Counterexamples for Stochastic Model Checking

Husain Aljazzar

Chair for Software Engineering
University of Constance

Joint work with

- Holger Hermanns, University of Saarland
- Stefan Leue, University of Constance

„Counterexamples for Timed Probabilistic Reachability“

FORMATS 2005

Overview

Software Engineering

- Introduction
- (Directed) Explicit-State Reachability Analysis
- Directed Probabilistic Reachability Analysis
- Case Study and Experimental Results
- Future Work & Conclusion

Overview

Software Engineering

- Introduction
- (Directed) Explicit-State Reachability Analysis
- Directed Probabilistic Reachability Analysis
- Case Study and Experimental Results
- Future Work & Conclusion

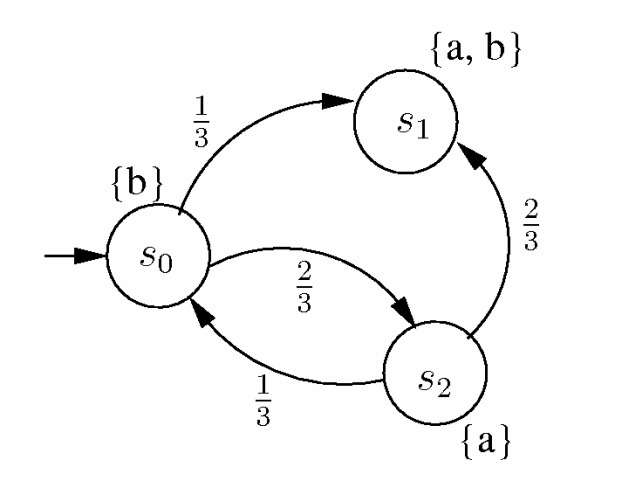
Motivation

- Stochastic models, e.g. *DTMC* and *CTMC*: performance and dependability analysis.
- A few model checking approaches for stochastic models have been presented.
- **Common weakness: Inability to give detailed debugging information (Counterexamples).**
- **Approach:** Use (Directed) Explicit-State Model Checking (ESMC/DESMC) in the reachability analysis of stochastic models to deliver counterexamples.

Stochastic Models

Software Engineering

- A DTMC is a quadruple (S, s_0, P, L), where
  - S is a finite set of states, and
  - $s_0 \in S$ is an initial state
  - $P : S \times S \to \mathbb{R}$ is the transition probability matrix,
  - $L : S \to 2^{AP}$ is labeling function.



- An *finite/ infinite run:*

$$s_0 \to s_1 \to s_2 \to \ldots \to s_n,$$

$$s_0 \to s_1 \to s_2 \to \ldots,$$

Overview

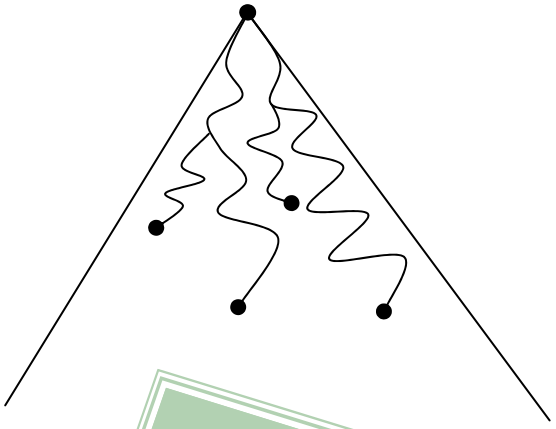
Software Engineering

- Introduction
- (Directed) Explicit-State Reachability Analysis
- Directed Probabilistic Reachability Analysis
- Case Study and Experimental Results
- Future Work & Conclusion

Explicit-State Model Checking (ESMC) -- Transition Systems

Software Engineering

- Explicit-State model checking (ESMC): exploring the state space using graph search algorithms like DFS and BFS.
- If an error is found, an offending system run is returned (Counterexample)
- What constitutes a *good* counterexample?
  - In typical non-stochastic transition systems: **good = short**
- How to obtain good (short) counterexamples?
  → Optimizing Search (Best First)
  - BFS
  - Directed Explicit-State Model Checking (**D**ESMC), i.e., Heuristic Search, e.g. Greedy Best First (GBestFS) or A*



Directed Explicit-State Model-Checking (DESMC) -- Transition Systems

Software Engineering

- Directed search algorithms use knowledge about
  - the state space or/and
  - the specification of the goal state
- A heuristic function ***h*** is used in the state evaluation**.**
- Advantages of DESMC: Improving the performance
  - Memory effort
  - Runtime

Overview

Software Engineering

- Introduction
- (Directed) Explicit-State Reachability Analysis
- Directed Probabilistic Reachability Analysis
- Case Study and Experimental Results
- Future Work & Conclusion

Counterexamples for Stochastic Models

Software Engineering

- Use ESMC or DESMC on stochastic models
- What is a good counterexample in stochastic models?
  - A counterexample which carries a high probability mass (more informative).
  - **The length of a run is not indicative of its probability mass.**
  - → Timed run probability

Timed Run Probability γ

Software Engineering

- Let $r = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots \rightarrow s_n$ be a run.

- The timed run probability of ***r***, $\gamma(r, k)$, is the probability to execute ***r*** within at most ***k*** time units.

$$\gamma(r,k) = P(s_{n-1}, s_n) \cdot \sum_{i=0}^{k-1} \pi(s_{n-1}, i)$$

Note: For CTMCs it is more complicated

The determination of the timed run probability is computationally very expensive.

→ An approximation based on Uniformisation of the model.

ESMC and DESMC for Stochastic Models

Software Engineering

Idea: Use of optimizing algorithms with the timed run probability as optimization criterion!

- **Dijkstra, (ESMC)**
- **GBestFS (DESMC)**
- **Z* (DESMC)**

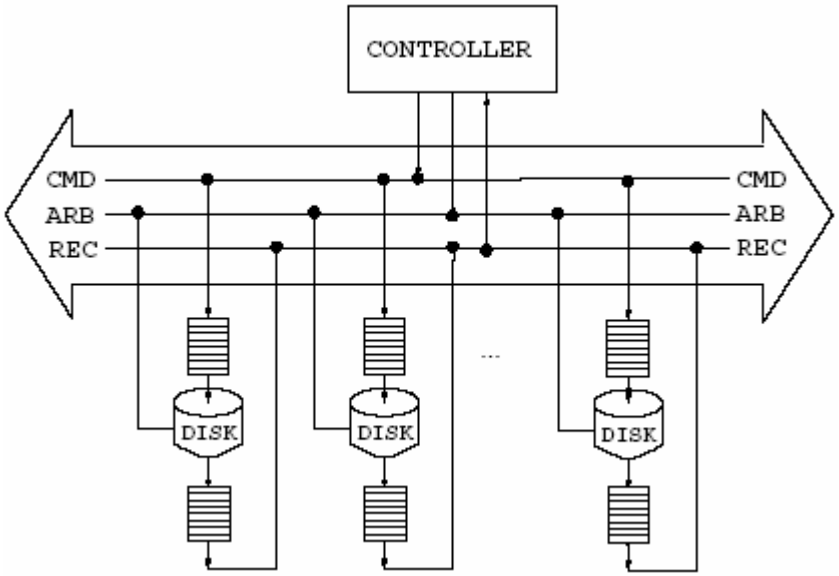
Overview

Software Engineering

- Introduction
- (Directed) Explicit-State Reachability Analysis
- Directed Probabilistic Reachability Analysis
- Case Study and Experimental Results
- Future Work
- Conclusion

Case-Study: SCSI-2-Protocol

- In our experiments:
  - One Controller
  - One main disk (frequently used)
  - Two backup disks (rarely used)
- The system was modeled in LOTOS and transformed into an interactive Markov chain (IMC) by the CADP toolbox.



SCSI-2-Protocol: A Timed Reachability Property

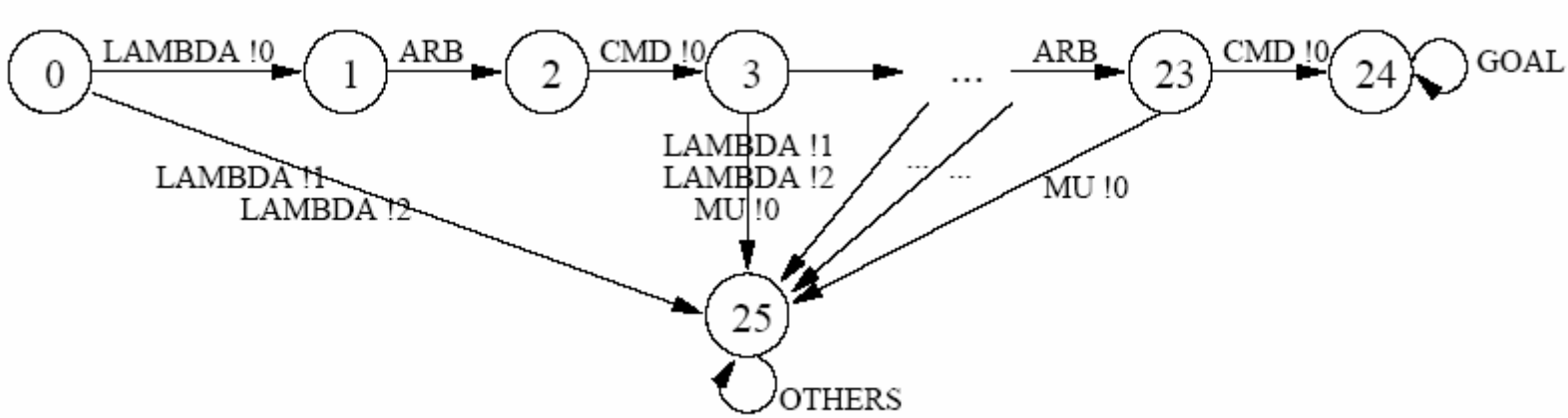
Software Engineering

- Main disk overload (MDOL): The main disk is overloaded while the backup disks are not accessed.

- The probability to reach a MDOL state within ***t*** time units does not exceed 0.3.

SCSI-2-Protocol: Counterexample

- The counterexample delivered by Z*



SCSI-Protocol: Experimental Results

Software Engineering

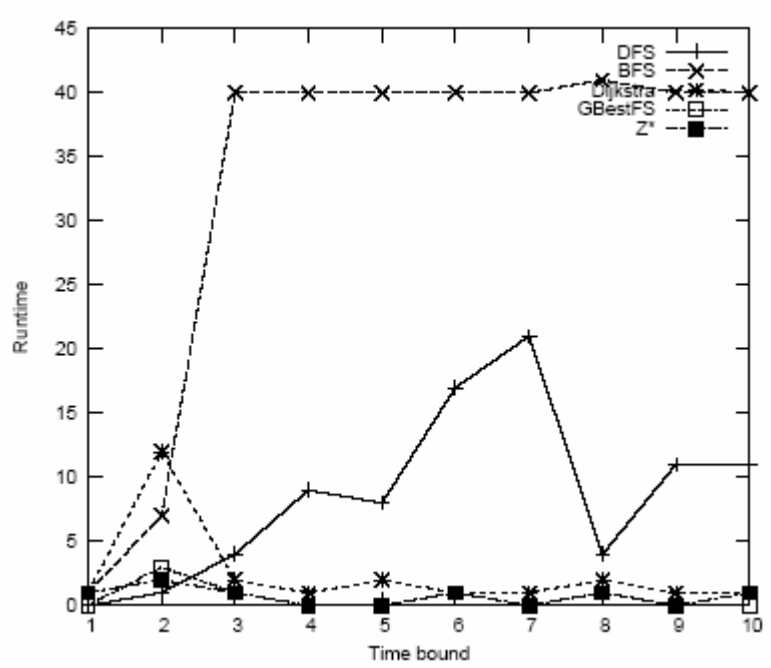
- For $t \in \{1, 2, \ldots, 10\}$

Time bound	1	2	3	4	5	6	7	8	9	10
Model	0.235	0.312	0.327	0.329	0.329	0.329	0.330	0.330	0.330	0.330
DFS	-	-	-	-	-	-	0.000	-	-	0.000
BFS	-	0.161	0.161	0.161	0.161	0.161	0.161	0.161	0.161	0.161
Dijkstra	-	0.161	0.161	0.161	0.161	0.161	0.161	0.161	0.161	0.161
GBestFS	-	0.012	0.012	0.012	0.012	0.012	0.012	0.012	0.012	0.012
Z^*	-	0.161	0.161	0.161	0.161	0.161	0.161	0.161	0.161	0.161

SCSI-Protocol: Experimental Results

Software Engineering

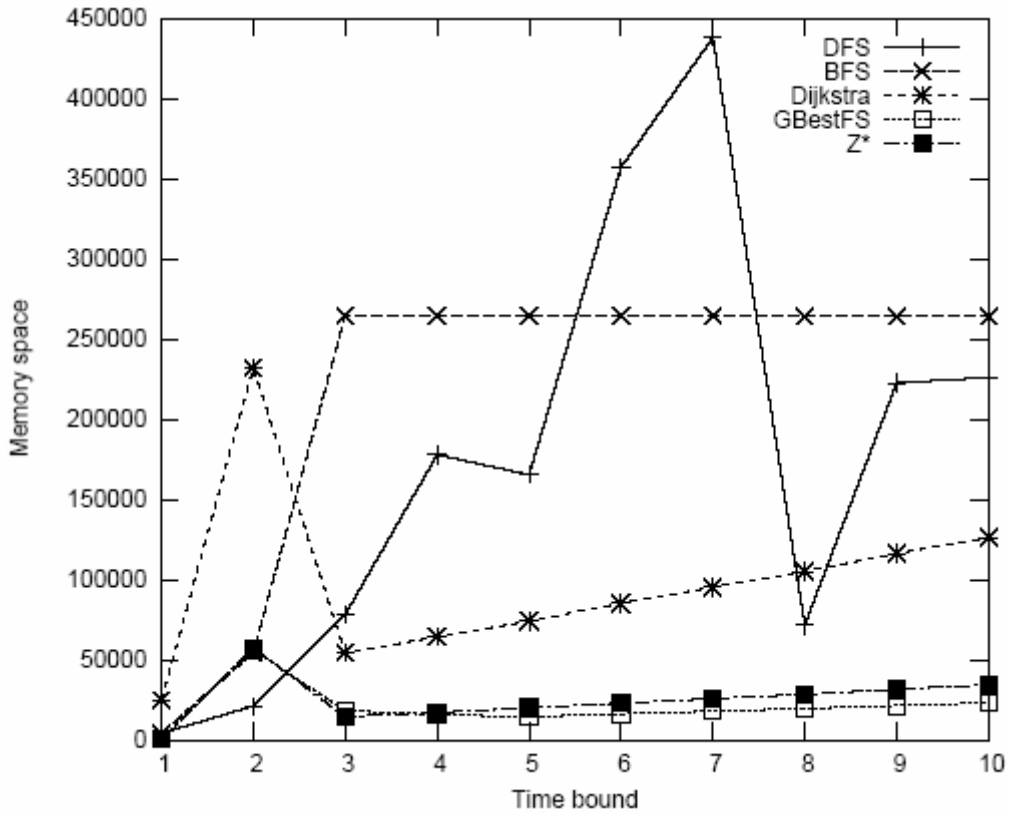
- Runtime
  - BFS and DFS do not scale to large models.
  - Good runtime behavior of Dijkstra, GBestFS, Z*
  - Directed algorithms GBestFS and Z* have the best runtime performance.



SCSI-Protocol: Experimental Results

Software Engineering

- Memory effort
  - The behavior of DFS and BFS is unacceptable.
  - Dijkstra does not scale to large models
  - Z* and GBestFS bring significant improvement
  - GBestFS has the best behavior.



Overview

Software Engineering

- Introduction
- (Directed) Explicit-State Reachability Analysis
- Directed Probabilistic Reachability Analysis
- Case Study and Experimental Results
- Future Work & Conclusion

Future Work

- More case studies
- Finding more than one path (counterexample = offending tree)
- Visualization of counterexamples
- General heuristics
- Non-Determinism (CT Markov Decision Processes)

Conclusion

Software Engineering

- Novel approach to generate counterexamples for timed probabilistic reachability analysis.
- Heuristic guided
- Good experimental results
- A good step in the right direction

Thanks for your attention!