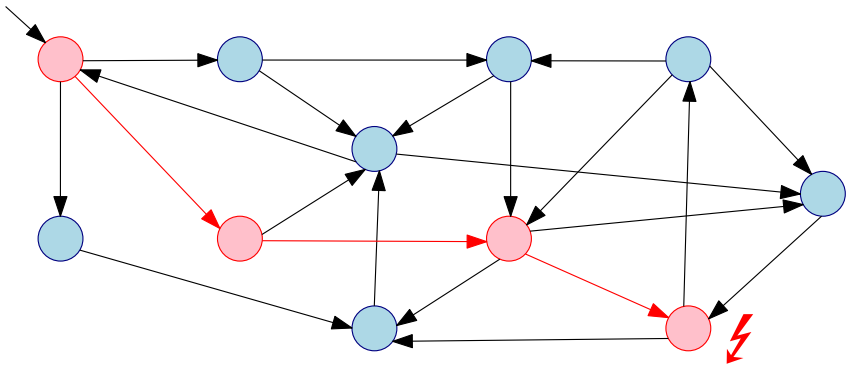
$$p_s = \begin{cases} 1, & \text{for } s \in T, \\ 0, & \text{if } T \text{ unreachable from } s, \\ \sum_{s' \in S} P(s,s') \cdot p_{s'}, & \text{otherwise.} \end{cases}$$

Probabilistic Model Checking



Property violation:

- Compute counterexample
  - Support for debugging
  - Abstraction refinement

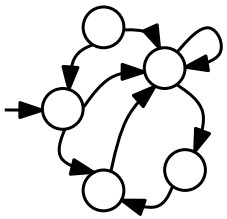


Counterexamples on Different Levels

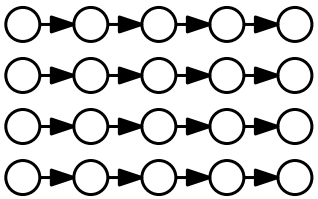
UNI FREIBURG

module M1
$[\alpha]\ g \to p_1 : f_1 + \cdots$
endmodule

Description → Minimal critical **command sets**



State space → Minimal critical **subsystems**



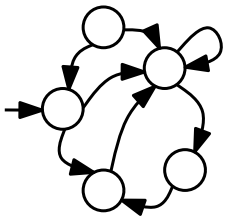
Executions → Minimal critical **path sets**

Counterexamples on Different Levels

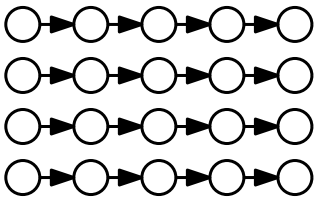
UNI FREIBURG

module M1
$[\alpha]\ g \to p_1 : f_1 + \cdots$
endmodule

Description → Minimal critical **command sets**



State space → Minimal critical **subsystems**



Executions → Minimal critical **path sets**

Path-based Counterexamples (1)

UNI FREIBURG

- **Digital systems:**
  - Safety property: $\mathcal{AG}\, safe$
  - Violation: $\mathcal{EF}\, \neg safe$
  - Counterexample: Path from the initial state to a $\neg safe$ state

Path-based Counterexamples (1)

UNI FREIBURG

- **Digital systems:**
  - Safety property: $\mathcal{AG}\,safe$
  - Violation: $\mathcal{EF}\,\neg safe$
  - Counterexample: Path from the initial state to a $\neg safe$ state

- **Probabilistic systems:**
  - Safety property: $\mathcal{P}_{\geq\lambda}(\mathcal{G}\,safe)$
  - Violation: $\mathcal{P}_{>1-\lambda}(\mathcal{F}\,\neg safe)$

Counterexample

Set C of finite paths from the initial state to a $\neg safe$ state with $\text{Prob}(C) > 1 - \lambda$

Path-based Counterexamples (2)

UNI FREIBURG

Han, Katoen, Damman
(Trans. Softw. Engin., 2009)

Smallest, most indicative counterexamples

- smallest number of paths
- highest probability among all smallest counterexamples

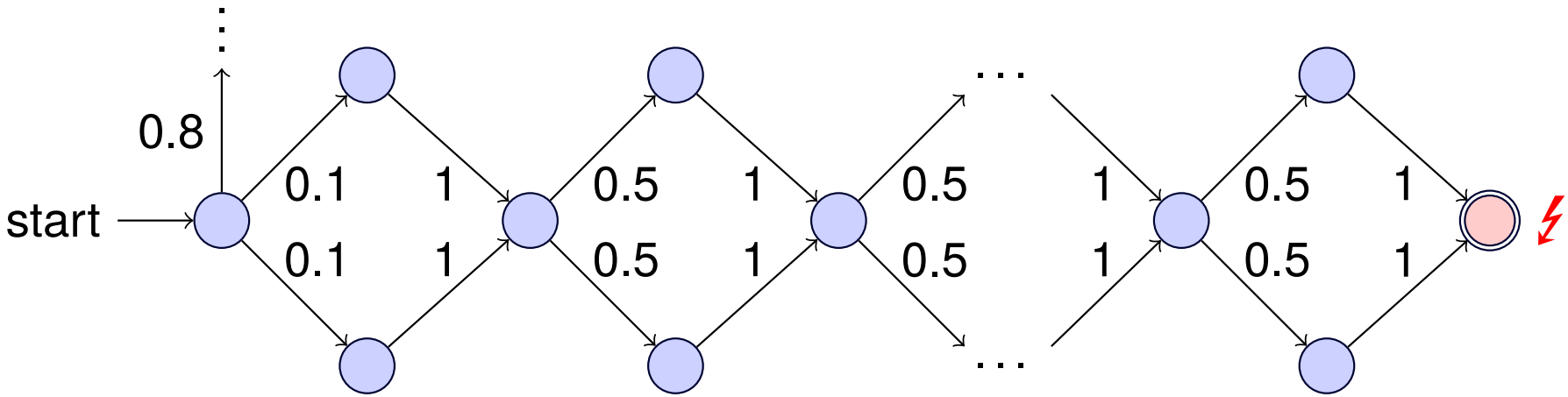
Computation: k shortest paths

- DTMC $M \rightarrow$ weighted graph $G = (S, E, w)$ with:
  - S = states of the DTMC
  - $E = \{(s, s') \in S \times S \mid P(s, s') > 0\}$
  - $w(s, s') = -\log P(s, s')$
- Shortest path in G = most probable path in M

Path-based Counterexamples (3)

Problem

The number of required paths can be extremely large—much larger than the number of states!



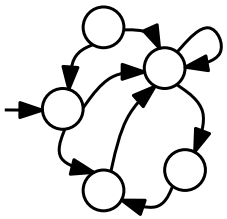
- Total probability to reach bad state: 0.2
- Probability of a single path: $0.1 \cdot 0.5^{n-1}$
- Number of paths: 2^n

Counterexamples on Different Levels

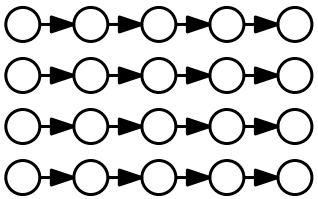
UNI FREIBURG

module M1
$[\alpha]\ g \to p_1 : f_1 + \cdots$
endmodule

Description → Minimal critical **command sets**



State space → Minimal critical **subsystems**



Executions → Minimal critical **path sets**

Critical Subsystems

UNI FREIBURG

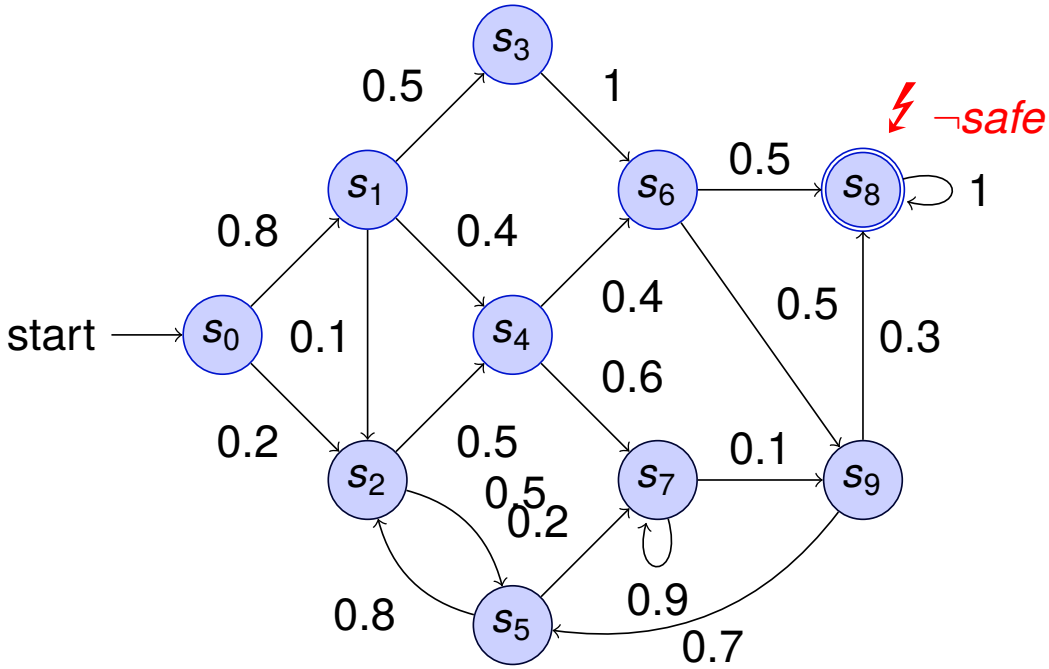
[Aljazzar/Leue, 2009; Jansen et al., 2011]

Critical subsystem

Subset S' of the states such that the probability of reaching a $\neg$*safe*-state **visiting only states from** S' is already beyond $1 - \lambda$.

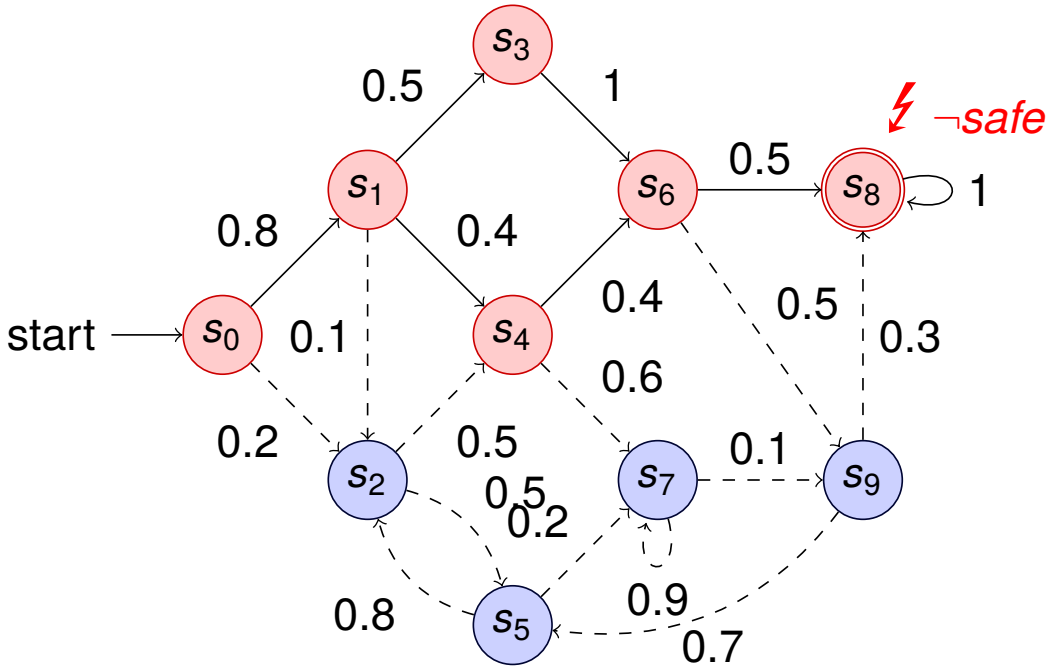
Critical subsystems: Example

$$\mathcal{P}_{\leq 0.25}(\mathcal{F}\,\neg safe)$$



Critical subsystems: Example

$$\mathcal{P}_{\leq 0.25}(\mathcal{F}\,\neg safe)$$



Minimal critical subsystems

UNI FREIBURG

Goal

Compute a critical subsystem with a minimum number of states.

Possible approaches:

- SAT-modulo-theories solving
- Mixed integer linear programming
  - Wimmer et al., TACAS 2012

MILP Formulation (DTMCs)

UNI FREIBURG

Variables

- $x_s \in \{0,1\}$ – decision variable
- $p_s \in [0,1]$ reachability probability within the subsystem

MILP Formulation (DTMCs)

UNI FREIBURG

Constraints

$$
\begin{aligned}
\text{minimize} \quad & \sum_{s \in S} x_s \\
\text{such that} \quad & \\
& p_{s_{\text{init}}} > 1 - \lambda \\
\text{target states } s\text{:} \quad & p_s = x_s \\
\text{non-target states } s\text{:} \quad & p_s \leq x_s \\
\text{non-target states } s\text{:} \quad & p_s \leq \sum_{s' \in S} P(s,s') \cdot p_{s'}
\end{aligned}
$$

Improvements

UNI FREIBURG

Speed-up by redundant constraints:

- Each state (except s_{init}) has a predecessor state

$$x_s \leq \sum_{s' \in \text{succ}(s)} x_{s'}$$

- Each state (except targets) has a successor state

$$x_s \leq \sum_{s' \in \text{pred}(s)} x_{s'}$$

- From each state a target state can be reached

$$\forall s \in S \setminus T \; \forall s' \in \text{succ}(s): \quad t_{s,s'} \leq x_s \quad \wedge \quad t_{s,s'} \leq x_{s'}$$

$$\forall s \in S \setminus T: \quad \sum_{s' \in \text{succ}(s)} t_{s,s'} = x_s$$

$$\forall s \in S \setminus T \; \forall s' \in \text{succ}(s): \quad r_s < r_{s'} + (1 - x_s)$$

- Each state can be reached from s_{init} ...

Supported Properties and Systems

UNI FREIBURG

	Reachability	ω-regular	PCTL
DTMCs	✓	✓	✓
MDPs	✓	✓	×
PAs	✓	✓	×

Experiments

UNI FREIBURG

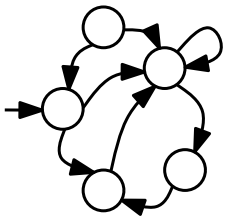
Model	States	λ	Subsystem	Time (s)	Memory
crowds5-8	68740	0.1	83	343	$<$ 1 GB
sleader4-8	12302	0.5	6150	22	$<$ 1 GB
consensus2-2	272	0.1	15	733	$<$ 1 GB
csma-2-6	66718	0.1	415	2364	$<$ 1 GB

Counterexamples on Different Levels

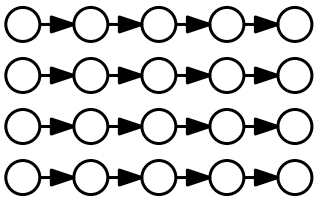
UNI FREIBURG

module M1
$[\alpha]\ g \to p_1 : f_1 + \cdots$
endmodule

Description → Minimal critical **command sets**



State space → Minimal critical **subsystems**



Executions → Minimal critical **path sets**

Counterexamples for PRISM programs

UNI FREIBURG

Wimmer et al., QEST 2013

Minimal critical command sets

Minimal subset of the commands such that their induced DTMC/MDP/PA is already buggy!

Counterexamples for PRISM programs

UNI FREIBURG

Wimmer et al., QEST 2013

Minimal critical command sets

Minimal subset of the commands such that their induced DTMC/MDP/PA is already buggy!

1. Assign a **unique label** to each command.
2. Construct the state space, labeling each transition with the commands it is created from (synchronization!)
3. Use an **MILP formulation** to minimize the number of commands.

Minimal critical command sets (DTMCs)

UNI FREIBURG

Variables

- $x_c \in \{0,1\}$ indicates whether command c is selected
- $p_s \in [0,1]$ reachability probability starting in s

Minimal critical command sets (DTMCs)

UNI FREIBURG

Constraints

$$\text{minimize} \quad \sum_{c \in C} x_c$$

such that

$$p_{s_{\text{init}}} > 1 - \lambda$$

$$s \in T: \quad p_s = 1$$

$$s \in S \setminus T: \quad p_s \leq \sum_{s' \in S} P(s,s') \cdot p_{s'}$$

$$s \in S \setminus T, c \in L(s): \quad p_s \leq x_c$$

Simplification of Commands

UNI FREIBURG

Idea

Selected command:

$$[\alpha]\, g(x) \;\rightarrow\; p_1 : f_1 \;+\; p_2 : f_2 \;+\; p_3 : f_3$$

Delete branches not relevant for the counterexample, e. g.:

$$[\alpha]\, g(x) \;\rightarrow\; \qquad\qquad p_2 : f_2$$

- Labels for the branching choices
- Track which choice generates which part of a transition
- Minimization via MILP similar to command selection

Reduction of variable domains

UNI FREIBURG

Idea

Remove values from the domains of the variables (= deletion of states) such that the remaining subsystem is still critical!

- Labels for the variable values
- Track which the mapping of states to variable assignments
- Minimization via MILP similar to command selection

Supported Properties and Systems

UNI FREIBURG

	Reachability	ω-regular	PCTL
DTMCs	✓	(✓)	(×)
MDPs	✓	(✓)	×
PAs	✓	(✓)	×

Experiments

UNI FREIBURG

Model	Comm.	Cex.	Time (s)	Lower bound	Removed branches
consensus-2-4	14	≤ 9	> 600	7	1 / 12
consensus-4-1	28	≤ 20	> 600	5	2 / 24
csma-2-4	38	36	184.05	—	20 / 90
firewire-10	68	28	545.68	—	38 / 68
wlan-2-1	76	8	0.04	—	6 / 14
wlan-2-3	76	≤ 38	> 600	32	31 / 72

- consensus-*N*-*K* = randomized consensus algorithm
- csma-*N*-*K* = IEEE 802.3 CSMA/CD network protocol
- firewire-*N* = IEEE 1394 High Performance Serial Bus protocol
- wlan-*N*-*K* = handshake protocol of IEEE 802.11 WLAN

Conclusion

UNI FREIBURG

Summary:

- Path-based counterexamples
- Critical subsystems
  - reachability
  - ω-regular
  - PCTL
- Critical command sets

Future work:

- High-level counterexamples
  - Skalability
  - Dedicated branch & bound algorithm
  - heuristic minimization
- Reward-based properties?
- Usefulness of cex for debugging/abstraction refinement?