Diagnosis, Synthesis and Analysis of Probabilistic Models

Tingting Han

October 16, 2009

What do I do?

What do I do?

Mom

Fixing computers

What do I do?

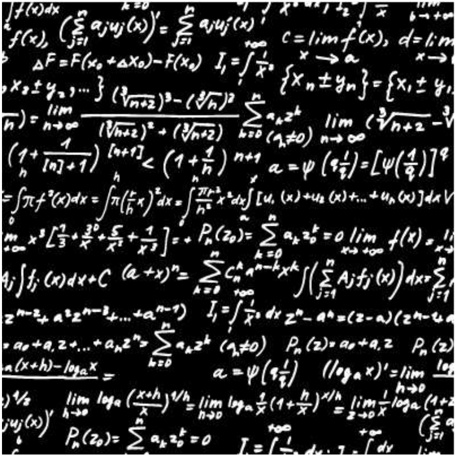
Mom

Programming

What do I do?

A college mate

Mom



Applied mathematics, Formal methods

What do I do?

A college mate

Mom

What do I do?

A college mate

Mom

Correctness!

What do I do?

A college mate

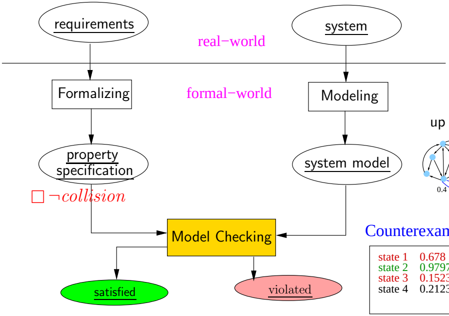
Mom

What do I do?

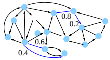
A college mate

Mom

There should be no collisions!



up to 10^7 states



Counterexample

state 1	0.678
state 2	0.9797
state 3	0.1523
state 4	0.2123

Model checking

What do I do?

A college mate

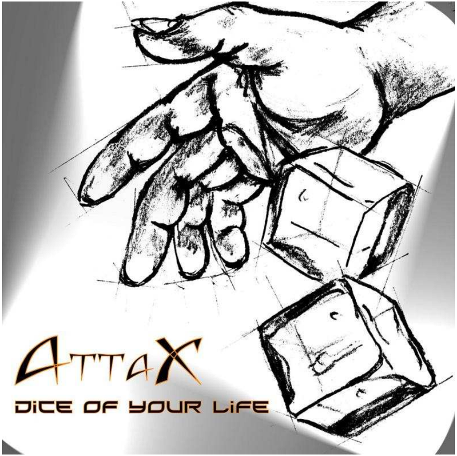
Mom

What do I do?

A researcher in a workshop

A college mate

Mom



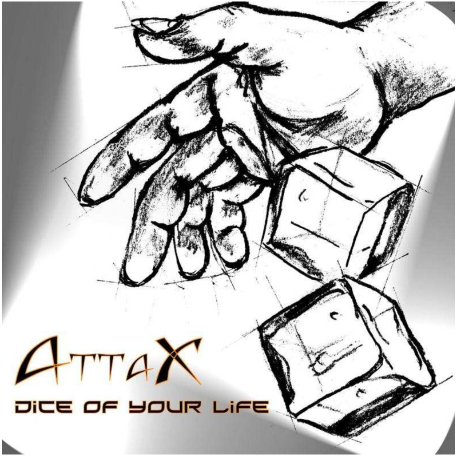
add Probability!

What do I do?

A researcher in a workshop

A college mate

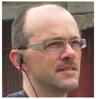
Mom



add Probability! $\Rightarrow$ probabilistic model checking

What do I do?

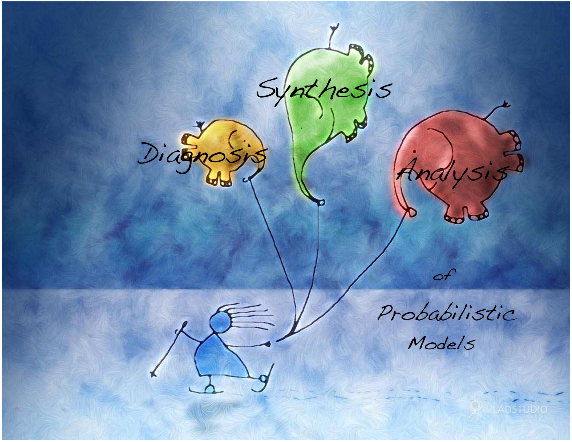
Boss



A researcher in a workshop

A college mate

Mom



Diagnosis

Counterexamples are of utmost importance

2008 Turing Award Winner Edmund Clarke:

"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."

Counterexamples are of utmost importance

2008 Turing Award Winner Edmund Clarke:

"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."

- provide diagnostic feedback

  $\Longrightarrow$ "model checking = bug hunting"

Counterexamples are of utmost importance

2008 Turing Award Winner Edmund Clarke:

"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."

- provide diagnostic feedback

  $\Longrightarrow$ "model checking = bug hunting"
- key to abstraction-refinement framework

Counterexamples are of utmost importance

2008 Turing Award Winner Edmund Clarke:

"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."

- provide diagnostic feedback

  $\Longrightarrow$ "model checking = bug hunting"
- key to abstraction-refinement framework
- schedule synthesis

Different shape of counterexamples

- LTL counterexamples are finite paths
  - "always Φ": a path ending in a $\neg\Phi$-state
  - "eventually Φ": a $\neg\Phi$-path leading to a $\neg\Phi$ cycle
  - BFS yields shortest counterexamples
- CTL counterexamples are (mostly) finite trees
- Our work: PCTL counterexamples for DTMCs

Different shape of counterexamples

- LTL counterexamples are finite paths
  - "always Φ": a path ending in a $\neg\Phi$-state
  - "eventually Φ": a $\neg\Phi$-path leading to a $\neg\Phi$ cycle
  - BFS yields shortest counterexamples

- CTL counterexamples are (mostly) finite trees

- Our work: PCTL counterexamples for DTMCs

Different shape of counterexamples

- LTL counterexamples are finite paths
  - "always Φ": a path ending in a $\neg\Phi$-state
  - "eventually Φ": a $\neg\Phi$-path leading to a $\neg\Phi$ cycle
  - BFS yields shortest counterexamples
- CTL counterexamples are (mostly) finite trees
  - universal CTL$\setminus$LTL: tree-like counterexample
or proof-like counterexample
  - existential CTL: witnesses, annotated counterexample
- Our work: PCTL counterexamples for DTMCs

Different shape of counterexamples

- LTL counterexamples are finite paths
  - "always Φ": a path ending in a $\neg\Phi$-state
  - "eventually Φ": a $\neg\Phi$-path leading to a $\neg\Phi$ cycle
  - BFS yields shortest counterexamples
- CTL counterexamples are (mostly) finite trees
  - universal CTL$\setminus$LTL: tree-like counterexample
or proof-like counterexample
  - existential CTL: witnesses, annotated counterexample
- Our work: PCTL counterexamples for DTMCs

Our contributions

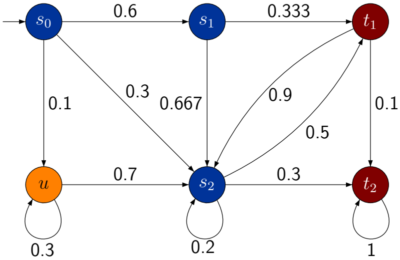
- A formal definition of (smallest) counterexamples
- Algorithms finding (smallest) counterexamples
- Compact representation of counterexamples
- Extensions to other models and logics

Our contributions

$\Longrightarrow$ A formal definition of (smallest) counterexamples

- Algorithms finding (smallest) counterexamples
- Compact representation of counterexamples
- Extensions to other models and logics

Discrete-time Markov chains



a DTMC is a triple $(S, \mathbf{P}, L)$ with state space S and state-labelling L

and $\mathbf{P}$ a stochastic matrix with $\mathbf{P}(s, s')$ = one-step probability to jump from s to s'

Probabilistic CTL (Hansson & Jonsson, 1994)

- For $a \in \mathrm{AP}$, $p \in [0,1]$, $\bowtie \; \in \{<, >, \leq, \geq\}$ and $h \in \mathbb{N}$:

$$\boxed{\begin{array}{c} \Phi, \Psi \;::=\; \mathrm{tt} \;\big|\; a \;\big|\; \Phi \wedge \Psi \;\big|\; \neg\Phi \;\big|\; \mathcal{P}_{\bowtie p}(\varphi) \\[2ex] \varphi \;::=\; \Phi \,\mathsf{U}\, \Psi \;\big|\; \Phi \,\mathsf{U}^{\leq h}\, \Psi \end{array}}$$

- $s_0 s_1 s_2 \ldots \models \Phi \,\mathsf{U}^{\leq h}\, \Psi$ if Φ holds until Ψ holds within h steps

- $s \models \mathcal{P}_{\bowtie p}(\varphi)$ if probability of set of φ-paths starting in s lies $\bowtie p$

Focus on reachability properties:

$$\boxed{\mathcal{P}_{\leq p}(\Diamond\, a) \quad \text{and} \quad \mathcal{P}_{\leq p}(\Diamond^{\leq h}\, a)}$$

where $\quad \Diamond\, a \;=\; \mathrm{tt}\,\mathsf{U}\, a \quad$ and $\quad \Diamond^{\leq h} a \;=\; \mathrm{tt}\,\mathsf{U}^{\leq h}\, a$

PCTL counterexamples for $s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a)$ $(h \in \mathbb{N} \cup \{\infty\})$

$s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a)$ iff $\Pr\left\{Paths(s, \Diamond^{\leq h} a)\right\} > p$

PCTL counterexamples for $s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a)$ $\quad (h \in \mathbb{N} \cup \{\infty\})$

$s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a)$ iff $\Pr\left\{Paths(s, \Diamond^{\leq h} a)\right\} > p$

A counterexample C such that $\Pr\left\{C\right\} > p$

PCTL counterexamples for $s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a)$ $\quad (h \in \mathbb{N} \cup \{\infty\})$

$s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a)$ iff $\Pr\left\{Paths(s, \Diamond^{\leq h} a)\right\} > p$

$$\cup$$

A counterexample C such that $\Pr\left\{C\right\} > p$

PCTL counterexamples for $s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a)$ $(h \in \mathbb{N} \cup \{\infty\})$

$s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a)$ iff $\Pr\{Paths(s, \Diamond^{\leq h} a)\} > p$

$\cup$

A counterexample C such that $\Pr\{C\} > p$

$$C = \{\underbrace{s \cdot s_1 \cdot ... \cdot s_n}_{\text{evidences}} \mid \exists s_n \models a,\ n \leq h\}$$

PCTL counterexamples for $s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a)$ $(h \in \mathbb{N} \cup \{\infty\})$

$$s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a) \quad \text{iff} \quad \Pr\{Paths(s, \Diamond^{\leq h} a)\} > p$$

$$\cup$$

A counterexample C such that $\Pr\{C\} > p$

$$C = \{\underbrace{s \cdot s_1 \cdot \ldots \cdot s_n}_{\text{evidences}} \mid \exists s_n \models a,\ n \leq h\}$$

$\rightsquigarrow$ strongest evi. $\arg\max \Pr\{\text{evi.}\}$

PCTL counterexamples for $s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a)$ $\quad (h \in \mathbb{N} \cup \{\infty\})$

$$s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a) \quad \text{iff} \quad \Pr\left\{Paths(s, \Diamond^{\leq h} a)\right\} > p$$

$$\cup$$

A counterexample C such that $\Pr\left\{C\right\} > p$

$$C = \{\underbrace{s \cdot s_1 \cdot \ldots \cdot s_n}_{\text{evidences}} \mid \exists s_n \models a,\ n \leq h\}$$

$\rightsquigarrow$ strongest evi. $\arg\max \Pr\{\text{evi.}\}$

Informative counterexamples: $|C|$ small, $\Pr\{C\}$ big

$|C|$ is #evidences in C

PCTL counterexamples for $s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a)$ $\quad (h \in \mathbb{N} \cup \{\infty\})$

$$s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a) \quad \text{iff} \quad \Pr\left\{Paths(s, \Diamond^{\leq h} a)\right\} > p$$

$$\cup$$

A counterexample C such that $\Pr\left\{C\right\} > p$

$$C = \{\underbrace{s \cdot s_1 \cdot \ldots \cdot s_n}_{\text{evidences}} \mid \exists s_n \models a,\ n \leq h\}$$

$\rightsquigarrow$ strongest evi. $\arg\max \Pr\{\text{evi.}\}$

Informative counterexamples: $|C|$ small, $\Pr\{C\}$ big

Minimal counterexamples: $|C|$ smallest

$|C|$ is $\#$evidences in C

PCTL counterexamples for $s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a)$ $\quad (h \in \mathbb{N} \cup \{\infty\})$

$$s \not\models \mathcal{P}_{\leq p}(\Diamond^{\leq h} a) \quad \text{iff} \quad \Pr\{Paths(s, \Diamond^{\leq h} a)\} > p$$

$$\cup$$

A counterexample C such that $\Pr\{C\} > p$

$$C = \{\underbrace{s \cdot s_1 \cdot \ldots \cdot s_n}_{\text{evidences}} \mid \exists s_n \models a,\ n \leq h\}$$

$\leadsto$ strongest evi. $\arg\max \Pr\{\text{evi.}\}$

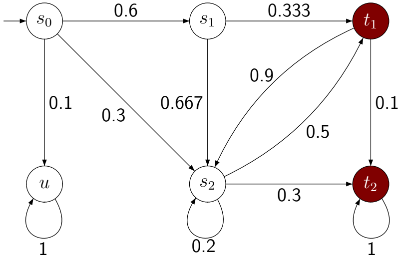
Informative counterexamples: $|C|$ small, $\Pr\{C\}$ big

Minimal counterexamples: $|C|$ smallest

Smallest counterexamples: $\arg\max_{C \text{ is minimal}} \Pr\{C\}$

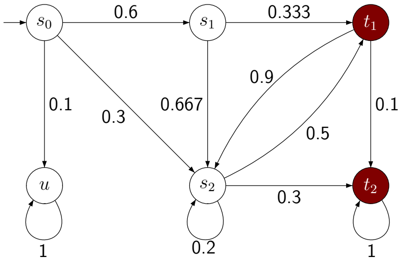
$|C|$ is #evidences in C

Evidences for $s_0 \not\models \mathcal{P}_{\leq \frac{1}{2}}(\Diamond\, \textit{red})$



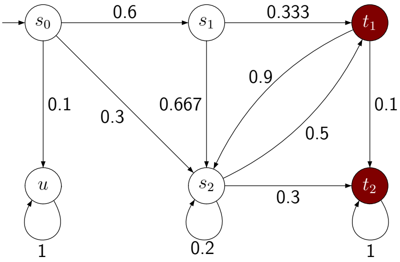
evidences	prob.
$\sigma_1 = s_0\, s_1\, t_1$	0.2
$\sigma_2 = s_0\, s_1\, s_2\, t_1$	0.2
$\sigma_3 = s_0\, s_2\, t_1$	0.15
$\sigma_4 = s_0\, s_1\, s_2\, t_2$	0.12
$\sigma_5 = s_0\, s_2\, t_2$	0.09
...	...

Strongest evidences



evidences	prob.
$\sigma_1 = s_0\, s_1\, t_1$	0.2
$\sigma_2 = s_0\, s_1\, s_2\, t_1$	0.2
$\sigma_3 = s_0\, s_2\, t_1$	0.15
$\sigma_4 = s_0\, s_1\, s_2\, t_2$	0.12
$\sigma_5 = s_0\, s_2\, t_2$	0.09
...	...

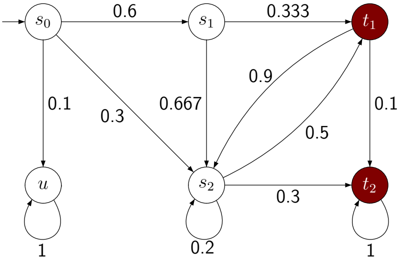
Counterexamples for $s_0 \not\models \mathcal{P}_{\leq \frac{1}{2}}(\Diamond\, red)$



evidences	prob.
$\sigma_1 = s_0\, s_1\, t_1$	0.2
$\sigma_2 = s_0\, s_1\, s_2\, t_1$	0.2
$\sigma_3 = s_0\, s_2\, t_1$	0.15
$\sigma_4 = s_0\, s_1\, s_2\, t_2$	0.12
$\sigma_5 = s_0\, s_2\, t_2$	0.09
...	...

Counterexample	$\lvert C\rvert$	Prob.
$\{\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$	5	0.76
$\{\sigma_1$ or $\sigma_2, \sigma_3, \sigma_4, \sigma_5\}$	4	0.56
$\{\sigma_1, \sigma_2, \sigma_4\}$	3	0.52
$\{\sigma_1, \sigma_2, \sigma_3\}$	3	0.55

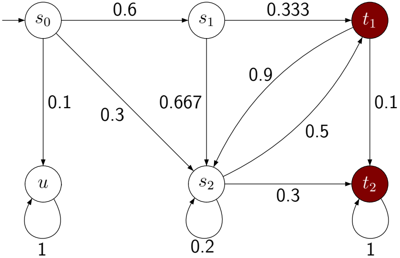
Counterexamples for $s_0 \not\models \mathcal{P}_{\leq \frac{1}{2}}(\Diamond\, \mathit{red})$



evidences	prob.
$\sigma_1 = s_0\, s_1\, t_1$	0.2
$\sigma_2 = s_0\, s_1\, s_2\, t_1$	0.2
$\sigma_3 = s_0\, s_2\, t_1$	0.15
$\sigma_4 = s_0\, s_1\, s_2\, t_2$	0.12
$\sigma_5 = s_0\, s_2\, t_2$	0.09
...	...

Counterexample	$\lvert C \rvert$	Prob.
$\{\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$	5	0.76
$\{\sigma_1$ or $\sigma_2, \sigma_3, \sigma_4, \sigma_5\}$	4	0.56
minimal $\longrightarrow \{\sigma_1, \sigma_2, \sigma_4\}$	3	0.52
minimal $\longrightarrow \{\sigma_1, \sigma_2, \sigma_3\}$	3	0.55

Counterexamples for $s_0 \not\models \mathcal{P}_{\leq \frac{1}{2}}(\Diamond\, \mathit{red})$



evidences	prob.
$\sigma_1 = s_0\, s_1\, t_1$	0.2
$\sigma_2 = s_0\, s_1\, s_2\, t_1$	0.2
$\sigma_3 = s_0\, s_2\, t_1$	0.15
$\sigma_4 = s_0\, s_1\, s_2\, t_2$	0.12
$\sigma_5 = s_0\, s_2\, t_2$	0.09
...	...

Counterexample	$\lvert C \rvert$	Prob.
$\{\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$	5	0.76
$\{\sigma_1 \text{ or } \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$	4	0.56
minimal $\longrightarrow \{\sigma_1, \sigma_2, \sigma_4\}$	3	0.52
smallest $\longrightarrow \{\sigma_1, \sigma_2, \sigma_3\}$	3	0.55

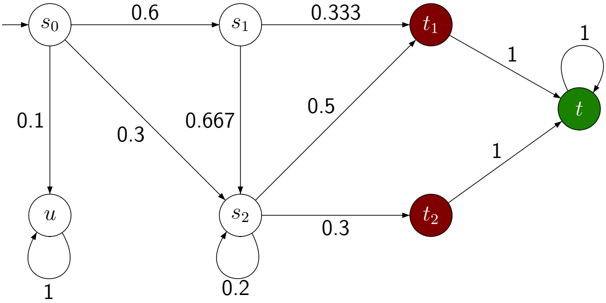
Our contributions

- A formal definition of (smallest) counterexamples

$\Longrightarrow$ Algorithms finding (smallest) counterexamples

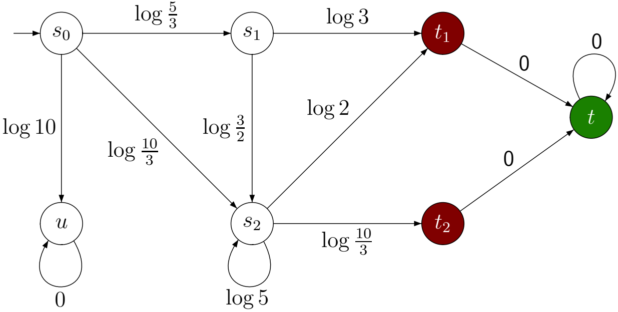
- Compact representation of counterexamples
- Extensions to other models and logics

Obtaining smallest counterexamples — Model adaptation



Insert a sink state and redirect all outgoing edges of red-states to it

A weighted digraph



Turn it into a weighted digraph with $w(s, s') = \log \left(\frac{1}{\mathbf{P}(s, s')} \right)$

A simple derivation

For finite path $\sigma = s_0\, s_1\, s_2\, \ldots\; s_n$:

$$\begin{aligned}
w(\sigma) &= w(s_0, s_1) + w(s_1, s_2) + \ldots + w(s_{n-1}, s_n) \\
&= \log \frac{1}{\mathbf{P}(s_0, s_1)} + \log \frac{1}{\mathbf{P}(s_1, s_2)} + \ldots + \log \frac{1}{\mathbf{P}(s_{n-1}, s_n)} \\
&= \log \frac{1}{\mathbf{P}(s_0, s_1) \cdot \mathbf{P}(s_1, s_2) \cdot \ldots \cdot \mathbf{P}(s_{n-1}, s_n)} \\
&= \log \frac{1}{\Pr(\sigma)}
\end{aligned}$$

$$\boxed{\underbrace{\Pr(\widehat{\sigma}) \;\geq\; \Pr(\sigma)}_{\text{in DTMC } \mathcal{D}} \quad \text{if and only if} \quad \underbrace{w(\widehat{\sigma}) \;\leq\; w(\sigma)}_{\text{in digraph } G(\mathcal{D})}}$$

What does this mean?

- Finding a $\underbrace{\text{strongest evidence}}_{\text{(most probable path)}}$ is a shortest path (SP) problem
  - apply standard SP algorithms

$\Longrightarrow$ linear time complexity

- Finding a $\underbrace{\text{smallest counterex}}_{\text{(}k\text{ most probable paths)}}$ is a k-shortest paths (KSP) problem

What does this mean?

- Finding a $\underbrace{\text{strongest evidence}}_{\text{(most probable path)}}$ is a shortest path (SP) problem
  - apply standard SP algorithms

$\Longrightarrow$ linear time complexity

- Finding a $\underbrace{\text{smallest counterex}}_{\text{(k most probable paths)}}$ is a k-shortest paths (KSP) problem

What does this mean?

- Finding a $\underbrace{\text{strongest evidence}}_{\text{(most probable path)}}$ is a shortest path (SP) problem
  - apply standard SP algorithms

$\Longrightarrow$ linear time complexity

- Finding a $\underbrace{\text{smallest counterex}}_{(k \text{ most probable paths})}$ is a k-shortest paths (KSP) problem
  - dynamically determine k

$\Longrightarrow$ generate C incrementally and halt when $\Pr\{C\} > p$

What does this mean?

- Finding a $\underbrace{\text{strongest evidence}}_{\text{(most probable path)}}$ is a shortest path (SP) problem
  - apply standard SP algorithms

$\Longrightarrow$ linear time complexity

- Finding a $\underbrace{\text{smallest counterex}}_{(k \text{ most probable paths})}$ is a k-shortest paths (KSP) problem
  - dynamically determine k

$\Longrightarrow$ generate C incrementally and halt when $\Pr\{C\} > p$

Algorithms and time complexity

counterexample problem	shortest path problem	algorithm	time complexity
unbounded SE	SP	Dijkstra	$\mathcal{O}(M + N \cdot \log N)$
bounded h SE	HSP	Bellman-Ford	$\mathcal{O}(h \cdot M)$
unbounded SC	KSP	Eppstein	$\mathcal{O}(M + N \cdot \log N + k)$
bounded h SC	HKSP	adapted REA	$\mathcal{O}\left(h \cdot M + h \cdot k \cdot \log N\right)$

$N = |S|$, $M = \#$ transitions, $h =$ hop count, $k = \#$ shortest paths

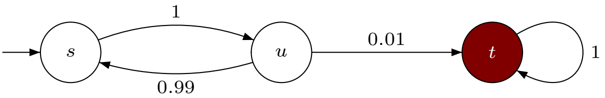
Our contributions

- A formal definition of (smallest) counterexamples
- Algorithms finding (smallest) counterexamples

$\Longrightarrow$ Compact representation of counterexamples

- Extensions to other models and logics

On the size of counterexamples



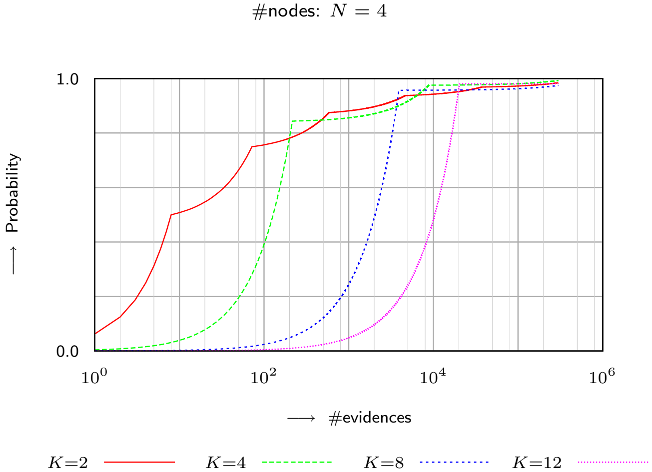
A smallest counterexample for $s \not\models \mathcal{P}_{\leq 0.9999}(\Diamond\, red)$ contains paths

$$s\,u\,t,\quad s\,u\,s\,u\,t,\quad s\,u\,s\,u\,s\,u\,t,\quad \ldots\ldots,\ \underbrace{s\,u}_{k\text{ times}}\ t$$

where k is the smallest integer such that $1 - 0.99^{k-1} > 0.9999$

The smallest counterexample has $k = 689$ evidences

Synchronous leader election $\mathcal{P}_{\leq 0.99}(\Diamond\ \mathit{leader})$



size of counterexample is double exponential in problem size

Use regular expressions!

- Size of counterexamples is mainly influenced by loops
  - each loop-traversal yields another path in counterexample

- Idea: represent sets of "similar" finite paths by a *regular expression*

- How?

$\boxed{\text{Automata theory!}}$

Use regular expressions!

- Size of counterexamples is mainly influenced by loops
  - each loop-traversal yields another path in counterexample
- Idea: represent sets of "similar" finite paths by a *regular expression*
- How?

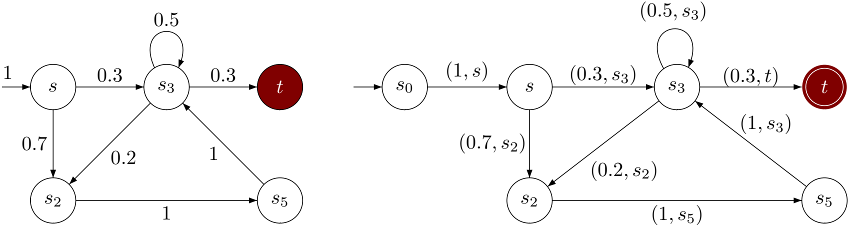
Automata theory!

Use regular expressions!

- Size of counterexamples is mainly influenced by loops
  - each loop-traversal yields another path in counterexample
- Idea: represent sets of "similar" finite paths by a *regular expression*
- How?

Automata theory!

From DTMCs to DFAs



alphabet Σ consists of symbols of the form (p, s)

- DTMC (rooted at s) $\longrightarrow$ DFA (t is accepting)
- a set of s-t paths $Paths_r$ $\longleftarrow$ a regular expression r

$\Longrightarrow$ probability $\Pr(Paths_r) = \Pr(r)$

Regular expressions

The set of regular expressions $\mathcal{R}(\Sigma)$:

$$
\begin{array}{llll}
r, r' & ::= & \varepsilon & \text{empty} \\
 & | & (p, s) & \text{letter} \\
 & | & r|r' & \text{choice} \\
 & | & r.r' & \text{concatenation} \\
 & | & r^* & \text{repetition}
\end{array}
$$

Evaluation $val : \mathcal{R}(\Sigma) \rightarrow [0, 1]$:

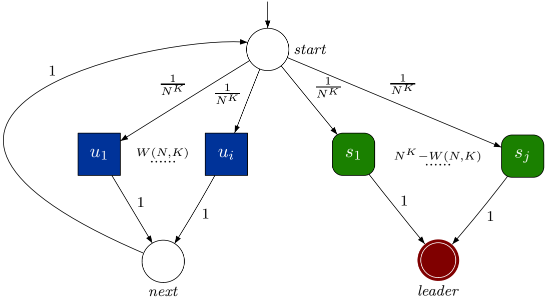
$$
\begin{array}{lcl}
val(\varepsilon) & = & 1 \\
val((p, s)) & = & p \\
val(r|r') & = & val(r) + val(r') \\
val(r.r') & = & val(r) \cdot val(r') \\
val(r^*) & = & \begin{cases} 1 & \text{if val(r)=1} \\ \dfrac{1}{1 - val(r)} & \text{otherwise} \end{cases}
\end{array}
$$

For regular express r of DFA $\mathcal{A}_{\mathcal{D}}$ with accept state t:

$$val(r) = \Pr^{\mathcal{D}}\{\sigma \in Paths(s) \mid \sigma \models \Diamond\, t\}$$

- $\Pr(r)$ exceeds p ($= r$ is a counterexample)
- r is "minimal": deletion of some "branch" of r yields no counterexample

Regular expression counterexample for leader election



Regular expression for the counterexample:

$$r(N,K) \;=\; start.\; [(u_1|\cdots|u_i)\,.next.\; start]^*.\; (s_1|\cdots|s_j).\; leader$$

Our contributions

- A formal definition of (smallest) counterexamples
- Algorithms finding (smallest) counterexamples
- Compact representation of counterexamples

$\Longrightarrow$ Extensions to other models and logics

Extensions

The counterexample generation for DTMC + $\mathcal{P}_{\leq p}(\Diamond^{\leqslant h} a)$ can be generalized to

- DTMC +
  - full PCTL
  - LTL, or PCTL*
- [nondeterminism]
  MDP + (PCTL, or LTL, or PCTL*, or with fairness conditions)
- [reward] DMRM + PRCTL
- [continuous-time] CTMC + CSL
- ...

Counterexamples are *en vogue*

- Different techniques
  - Heuristic search algo. for DTMCs/CTMCs (Aljazzar *et al.* FORMATS'05, '06, TSE'09)
  - Bounded model checking for DTMC counterexamples (Becker *et al.* VMCAI'09)
- Different models and logics
  - Counterexamples for CTMCs (Han & Katoen ATVA'07)
  - Counterexamples for MDPs (Andres *et al.*, HVC'08, Aljazzar & Leue QEST'09)
  - Counterexamples for conditional PCTL (Andres & van Rossum TACAS'08)
  - Counterexamples for LTL (Schmalz *et al.*, CONCUR'09)
- Different applications
  - Proof refutations for probabilistic programs (McIver *et al.* FM'08)
  - Counterexample-guided abstraction refinement (Hermanns *et al.* CAV'08)
    (Chadha & Viswamanathan Corr abs 08)
- Counterexample visualization (Aljazzar & Leue QEST'08)

Diagnosis — Summary

- A formal definition of (smallest) counterexamples
  - a set of paths with sufficient probability mass
- Algorithms finding (smallest) counterexamples
  - exploit k-shortest path algorithms
- Compact representation of counterexamples
  - by regular expressions!
- Extensions to other models and logics
  - the framework is general!

Synthesis

Problem Statement

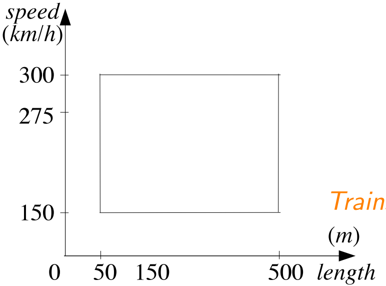
Parameters in the model:

- Values are hard to settle in the early phase of design
- Usually have physical meanings: temperature, time, pressure, speed...

Problem Statement

Parameters in the model:

- Values are hard to settle in the early phase of design
- Usually have physical meanings: temperature, time, pressure, speed...

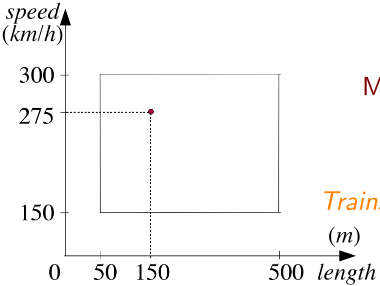


$Trains[speed := ?, length := ?]$ and $\mathcal{P}_{>0.9999}(\neg col)$

Problem Statement

Parameters in the model:

- Values are hard to settle in the early phase of design
- Usually have physical meanings: temperature, time, pressure, speed...



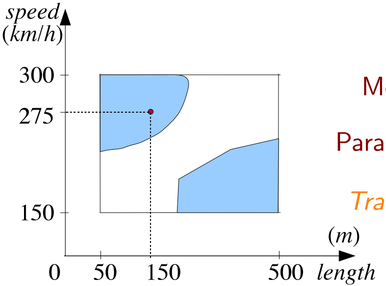
Model checking: $\mathcal{M}[\mathcal{X} := \mathcal{V}] \quad \overset{?}{\models} \quad \Phi$

$\mathit{Trains}[\mathit{speed} := 275, \mathit{length} := 150] \overset{?}{\models} \mathcal{P}_{>0.9999}(\neg \mathit{col})$

Problem Statement

Parameters in the model:

- Values are hard to settle in the early phase of design
- Usually have physical meanings: temperature, time, pressure, speed...



Model checking: $\mathcal{M}[\mathcal{X} := \mathcal{V}] \quad \overset{?}{\models} \quad \Phi$

Parameter synthesis: $\mathcal{M}[\mathcal{X} := ?] \quad \models \quad \Phi$

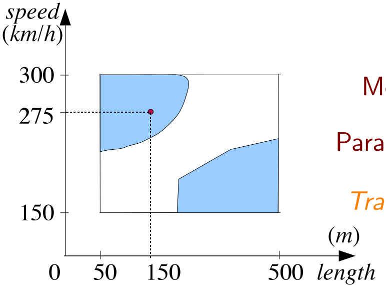
$\mathit{Trains}[\mathit{speed} := ?, \mathit{length} := ?] \models \mathcal{P}_{>0.9999}(\neg \mathit{col})$

For which parameter values Φ holds?

Problem Statement

Parameters in the model:

- Values are hard to settle in the early phase of design
- Usually have physical meanings: temperature, time, pressure, speed...



Model checking: $\mathcal{M}[\mathcal{X} := \mathcal{V}] \quad \models^{?} \quad \Phi$

Parameter synthesis: $\mathcal{M}[\mathcal{X} := ?] \quad \models \quad \Phi$

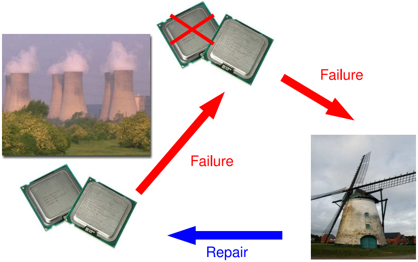
$\mathit{Trains}[\mathit{speed} := ?, \mathit{length} := ?] \models \mathcal{P}_{>0.9999}(\neg \mathit{col})$

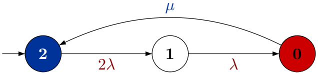
For which parameter values Φ holds?

Parameter synthesis is much harder than model checking!

Parametric CTMC

- Continuous-Time Markov chain (CTMC) $\mathcal{C} = (S, \mathbf{R}, s_0)$



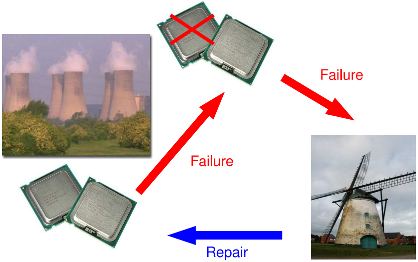


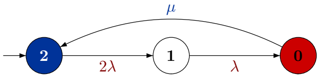
failure rate: $\lambda = 3$

repair rate: $\mu = 5$

Parametric CTMC

- Continuous-Time Markov chain (CTMC) $\mathcal{C} = (S, \mathbf{R}, s_0)$

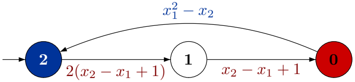




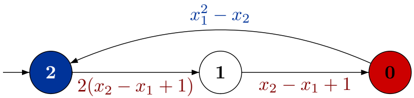
failure rate: $\lambda = 3$
repair rate: $\mu = 5$

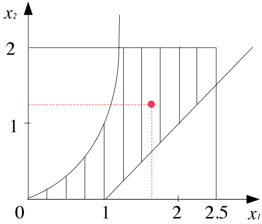
- Parametric CTMC (pCTMC) $\mathcal{C}^{(\mathcal{X})} = (S, \mathbf{R}^{(\mathcal{X})}, s_0)$

Rates are polynomial expressions instead of constants!



Initial region





- Every parameter has a closed range $x_i \in [l_i, u_i]$, $x \in \mathcal{X}$

$$\zeta_{range} = \bigwedge_{i=1}^{m} l_i \leqslant x_i \leqslant u_i \qquad \text{(range region)}$$

$0 \leqslant x_1 \leqslant 2.5$
$0 \leqslant x_2 \leqslant 2$

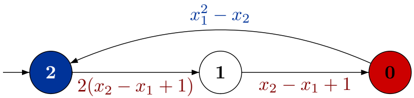
- Every rate is nonnegative

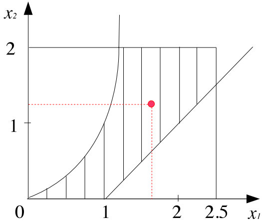
$$\zeta_{rate} = \bigwedge_{s,s' \in S} \mathbf{R}^{(x)}(s,s') \geq 0 \qquad \text{(rate region)}$$

- Initial region $\zeta_{init} = \zeta_{range} \wedge \zeta_{rate}$

- Instance CTMC: $\mathcal{C}[u]$: instantiation with a valuation (point) in ζ_{init}

Initial region





- Every parameter has a closed range $x_i \in [l_i, u_i]$, $x \in \mathcal{X}$

$$\zeta_{range} = \bigwedge_{i=1}^{m} l_i \leqslant x_i \leqslant u_i \qquad \text{(range region)}$$

$$0 \leqslant x_1 \leqslant 2.5$$
$$0 \leqslant x_2 \leqslant 2$$

- Every rate is nonnegative

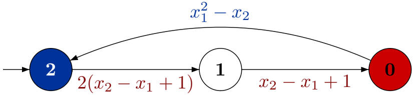
$$\zeta_{rate} = \bigwedge_{s,s' \in S} \mathbf{R}^{(\mathcal{X})}(s, s') \geqslant 0 \qquad \text{(rate region)}$$

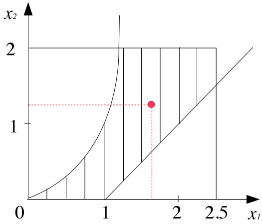
$$x_1^2 - x_2 \geqslant 0$$
$$x_2 - x_1 + 1 \geqslant 0$$

- Initial region $\zeta_{init} = \zeta_{range} \wedge \zeta_{rate}$

- Instance CTMC: $\mathcal{C}[v]$, instantiation with a valuation (point) in ζ_{init}

Initial region





- Every parameter has a closed range $x_i \in [l_i, u_i]$, $x \in \mathcal{X}$

$$\zeta_{range} = \bigwedge_{i=1}^{m} l_i \leqslant x_i \leqslant u_i \qquad \text{(range region)}$$

$$0 \leqslant x_1 \leqslant 2.5$$
$$0 \leqslant x_2 \leqslant 2$$

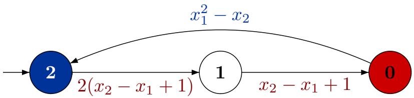
- Every rate is nonnegative

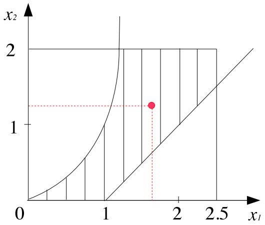
$$\zeta_{rate} = \bigwedge_{s,s' \in S} \mathbf{R}^{(\mathcal{X})}(s, s') \geqslant 0 \qquad \text{(rate region)}$$

$$x_1^2 - x_2 \geqslant 0$$
$$x_2 - x_1 + 1 \geqslant 0$$

- Initial region $\zeta_{init} = \zeta_{range} \wedge \zeta_{rate}$

Initial region





- Every parameter has a closed range $x_i \in [l_i, u_i]$, $x \in \mathcal{X}$

$$\zeta_{range} = \bigwedge_{i=1}^{m} l_i \leqslant x_i \leqslant u_i \qquad \text{(range region)}$$

$$0 \leqslant x_1 \leqslant 2.5$$
$$0 \leqslant x_2 \leqslant 2$$

- Every rate is nonnegative

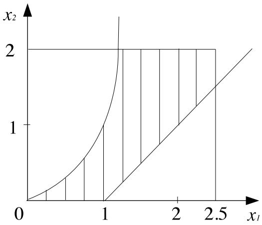
$$\zeta_{rate} = \bigwedge_{s,s' \in S} \mathbf{R}^{(\mathcal{X})}(s, s') \ \geqslant \ 0 \qquad \text{(rate region)}$$

$$x_1^2 - x_2 \geqslant 0$$
$$x_2 - x_1 + 1 \geqslant 0$$

- Initial region $\zeta_{init} = \zeta_{range} \wedge \zeta_{rate}$

- Instance CTMC: $\mathcal{C}[v]$, instantiation with a valuation (point) in ζ_{init}

Tasks: Finding synthesis regions

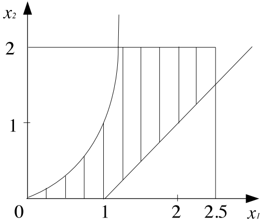


initial region ζ_{init}

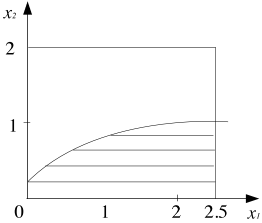
Tasks: Finding synthesis regions

1. Find a synthesis region ζ_{syn} in ζ_{range}, such that for all valuations v in ζ_{syn},

$$\mathcal{C}[v] \models \mathcal{P}_{[p_l,p_u]}(\Diamond^{\leqslant t} s_{goal})$$



initial region ζ_{init}



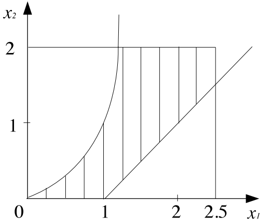
ζ_{syn}

Tasks: Finding synthesis regions

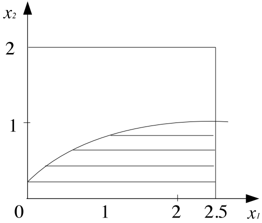
1. Find a synthesis region ζ_{syn} in ζ_{range}, such that for all valuations v in ζ_{syn},

$$\mathcal{C}[v] \models \mathcal{P}_{[p_l, p_u]}(\Diamond^{\leqslant t} s_{goal})$$

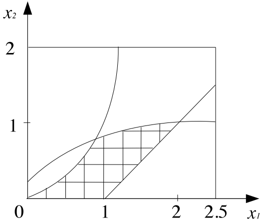
2. Intersect with initial region ζ_{init} s.t. $\zeta'_{syn} = \zeta_{init} \cap \zeta_{syn}$



initial region ζ_{init}

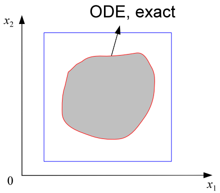


ζ_{syn}



$\zeta'_{syn} = \zeta_{init} \cap \zeta_{syn}$

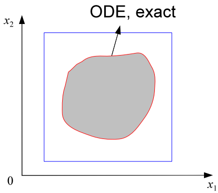
Solutions



transient probability ↰

[BHHK03] $s_0 \models^{\mathcal{C}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\boxed{\pi_{s_{goal},t}} < p$

Solutions

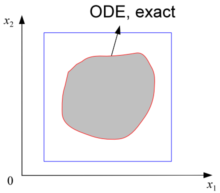


transient probability ↰

$$\text{[BHHK03]} \quad s_0 \models^{\mathcal{C}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal}) \quad \text{iff} \quad \boxed{\pi_{s_{goal},t}} < p$$

$$s_0 \models^{\mathcal{C}(\mathcal{X})} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal}) \quad \text{iff} \quad \pi_{s_{goal},t}(x_1, x_2) < p$$

Solutions



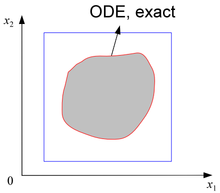
transient probability ↰

[BHHK03] $s_0 \models^{\mathcal{C}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\boxed{\pi_{s_{goal},t}} < p$

$s_0 \models^{\mathcal{C}^{(\mathcal{X})}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\pi_{s_{goal},t}(x_1, x_2) < p$

region boundary curve: $\pi_{s_{goal},t}(x_1, x_2) = p$

Solutions



transient probability ↰

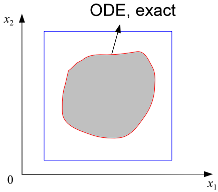
[BHHK03] $s_0 \models^{\mathcal{C}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\boxed{\pi_{s_{goal},t}} < p$

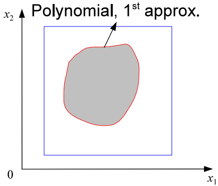
$s_0 \models^{\mathcal{C}^{(\mathcal{X})}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\pi_{s_{goal},t}(x_1, x_2) < p$

region boundary curve: $\pi_{s_{goal},t}(x_1, x_2) = p$

ODE $$\frac{d\vec{\pi}_t(x_1, x_2)}{dt} = \vec{\pi}_t \cdot \mathbf{Q}^{(\mathcal{X})}$$

Solutions





transient probability ↰

[BHHK03] $s_0 \models^{\mathcal{C}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\boxed{\pi_{s_{goal},t}} < p$

$s_0 \models^{\mathcal{C}^{(\mathcal{X})}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\pi_{s_{goal},t}(x_1, x_2) < p$

region boundary curve: $\pi_{s_{goal},t}(x_1, x_2) = p$

ODE

|

(uniformization)

↓

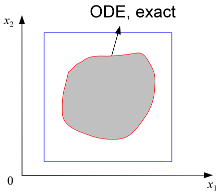
polynomial

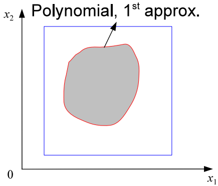
$$\frac{d\vec{\pi}_t(x_1, x_2)}{dt} = \vec{\pi}_t \cdot \mathbf{Q}^{(\mathcal{X})}$$

$\|\vec{\pi}_t - \vec{\tilde{\pi}}_t\| \leqslant \varepsilon$ (required accuracy)

$$\vec{\tilde{\pi}}_t(x_1, x_2) = \vec{\tilde{\pi}}_0 \cdot \sum_{i=0}^{k_\varepsilon} e^{-qt} \frac{(qt)^i}{i!} (\mathbf{P}^{(\mathcal{X})})^i$$

Solutions





transient probability ↰

[BHHK03] $s_0 \models^{\mathcal{C}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\boxed{\pi_{s_{goal},t}} < p$

$s_0 \models^{\mathcal{C}^{(\mathcal{X})}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\pi_{s_{goal},t}(x_1, x_2) < p$

region boundary curve: $\pi_{s_{goal},t}(x_1, x_2) = p$

ODE $\qquad \dfrac{d\vec{\pi}_t(x_1, x_2)}{dt} = \vec{\pi}_t \cdot \mathbf{Q}^{(\mathcal{X})}$

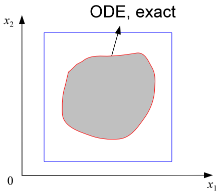
|

(uniformization) $\qquad \|\vec{\pi}_t - \vec{\tilde{\pi}}_t\| \leqslant \varepsilon$ (required accuracy)

↓

polynomial $\qquad \vec{\tilde{\pi}}_t(x_1, x_2) = \vec{\tilde{\pi}}_0 \cdot \sum_{i=0}^{k_\varepsilon} e^{-qt} \frac{(qt)^i}{i!} (\mathbf{P}^{(\mathcal{X})})^i$

Solutions

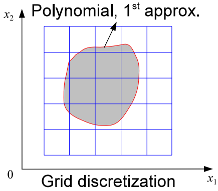


transient probability ↰

[BHHK03] $s_0 \models^{\mathcal{C}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\boxed{\pi_{s_{goal},t}} < p$

$s_0 \models^{\mathcal{C}^{(\mathcal{X})}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\pi_{s_{goal},t}(x_1, x_2) < p$

region boundary curve: $\pi_{s_{goal},t}(x_1, x_2) = p$



ODE

|

(uniformization)

↓

polynomial

|

(discretization)

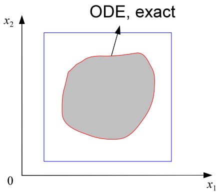
$$\frac{d\vec{\pi}_t(x_1, x_2)}{dt} = \vec{\pi}_t \cdot \mathbf{Q}^{(\mathcal{X})}$$

$\|\vec{\pi}_t - \vec{\tilde{\pi}}_t\| \leqslant \varepsilon$ (required accuracy)

$$\vec{\tilde{\pi}}_t(x_1, x_2) = \vec{\tilde{\pi}}_0 \cdot \sum_{i=0}^{k_\varepsilon} e^{-qt} \frac{(qt)^i}{i!} (\mathbf{P}^{(\mathcal{X})})^i$$

using grid

Solutions

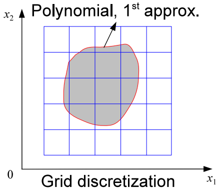


transient probability ↰

[BHHK03] $s_0 \models^{\mathcal{C}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\boxed{\pi_{s_{goal},t}} < p$

$s_0 \models^{\mathcal{C}^{(\mathcal{X})}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\pi_{s_{goal},t}(x_1, x_2) < p$

region boundary curve: $\pi_{s_{goal},t}(x_1, x_2) = p$



ODE | (uniformization) ↓ polynomial | (discretization) | (2 algorithms) ↓ a set of line segments

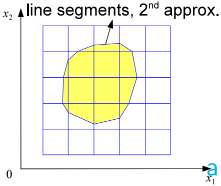
$$\frac{d\vec{\pi}_t(x_1, x_2)}{dt} = \vec{\pi}_t \cdot \mathbf{Q}^{(\mathcal{X})}$$

$\|\vec{\pi}_t - \tilde{\vec{\pi}}_t\| \leqslant \varepsilon$ (required accuracy)

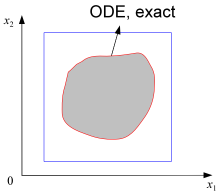
$$\tilde{\vec{\pi}}_t(x_1, x_2) = \tilde{\vec{\pi}}_0 \cdot \sum_{i=0}^{k_\varepsilon} e^{-qt} \frac{(qt)^i}{i!} (\mathbf{P}^{(\mathcal{X})})^i$$

using grid

symbolic/non-symbolic algorithms



Solutions

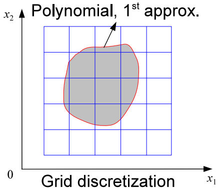


transient probability ↰

[BHHK03] $s_0 \models^{\mathcal{C}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\boxed{\pi_{s_{goal},t}} < p$

$s_0 \models^{\mathcal{C}^{(\mathcal{X})}} \mathcal{P}_{<p}(\Diamond^{\leqslant t} s_{goal})$ iff $\pi_{s_{goal},t}(x_1, x_2) < p$

region boundary curve: $\pi_{s_{goal},t}(x_1, x_2) = p$

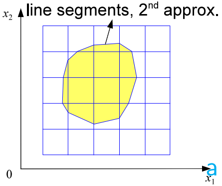


ODE $\qquad \dfrac{d\vec{\pi}_t(x_1, x_2)}{dt} = \vec{\pi}_t \cdot \mathbf{Q}^{(\mathcal{X})}$

(uniformization) $\qquad \|\vec{\pi}_t - \tilde{\vec{\pi}}_t\| \leqslant \varepsilon$ (required accuracy)

↓

polynomial $\qquad \tilde{\vec{\pi}}_t(x_1, x_2) = \tilde{\vec{\pi}}_0 \cdot \sum_{i=0}^{k_\varepsilon} e^{-qt} \frac{(qt)^i}{i!} (\mathbf{P}^{(\mathcal{X})})^i$



(discretization) $\qquad$ using grid

(2 algorithms) $\qquad$ symbolic/non-symbolic algorithms

↓

a set of line segments $\qquad$ polygon approximation

Non-symbolic method

- For each grid point: Instantiation and checking

$$\mathcal{C}[\textit{each grid point}] \overset{?}{\models} \mathcal{P}_{<p}(\Diamond^{\leqslant t} goal)$$

Non-symbolic method

- For each grid point: Instantiation and checking

$$\mathcal{C}[\textit{each grid point}] \overset{?}{\models} \mathcal{P}_{<p}(\Diamond^{\leqslant t} goal)$$

$\Longrightarrow$ CTMC model checking

Non-symbolic method

- For each grid point: Instantiation and checking

$$\mathcal{C}[\textit{each grid point}] \overset{?}{\models} \mathcal{P}_{<p}(\Diamond^{\leqslant t} goal)$$

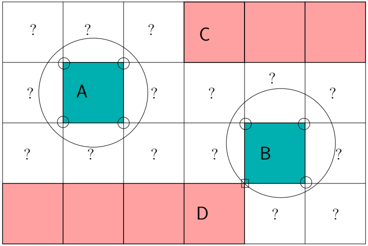
$\Longrightarrow$ CTMC model checking

$$\Longrightarrow \quad \textit{point} = \begin{cases} \top & \text{if } Prob(\Diamond^{\leqslant t} goal) < p \text{ in } \mathcal{C}[\textit{point}] \quad (\textit{point} \text{ in}) \\ \bot & \text{if } Prob(\Diamond^{\leqslant t} goal) > p \text{ in } \mathcal{C}[\textit{point}] \quad (\textit{point} \text{ out}) \\ \top\!\!\!\!\bot & \text{if } Prob(\Diamond^{\leqslant t} goal) = p \text{ in } \mathcal{C}[\textit{point}] \quad (\textit{point} \text{ on}) \end{cases}$$

Marking

- For each grid cell: Marking

$$\text{grid cell } cell = \begin{cases} \boxed{\top} & \text{if } cell\text{'s grid points are } 4\top \text{ or } 3\top 1\mathbb{I} \\ \boxed{\bot} & \text{if } cell\text{'s grid points are } 4\bot \text{ or } 3\bot 1\mathbb{I} \\ \boxed{?} & \text{otherwise} \end{cases}$$

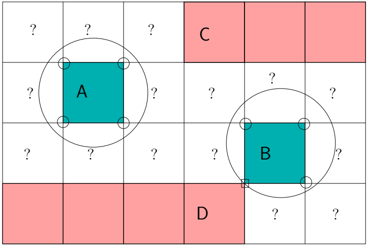


(a) Marking criterion

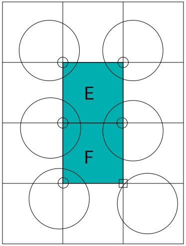
Marking

- For each grid cell: Marking

$$\text{grid cell } cell = \begin{cases} \boxed{\top} & \text{if } cell\text{'s grid points are } 4\top \text{ or } 3\top 1 \text{I} \\ \boxed{\perp} & \text{if } cell\text{'s grid points are } 4\perp \text{ or } 3\perp 1 \text{I} \\ \boxed{?} & \text{otherwise} \end{cases}$$



(d) Marking criterion

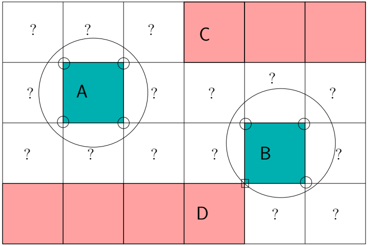


(e) False positive

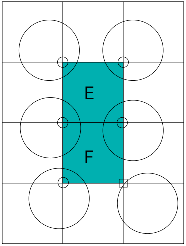
Marking

- For each grid cell: Marking

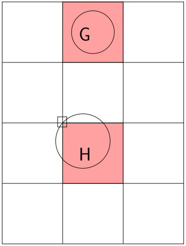
$$\text{grid cell } cell = \begin{cases} \boxed{\top} & \text{if } cell\text{'s grid points are } 4\top \text{ or } 3\top 1\text{⌶} \\ \boxed{\bot} & \text{if } cell\text{'s grid points are } 4\bot \text{ or } 3\bot 1\text{⌶} \\ \boxed{?} & \text{otherwise} \end{cases}$$



(g) Marking criterion



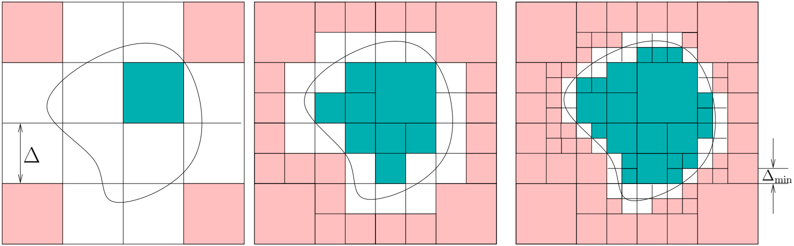
(h) False positive



(i) False negative

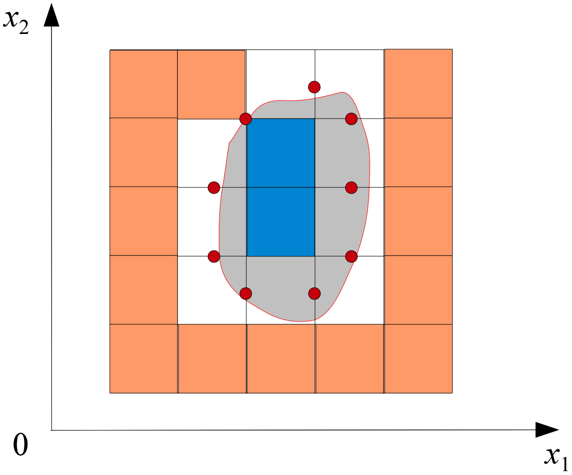
Refinement

- For each ?-grid cell: Refinement



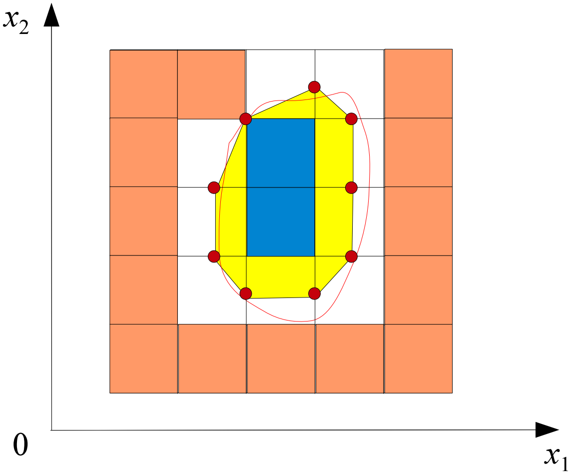
(j) Initial grid (k) 1^{st} refinement (l) 2^{nd} refinement

Finding polygon regions



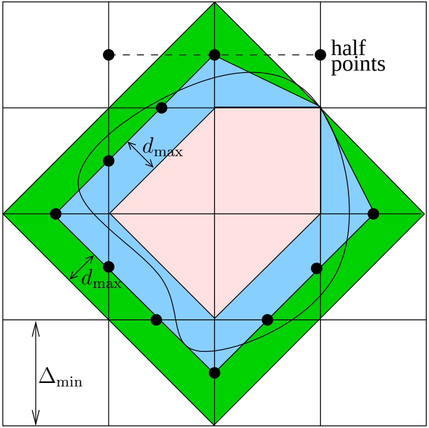
Connect middle points or grid points

Finding polygon regions



Connect middle points or grid points

Error bound



$$d_{\max} = \frac{\sqrt{2}}{4}\Delta_{\min}$$

Time complexity

- Assumption: c percent of "fresh" grid cells are refined in each round
- # rounds of refinement: $j = \log_2 \frac{\Delta}{\Delta_{\min}}$
- # instantiations in the initial grid: $\mathcal{O}(\hat{M} N_1 N_2)$

  $\hat{M}$: #non-constants in $\mathbf{R}^{\mathcal{X}}$; N_1, N_2: #grid points

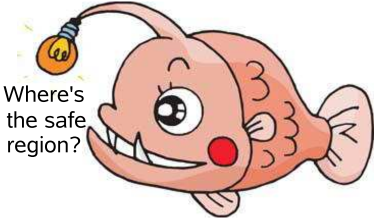
- Total # instantiations: $\mathcal{O}(\frac{c(1-c)^j}{1-c} \cdot \hat{M} N_1 N_2)$ (grid points)
- Each point (model checking) takes $\mathcal{O}(Mqt)$

  M: $|\mathbf{R}^{\mathcal{X}}|$, q: uniformization rate; t: time bound

$\Longrightarrow$ Total time cost: $\mathcal{O}(\frac{c(1-c)^j}{1-c} \cdot \hat{M} N_1 N_2 \cdot Mqt)$

Synthesis — Summary

- A first try on parameter synthesis in probabilistic setting
- Two algorithms for polygon approximation
- The non-symbolic algorithm can be generally applied



Analysis

Analysis

Specifications

	branching-time		linear-time	
logic				
automata				
	untimed	real-time	untimed	real-time

Analysis

How to model check $\overbrace{\text{CTMC}}^{\text{model}}$ against ?

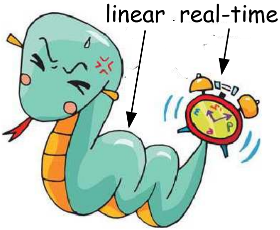
For CTMC model:

	branching-time		linear-time	
logic				
automata				
	untimed	real-time	untimed	real-time

Analysis

How to model check $\overbrace{\text{CTMC}}^{\text{model}}$ against $\overbrace{\text{linear real-time specification}}^{\text{specification}}$?

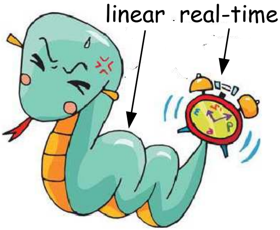
For CTMC model:



	branching-time		linear-time	
logic				
automata				Here!
	untimed	real-time	untimed	real-time

Analysis

How to model check $\overbrace{\text{CTMC}}^{\text{model}}$ against $\overbrace{\text{linear real-time specification}}^{\text{specification}}$?



For CTMC model:

	branching-time		linear-time	
logic	☺	☺	☺	
automata			☺	Here!
	untimed	real-time	untimed	real-time

deterministic timed automata (DTA)

Analysis

How to model check $\overbrace{\text{CTMC}}^{\text{model}}$ against $\overbrace{\text{linear real-time specification}}^{\text{specification}}$?

For CTMC model:

	branching-time		linear-time	
logic	☺	☺	☺	
automata			☺	Here!
	untimed	real-time	untimed	real-time

CTMC $\Longrightarrow$ probabilistic model checking $\Longleftarrow$ deterministic timed automata (DTA)

Main result

Given a CTMC $\mathcal{C}$ and a DTA $\mathcal{A}$, the model checking problem can be reduced to the reachability problem in piecewise deterministic Markov processes.

Conclusion

In the setting of probabilistic model checking,

- Diagnosis — counterexample generation
- Synthesis — parameter synthesis
- Analysis — CTMC vs. DTA model checking


köszönöm !תודה děkuji
mahalo 고맙습니다
thank you
merci 谢谢 danke
Ευχαριστώ شكرا
どうもありがとう gracias
Bedankt
2008
EDUCATOR OF
YEAR
PRINCIPAL