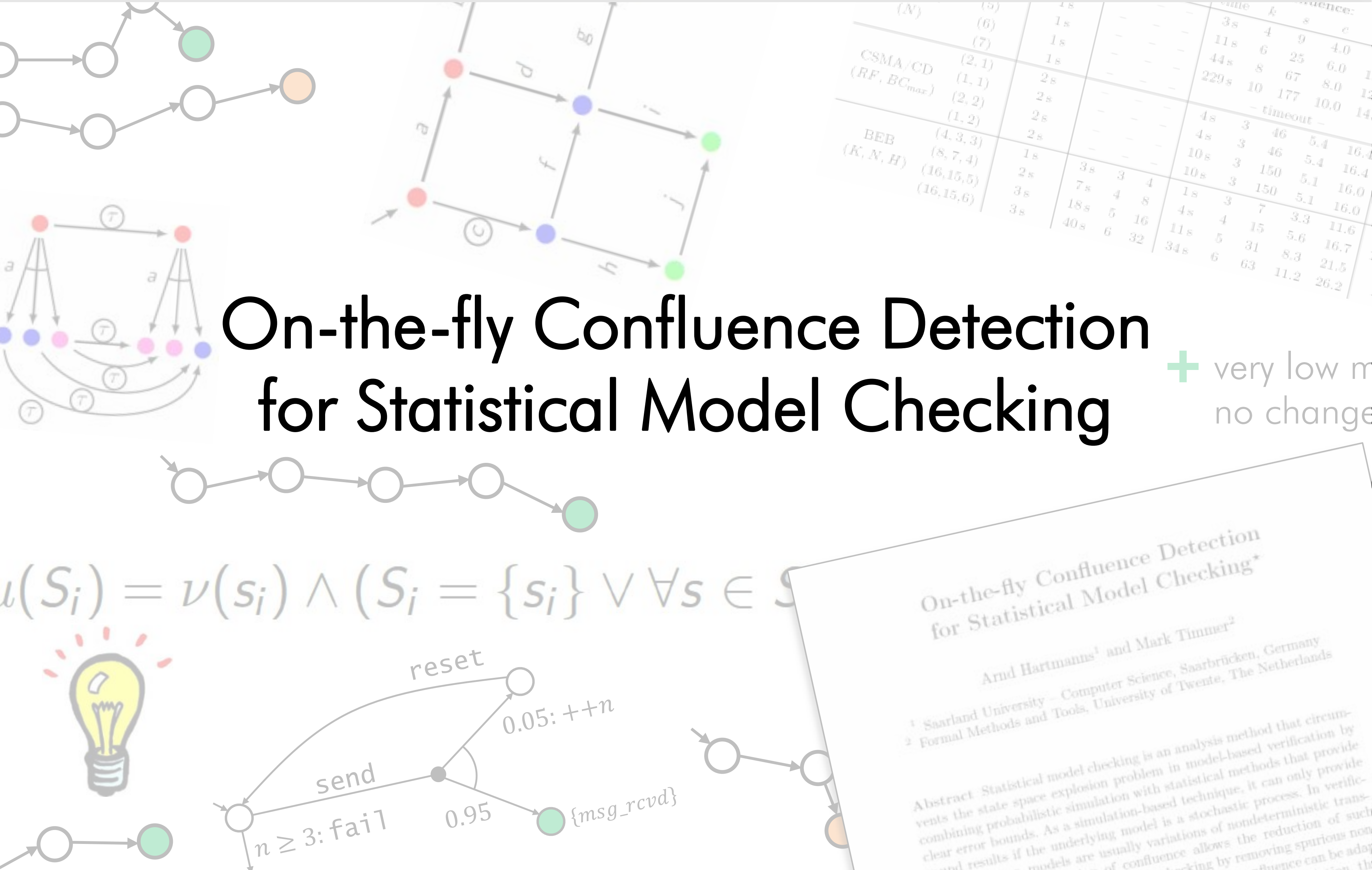
NFM 2013, MOFFETT FIELD, CA, USA

On-the-fly Confluence Detection for Statistical Model Checking



Arnd Hartmanns and Mark Timmer
Saarland University, Germany
University of Twente, The Netherlands

Model Checking

$\lozenge \, msg\_rcvd$?

LTS Model ⟷ model checking ⟷ Requirements

model-based testing

WANTED

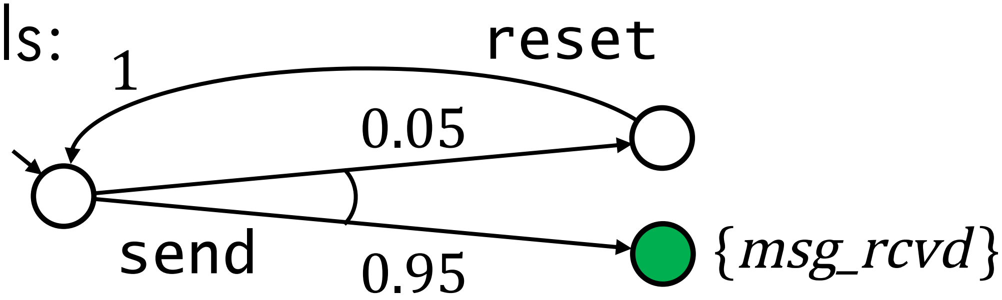
System/implementation

Problem: State-space explosion **memory consumption**

Probabilistic Model Checking

$P(\lozenge \, msg\_rcvd) \geq 0.99$?

DTMC models:

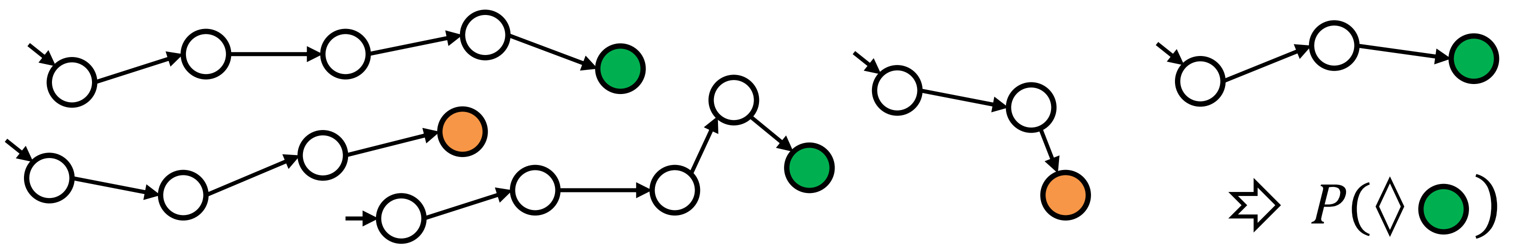


runtime
stability

Problem: State-space explosion plus numerical complexity

Statistical Model Checking [SMC]

SMC = Simulation + Statistics



$\Rightarrow P(\lozenge \bullet) \approx 0.6$

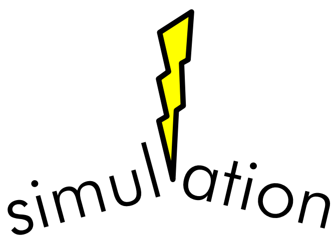
...confidence intervals, Chernoff-Hoeffding bound, SPRT...

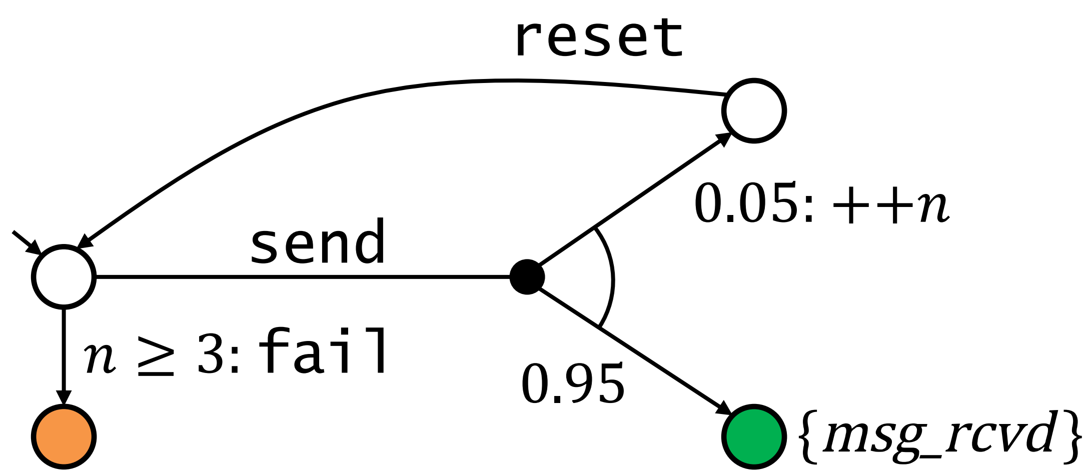
error bounds: e.g. result is ε-correct with probability δ

- **+** constant memory usage (store only current state)
  no numeric surprises (e.g. with imprecise arithmetics)
- **–** runtime strongly dependent on desired accuracy

Statistical Model Checking **versus Nondeterminism**

MDP**/PA** models:





⇨ need to resolve nondeterministic choices

$P_{\min}(\Diamond\, msg\_rcvd) \geq 0.99$?

$P_{\max}(\Diamond\, msg\_rcvd) \geq 1$?

Standard technique:

☹ implicit uniformly distributed resolution

⇨ some value $\in [P_{\min}, P_{\max}]$

widely implemented: PRISM, UPPAAL, ...

Previous approaches to SMC for MDPs

Partial order reduction-based method: "POR"

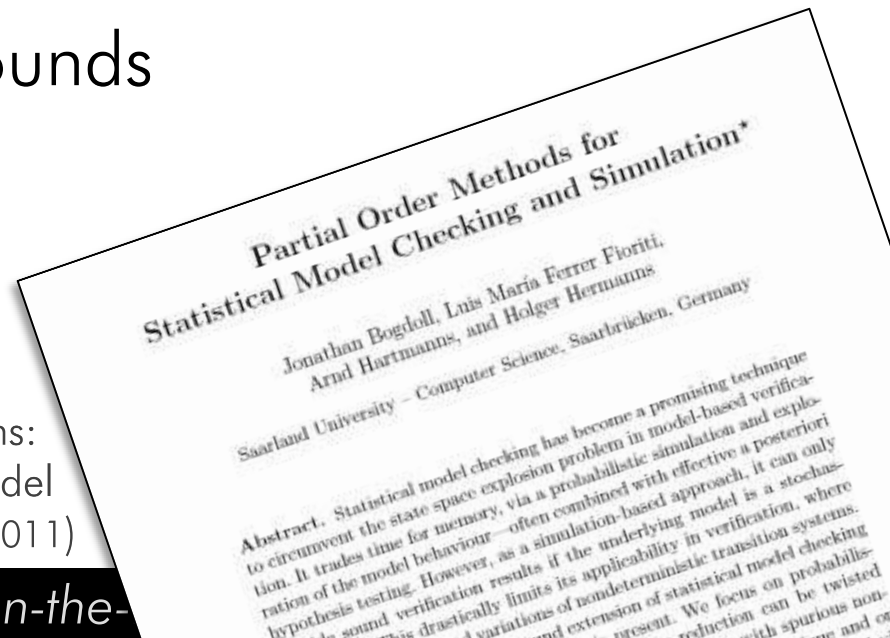
Nondeterminism may be **spurious**
= irrelevant for the results, i.e. $P_{\min} = P_{\max}$

⇨ check for spuriousness on-the-fly and ignore

\+ very low memory overhead
no change to SMC error bounds

– spurious interleavings only

Bogdoll, Ferrer Fioriti, H., Hermanns:
Partial Order Methods for Statistical Model
Checking and Simulation (FMOODS/FORTE 2011)



Partial Order Methods for
Statistical Model Checking and Simulation*

Jonathan Bogdoll, Luis María Ferrer Fioriti,
Arnd Hartmanns, and Holger Hermanns

Saarland University – Computer Science, Saarbrücken, Germany

Abstract. Statistical model checking has become a promising technique to circumvent the state space explosion problem in model-based verification. It trades time for memory, via a probabilistic simulation and exploration of the model behaviour—often combined with effective a posteriori hypothesis testing. However, as a simulation-based approach, it can only

Previous approaches to SMC for MDPs

Learning-based method:

Use reinforcement learning to obtain memoryless scheduler using simulation

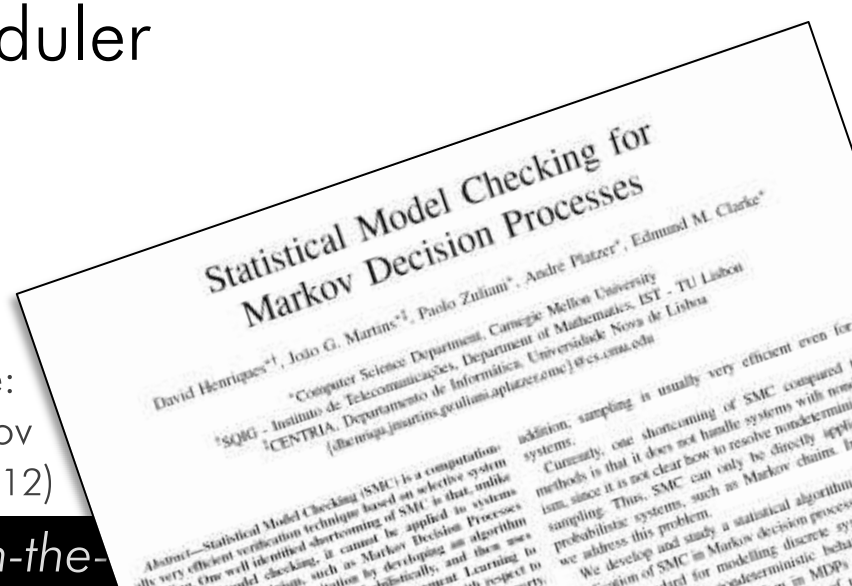
technique from AI

⇨ use that scheduler for SMC for $P_{\max}$(bounded LTL)

\+ works for every MDP

− memory usage to store scheduler
no error bounds, converges
to actual result only

Henriques, Martins, Zuliani, Platzer, Clarke:
Statistical Model Checking for Markov
Decision Processes (QEST 2012)


Statistical Model Checking for Markov Decision Processes

David Henriques, João G. Martins, Paolo Zuliani, André Platzer, Edmund M. Clarke

Outline

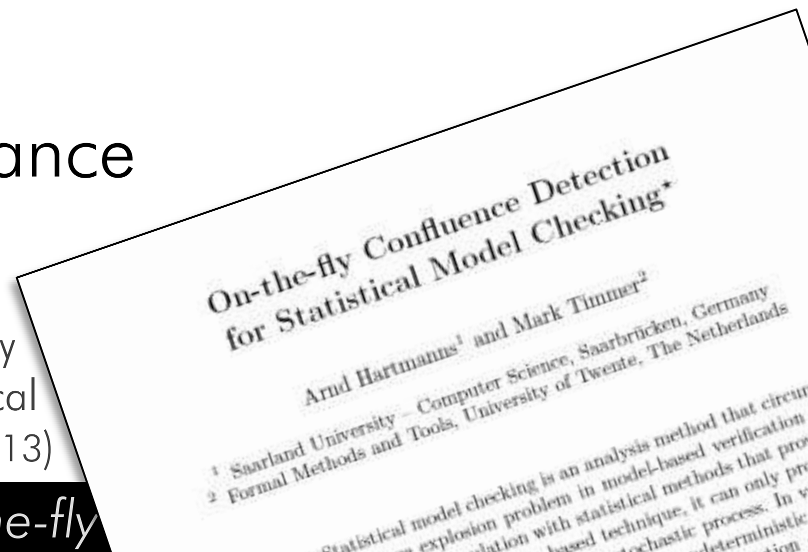
In this talk: a new method
based on **on-the-fly confluence detection** 

1 Probabilistic Confluence (MT)
Adaption to SMC & advantages over POR

2 On-the-fly Detection (MT)
A correct algorithm for use during simulation

3 Evaluation
Tools, applicability, performance

Hartmanns, Timmer: On-the-fly Confluence Detection for Statistical Model Checking (NFM 2013)


On-the-fly Confluence Detection for Statistical Model Checking*

Arnd Hartmanns[1] and Mark Timmer[2]

[1] Saarland University – Computer Science, Saarbrücken, Germany
[2] Formal Methods and Tools, University of Twente, The Netherlands

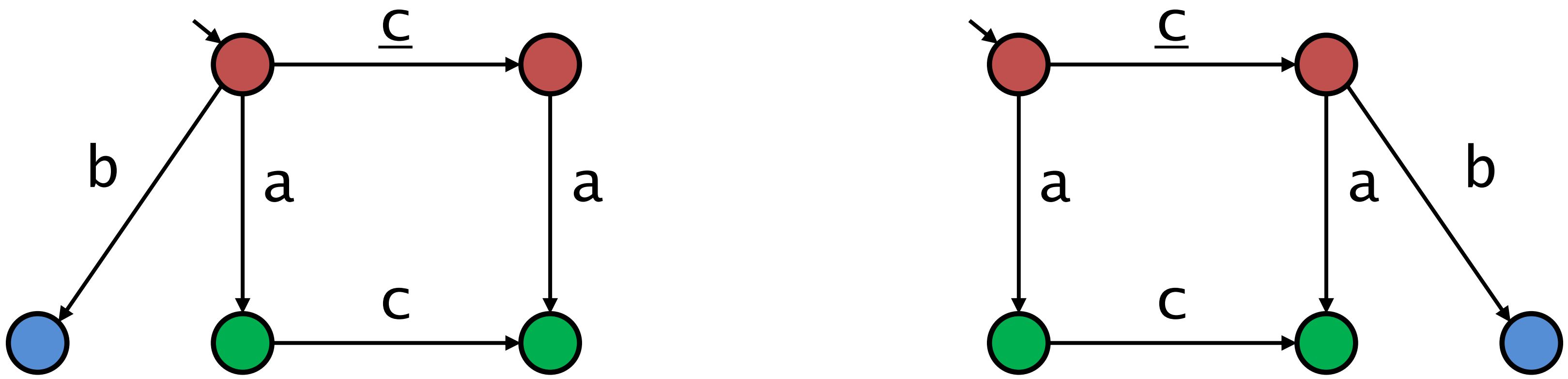
Confluence

Transitions can sometimes be given priority:

- Stuttering
- Nonprobabilistic

Just like for POR

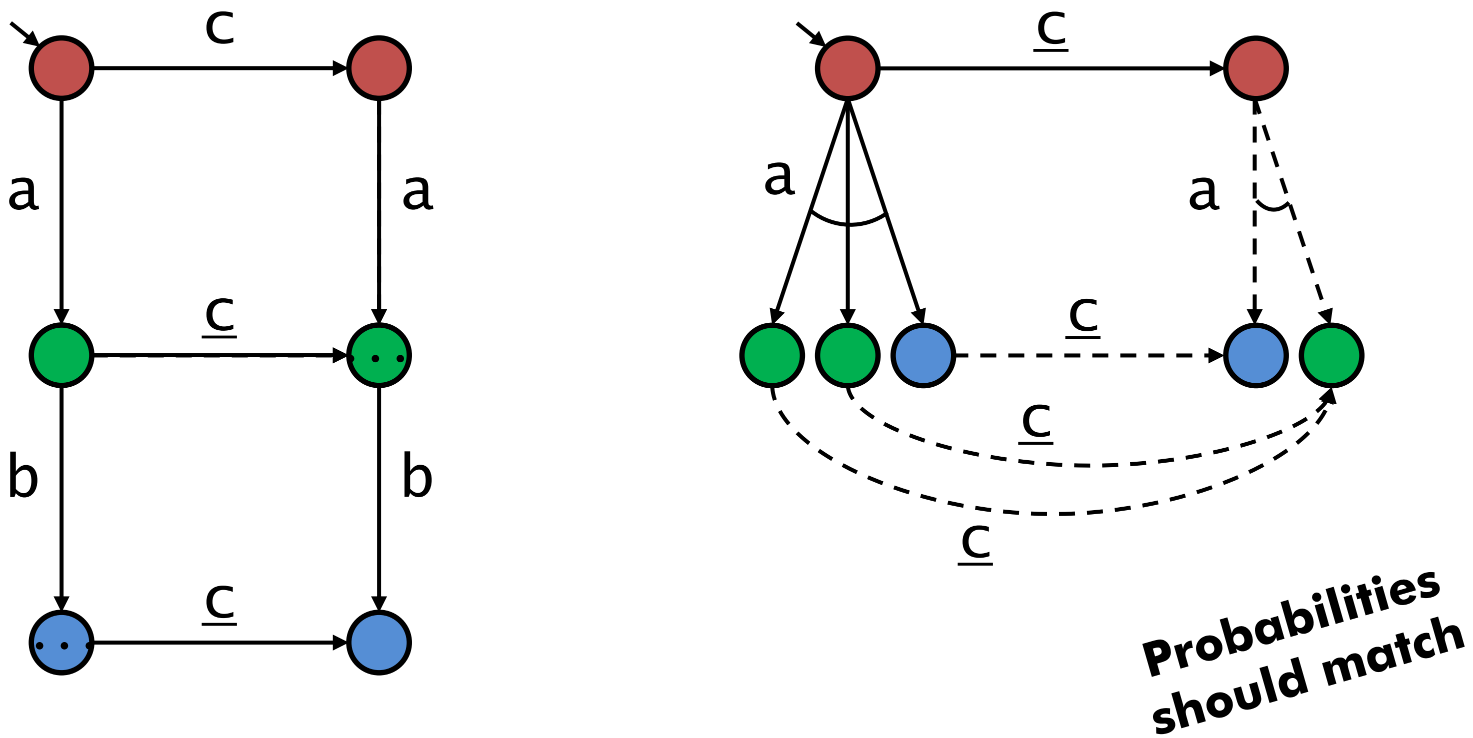
Invisible transitions may *disable behaviour*…
though often they *connect equivalent states*



Confluence

How to be sure?

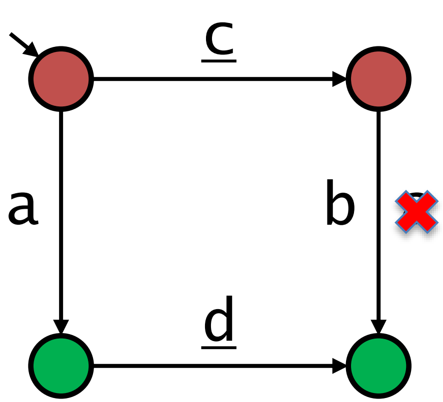
⇨ Check *confluence diagram* for a set of transitions

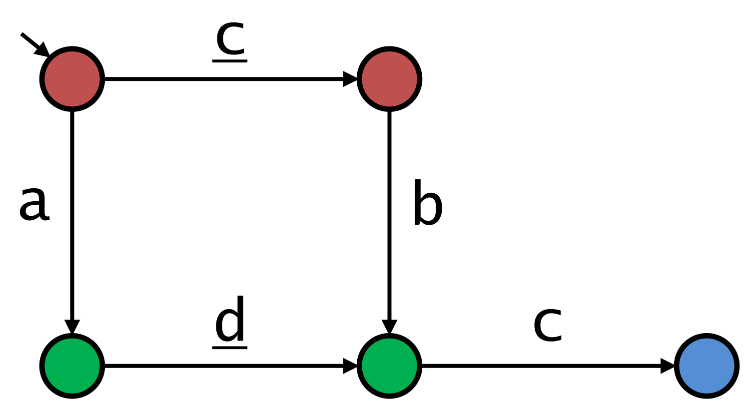


Confluence for SMC

We relax a previous notion in three ways:

1. Transitions may be mimicked by *different actions*
2. Transitions have to be stuttering and nonprobabilistic *only locally*
3. Distributions may be related in a *more liberal* way





Confluent transitions can still be given priority while preserving all properties in $\mathbf{PCTL}^*_{-X}$

Confluence versus POR

Partial Order Reduction:

- Preserves *probabilistic* LTL_{-X}
- Based on *independent actions* and *ample sets*
- Allows ample actions to be *probabilistic*

Advantage: can prioritise probabilistic transitions
Disadvantage: not defined for concrete state spaces

reduction powers incomparable

Confluence Reduction:

- Preserves $PCTL^{*}_{-X}$
- Based on *confluent transitions (commuting diagrams)*

Advantage: defined for concrete state spaces
Disadvantage: cannot prioritise probabilistic transitions

On-the-fly detection

Simulation / SMC using on-the-fly confluence detection:

Upon arrival at a nondeterministic state:

Look for at least one outgoing confluent transition

- If no such transition is found, abort **(or try POR)**
- If at least one transition is found, take it

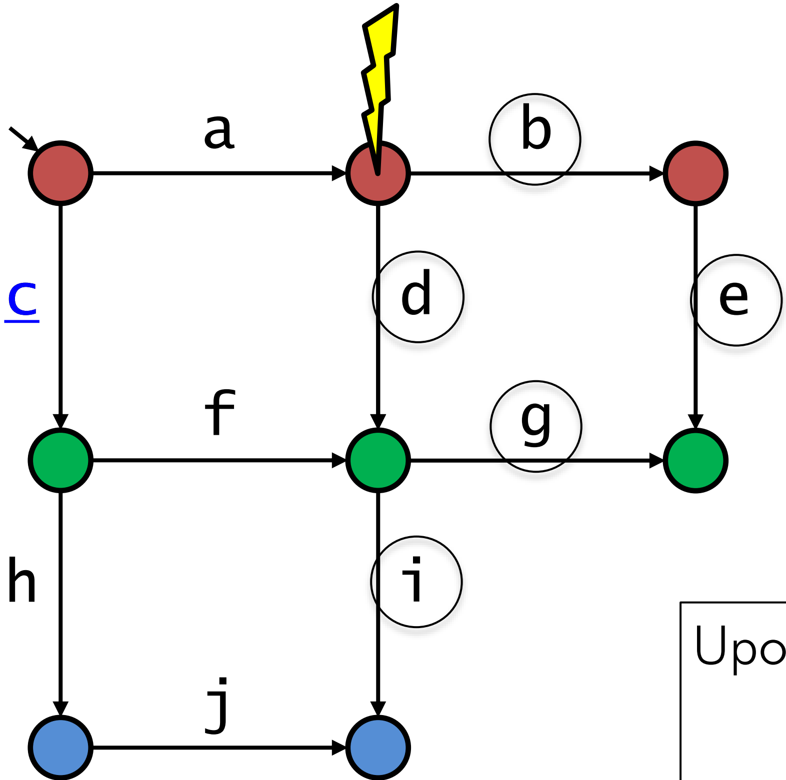
1. Check if it is *nonprobabilistic* and *stuttering*
2. Check if all its neighbouring transitions are mimicked

(recursion)

Careful: ignoring problem

⇨ Check if at least every *l* steps a state is *fully explored*

On the fly detection



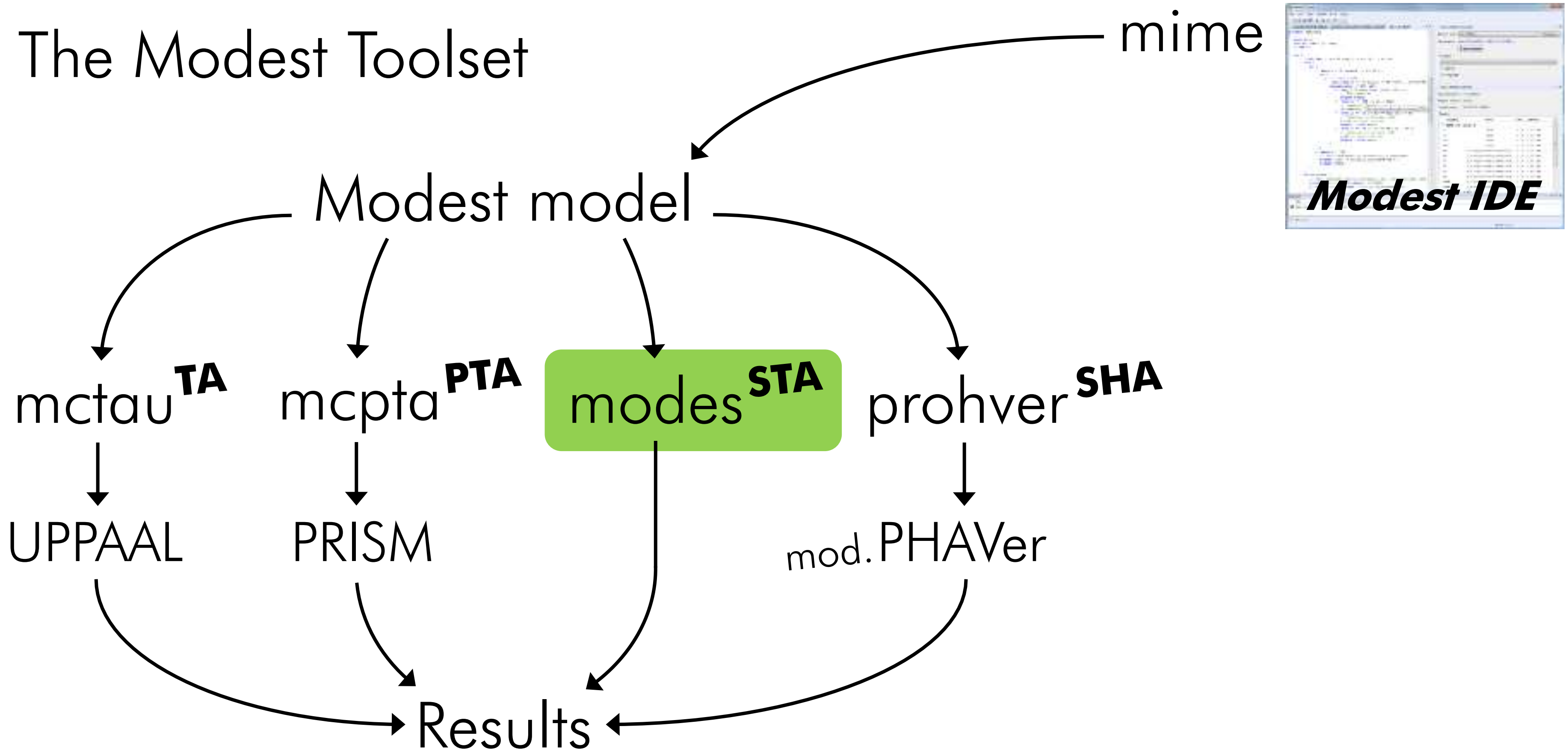
- Check if it is nonprobabilistic and stuttering
- Check if all its neighbouring transitions are mimicked

Upon arrival at a nondeterministic state:

- Look for at least one outgoing confluent transition
  - If no such transition is found, abort
  - If at least one transition is found, take it

Tool Support

The Modest Toolset



modes: SMC for Modest/STA

⇨ MDP as special case

POR + **Confluence** NEW

www.modestchecker.net

Evaluation

Examples

Example	POR	confluence
Dining Cryptographers (**PRISM model**) N cryptographers, two neighbours each Nondeterminism: communication order	✗	✓
CSMA/CD (**"DPTA"**) Two senders, one shared channel, collisions Nondeterministic choice of station inside channel	✗	✓
BEB (Bounded Exponential Backoff) Detailed MDP model of exponential backoff K: max. backoff, N: n° of retries, H: n° of hosts **huge state space**	✓	✓

Evaluation

Results

reachability properties

(10000 runs → $\varepsilon < 0.01, \delta > 0.98$)

model	params	uniform: time	partial order: time	partial order: k	partial order: s	**confluence**: time	**confluence**: k	**confluence**: s	**confluence**: c	**confluence**: t	model checking: states	model checking: time
dining cryptographers (N)	(3)	1 s	–	–	–	3 s	4	9	4.0	8.0	609	1 s
	(4)	1 s	–	–	–	11 s	6	25	6.0	10.0	3 841	2 s
	(5)	1 s	–	–	–	44 s	8	67	8.0	12.0	23 809	7 s
	(6)	1 s	–	–	–	229 s	10	177	10.0	14.0	144 705	26 s
	(7)	1 s	–	–	–	– timeout –					864 257	80 s
CSMA/CD (RF, BC_{max})	(2, 1)	2 s	–	–	–	4 s	3	46	5.4	16.4	15 283	11 s
	(1, 1)	2 s	–	–	–	4 s	3	46	5.4	16.4	30 256	49 s
	(2, 2)	2 s	–	–	–	10 s	3	150	5.1	16.0	98 533	52 s
	(1, 2)	2 s	–	–	–	10 s	3	150	5.1	16.0	194 818	208 s
BEB (K, N, H)	(4, 3, 3)	1 s	3 s	3	4	1 s	3	7	3.3	11.6	$> 10^3$	> 0 s
	(8, 7, 4)	2 s	7 s	4	8	4 s	4	15	5.6	16.7	$> 10^7$	> 7 s
	(16,15,5)	3 s	18 s	5	16	11 s	5	31	8.3	21.5	– memout –	
	(16,15,6)	3 s	40 s	6	32	34 s	6	63	11.2	26.2	– memout –	

\+ performance on BEB & CSMA/CD models
\+ vs. model-checking

\+ a bit faster than POR
\− does not work well for dining cryptographers

Conclusion

A new approach to SMC for MDPs
based on **on-the-fly confluence detection** 

– detect confluence on-the-fly on the concrete state space
– handle more kinds of nondeterminism than POR method

approach	nondeterminism	probabilities	memory	error bounds
POR	spurious interleavings	$P_{\max} = P_{\min}$	$s \ll n$	unchanged
⇨ confluence	confluent spurious	$P_{\max} = P_{\min}$	$s \ll n$	unchanged
learning	any	$P_{\max}$ only	$s \to n$	convergence

See also

www.modestchecker.net

& H, T.: On-the-fly Confluence Detection for Statistical Model Checking (NFM 2013)