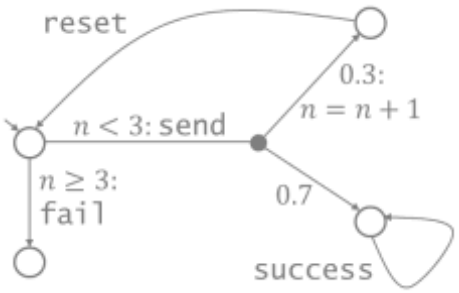
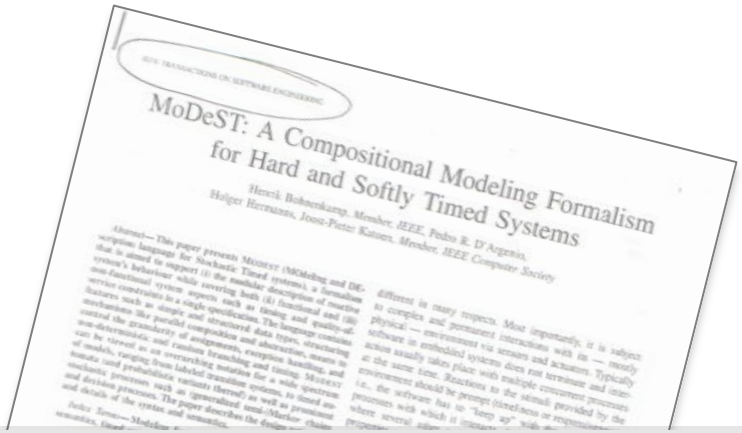
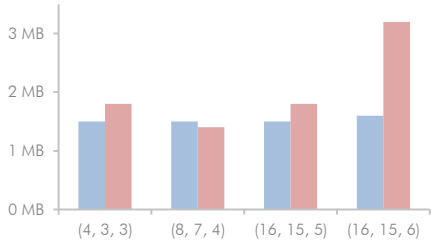


Partial Order Methods for Statistical Model Checking and Simulation

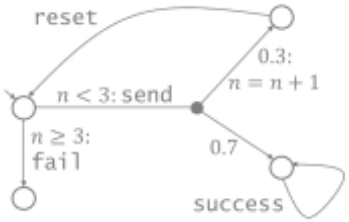

MoDeST: A Compositional Modeling Formalism for Hard and Softly Timed Systems



Jonathan Bogdoll, Luis María Ferrer Fioriti, **Arnd Hartmanns**, Holger Hermanns
Saarland University, Saarbrücken, Germany

Outline

1 The **model** The **analysis techniques**



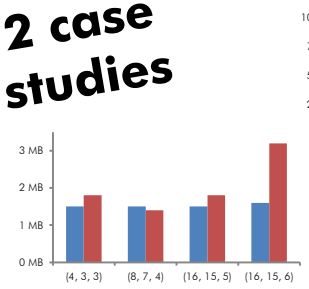
2

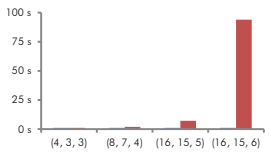
3

Idea + Solution

4 Evaluation

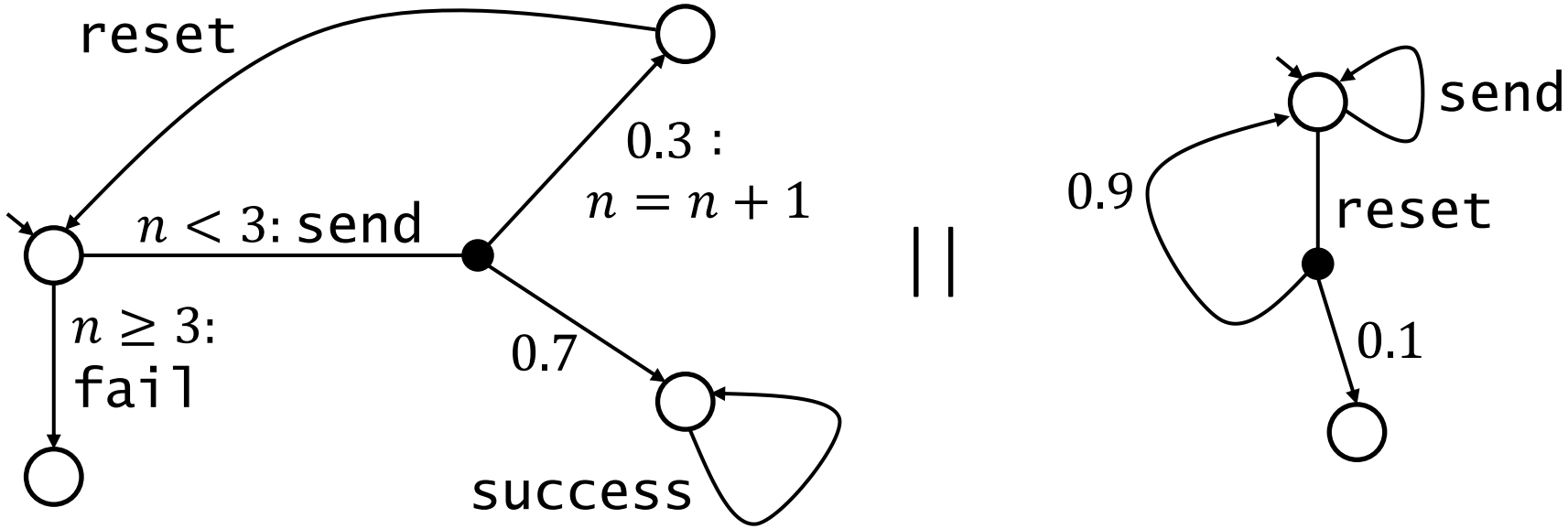
2 case studies





The Model

Networks of probabilistic automata with variables **PA/MDP**



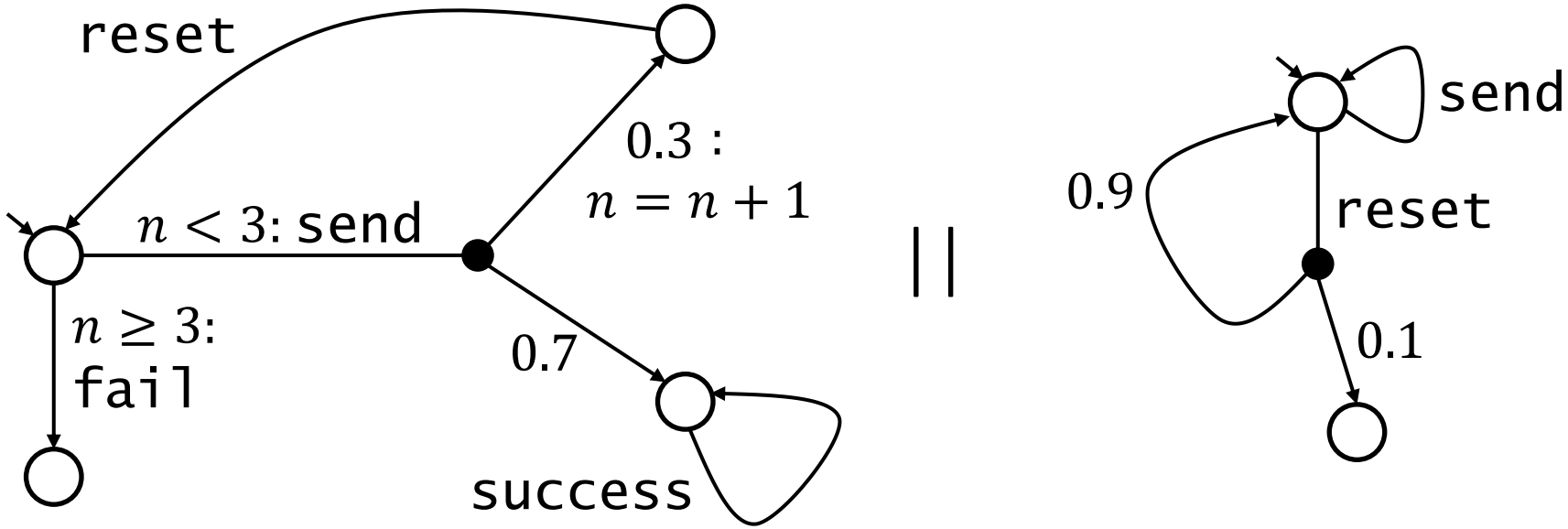
$P_{\leq 10}(\text{success}) > 0.9$? $P_{\leq 10}(\text{deadlock}) \leq 0.05$?

Synchronisation, discrete variables (possibly shared)

Transitions lead to distributions: $\rightarrow \subseteq S \times \Sigma \times \text{Dist}(S)$

The Analysis Techniques

Simulation: exploring concrete paths through the model



$P_{\leq 10}(\text{success}) > 0.9$?

yes yes yes no ⇨ **no?**

$P_{\leq 10}(\text{deadlock}) \leq 0.05$?

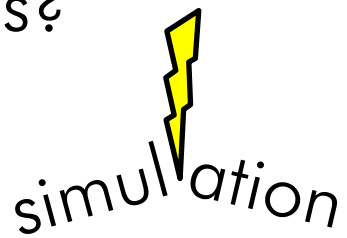
how many runs needed?

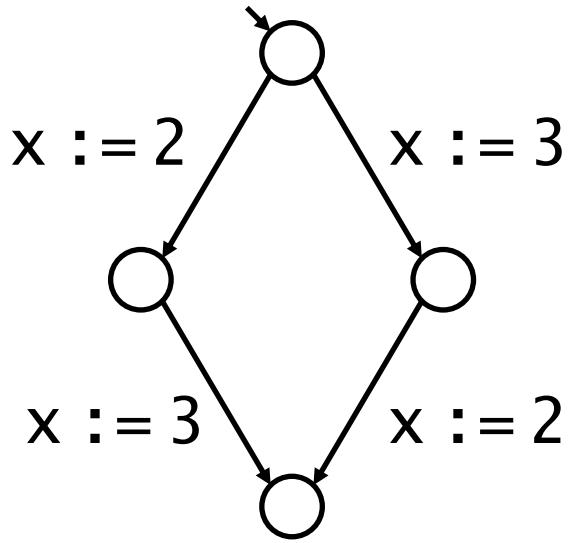
+ advanced statistics: Statistical Model Checking **SMC**

The Problem

PA can be nondeterministic

⇨ "paths": trees?





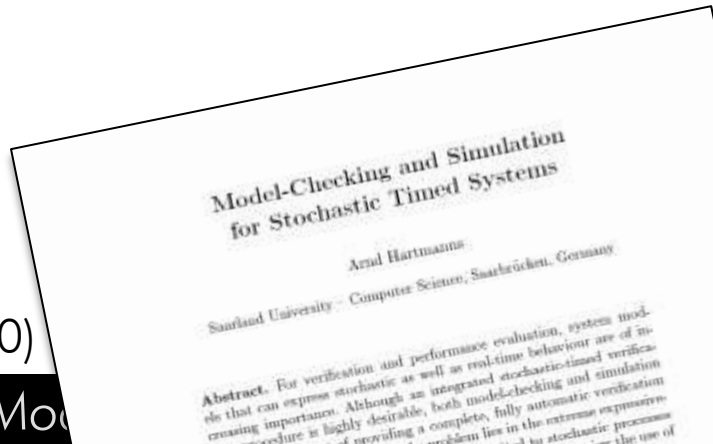
Standard solution: simulate anyway

⇨ implicit *scheduler* resolves nondeterminisn **bad idea**

$P(x \geq 3)$ $P(x \geq 3) = 0$

...

$P(x \geq 3) = 1$ **???**


Model-Checking and Simulation for Stochastic Timed Systems

Arnd Hartmanns

Saarland University – Computer Science, Saarbrücken, Germany

Abstract. For verification and performance evaluation, system models that can express stochastic as well as real-time behaviour are of increasing importance. Although an integrated stochastic-timed verification is highly desirable, both model-checking and simulation provide a complete, fully automatic verification

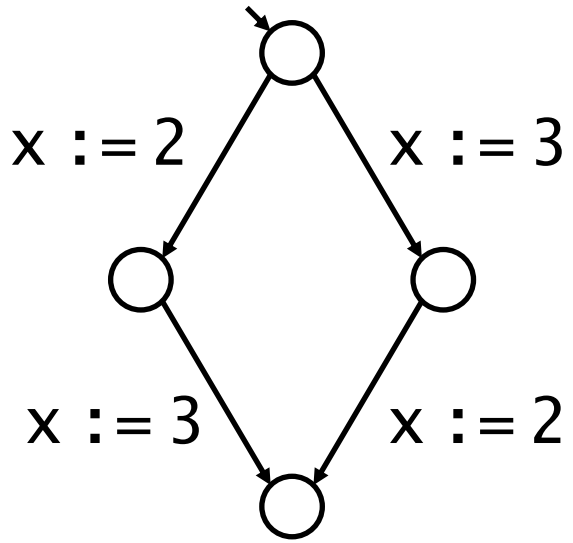
more motivation and examples: see H. (FMCO 2010)

The Idea

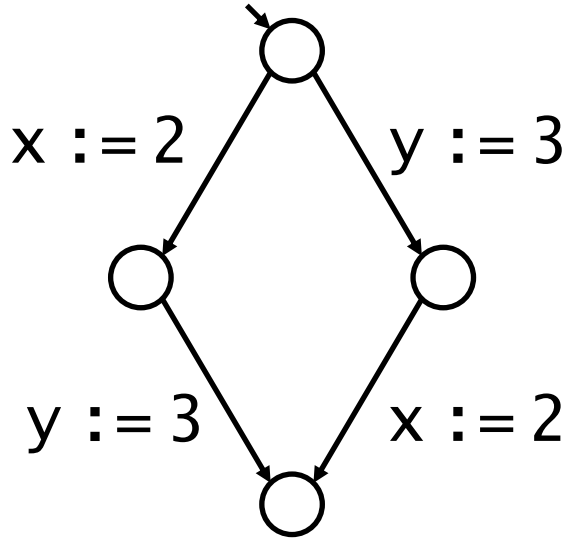
Nondeterminism may be spurious

= irrelevant for the results

try to detect spuriousness
on-the-fly and ignore
using methods derived
from partial order reduction



$P(x \geq 3) = 0$! ⇦

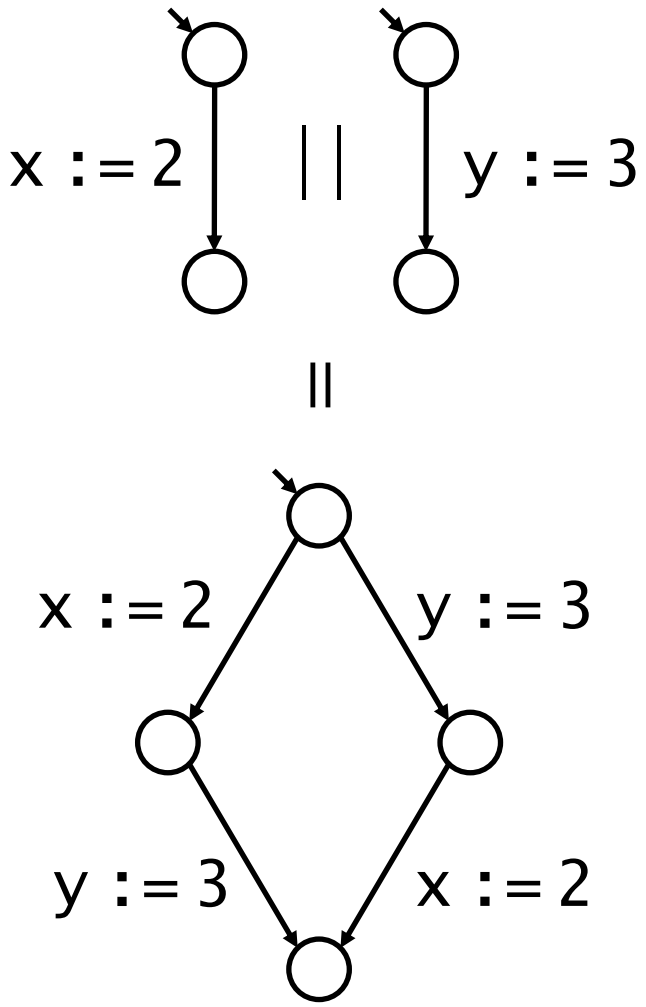


Partial order reduction [1]

Speed up model checking by removing spurious paths

Implemented on-the-fly during state-space generation **e.g. in SPIN**

This looks *precisely* like what we want!



Limitation:
In practice, can only deal with spurious interleavings

[1] Developed concurrently by Godefroid, Peled and Valmari

Detecting spuriousness on-the-fly via ample sets[1]

A1 If $ample(s) \neq T(s)$, then all its transitions are invisible

A2 On all original paths, transitions in ample(s) must occur before transitions *dependent* on ample(s)

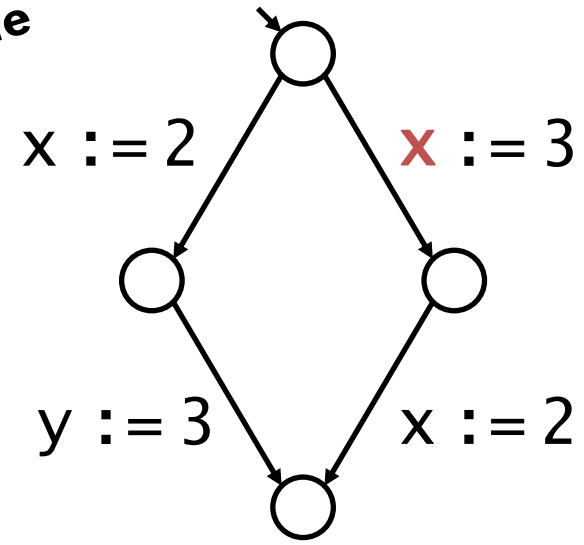
A3 Every end component has a state s such that $ample(s) = T(s)$

A4 If $ample(s) \neq T(s)$, then $|ample(s)| = 1$

need to bound lookahead

over-approximate

= determinism!



[1] D. Peled (CAV 1994) / Baier, D'Argenio, Größer (ENTCS 2006)

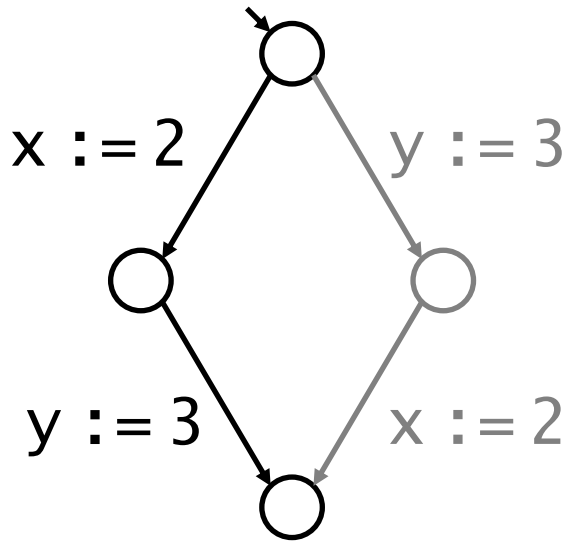
Detecting spuriousness on-the-fly via ample sets

A1 If ample(s) ≠ T(s), then all its transitions are invisible

A2′ All original paths have a prefix of length ≤ k on which a transition in ample(s) occurs before transitions…

A3′ Of the last l states, there is at least one with ample(s) = T(s)

A4′ If ample(s) ≠ T(s), then |ample(s)| = 1 or T(s) = ∅



⇨ Two parameters: k and l

k_{min}: minimal k necessary

On-the-fly partial order simulation algorithm

User need not worry about k and l:

$s :=$ initial state
do
  $F :=$ null
  $T := Succ(s)$ (succ. *distributions*)
  if $|T| = 1$, then $F := T$
  else for each singleton $A \subseteq T$:
    if s is a valid ample set, then $F := A$
  if $F =$ null, abort simulation
  else $s :=$ sample(F)
until termination criterion satisfied

On abort because k or l exceeded, error message gives precise reason

The Solution

Implementation

Implemented in modes,
the discrete-event simulator for the Modest[1] language

www.modestchecker.net

Part of the Modest toolset

- modes: Simulation
- mcpta: Model checking
- mime: Graphical interface


MoDeST: A Compositional Modeling Formalism for Hard and Softly Timed Systems

Henrik Bohnenkamp, Member, IEEE, Pedro R. D'Argenio,
Holger Hermanns, Joost-Pieter Katoen, Member, IEEE Computer Society

[1] Bohnenkamp, D'Argenio, Hermanns, Katoen (IEEE TSE 2006)

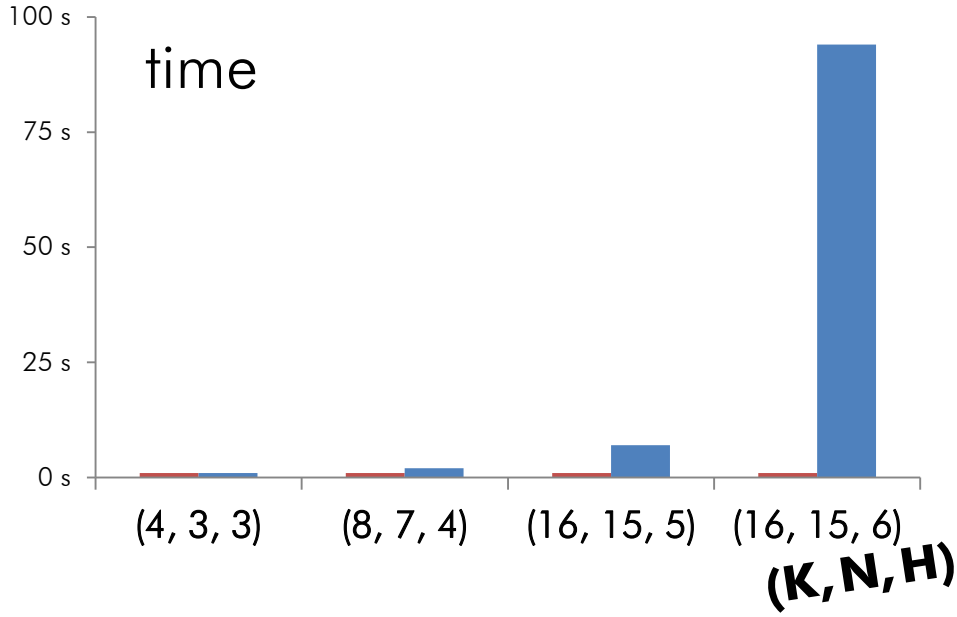
The Solution

Case studies

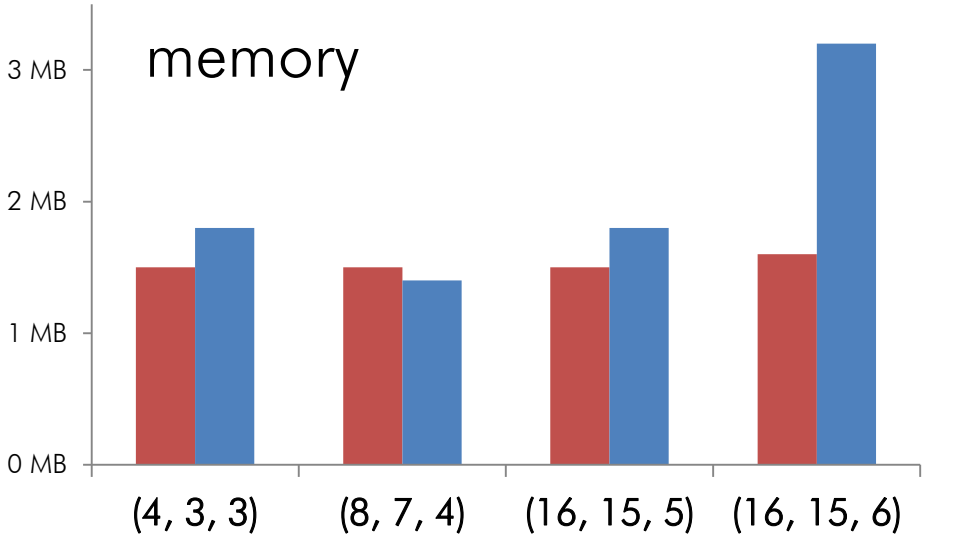
IEEE 802.3 BEB protocol

implicit-scheduler vs.
partial-order simulation

k_{min} = H (number of hosts)



prototype implementation



Model checking: infeasible
> 100 GB for (16, 15, 5)

Case studies

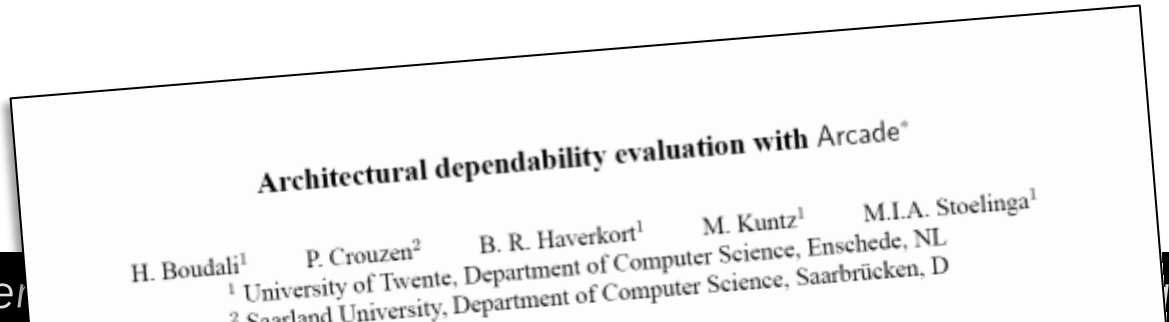
ARCADE[1] dependability evaluation models

Stochastic timed systems, lots of component automata

↳ *orthogonal to our method*

model	||	t	implicit-sched.	partial-order	l	k_{min}
1bc-1ruded	3	100	1s / 1.3MB	1s / 1.5MB	16	2
2bc-1rufcfs	4	100	2s / 1.1MB	4s / 1.1MB	16	3
dda-scaled	41	15	41s / 1.5MB	379s / 2.6MB	16	6
rcs-scaled	36	15	24s / 1.6MB	132s / 2.0MB	16	4

[1] Boudali, Crouzen, Haverkort, Kuntz, Stoëlinga (DSN 2008)


Architectural dependability evaluation with Arcade*

H. Boudali[1] P. Crouzen[2] B. R. Haverkort[1] M. Kuntz[1] M.I.A. Stoelinga[1]

[1] University of Twente, Department of Computer Science, Enschede, NL

[2] Saarland University, Department of Computer Science, Saarbrücken, D

The Conclusion

Simulation for nondeterministic models **PA**

Previously: Impossible – or wrong results **implicit scheduler**

Now: Possible – for spurious interleavings  **partial order reduction**

– Time and memory penalty

+ Applies to any simulation-based approach **statistical model checking**

"A little bit of on-the-fly model checking during simulation"

Implementation (prototype) available at

`www.modestchecker.net`