

Markov Reward Model Checker

Joost-Pieter Katoen and Ivan S. Zapreev

MOVES, RWTH Aachen, Germany

FMT, Twente, The Netherlands

November 9, 2007

Outline

Introduction to MRMC

Model checking with MRMC

Getting MRMC models

MRMC versus PRISM, Ymer, VESTA, ETMCC

Implementation metrics

The third-party projects

The next release

Conclusions and future work

Introduction to MRMC

What is MRMC?

- ▶ A probabilistic model checker for DTMCs, CTMCs and MRMs
- ▶ A command-line tool implemented in C
- ▶ The tool is:
 - ▶ available for Windows, Mac OS X and Linux
 - ▶ distributed under the GPL license

What makes MRMC so special?

- ▶ It is small and fast, perfect as a backend
- ▶ It supports:
 - ▶ Bisimulation minimization [KKZJ07]
 - ▶ Precise on-the-fly steady-state detection [KZ05]
 - ▶ Improved model checking for steady-state properties

The hart of MRMC

Sparse matrices

$$\mathbf{P} = \begin{bmatrix} 0.50 & 0.50 & 0.00 \\ 0.25 & 0.00 & 0.75 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

nrows # rows

succ # successors

pred # predecessors

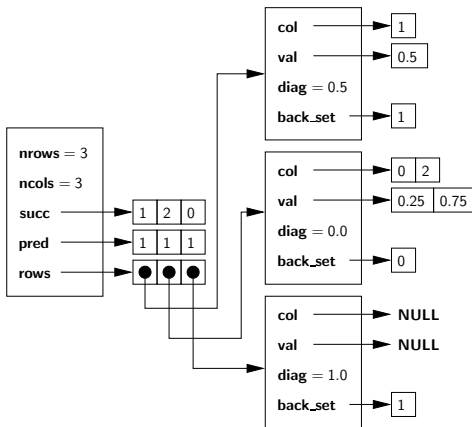
rows matrix rows

col non-zero col. indexes

val non-zero elem. values

diag a diagonal value

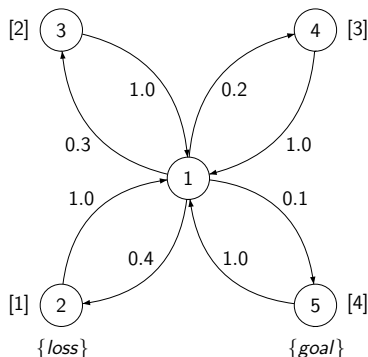
back_set predecessors



A simple DMRM model

- ▶ Consider a dice with only four wedges: 1, 2, 3 and 4
- ▶ The outcomes have probabilities 0.4, 0.3, 0.2 and 0.1
- ▶ the outcome 4 is the *goal* state
- ▶ the outcome 1 is the *loss* state
- ▶ $P_{>0.5}(\neg \text{loss} \cup_{[5,50]}^{[0,199]} \text{goal})$

game.tra	game.lab	game.rew
STATES 5	#DECLARATION	2 1
TRANSITIONS 8	loss goal	3 2
1 2 0.4	#END	4 3
1 3 0.3	2 loss	5 4
1 4 0.2	5 goal	
1 5 0.1		
2 1 1.0		
3 1 1.0		
4 1 1.0		
5 1 1.0		



Verifying $P_{>0.5} \left(\neg \text{loss} \ U_{[5,50]}^{[0,199]} \text{goal} \right)$ with MRMC

```
1  MRMC/bin> mrmc prctl game.tra game.lab game.rewi
2  ...
3  Logic                      = PRCTL
4  Loading the 'game.tra' file, please wait.
5  States=5, Transitions=8
6  Loading the 'game.lab' file, please wait.
7  Loading the 'game.rew' file, please wait.
8  The Occupied Space is 992 Bytes.
9  Type 'help' to get help.
10 >>P{>0.5}[ !loss U[0,199][5,50] goal]
11 $RESULT: ( 0.0647999, 0.0000000, 0.0959998, 0.1199998, 0.1199997 )
12 $STATE: {  }
13 The Total Elapsed Model-Checking Time is 45 milli sec(s).
14 >>
```

Getting MRMC models

PRISM [HKNP06]

- ▶ A high-level state-based description language based on the Reactive Modules formalism.
- ▶ The underlying models are DTMCs, CTMCs, MDPS and MRMs.
- ▶ The tool allows for exporting its models into the MRMC file formats.

PEPA Workbench [TG06]

- ▶ An algebraic process-oriented language for modeling concurrent systems.
- ▶ The underlying models are CTMCs.
- ▶ The tool allows for exporting PEPA models into the MRMC file formats.

MRMC versus PRISM, Ymer, VESTA, ETMCC

Supported platforms

Tool	Linux	Windows	Solaris	Mac OS X
PRISM	✓	✓	✓	✓
MRMC	✓	✓		✓
ETMCC	✓	✓	✓	
VESTA	✓	✓	?	
Ymer	✓			

MRMC versus PRISM, Ymer, VESTA, ETMCC

Models and logics

Tool	DTMC	CTMC	MDP	DMRM	CMRM	GSMP
MRMC	✓	✓		✓	✓	
PRISM	✓	✓	✓	✓	✓	
VESTA	✓	✓				
Ymer		✓				✓
ETMCC		✓				

Only MRMC supports all of PCTL, CSL, PRCTL and CSRL!

MRMC versus PRISM, Ymer, VESTA, ETMCC

The supported operators of PCTL

Tool	$\mathcal{L}_{\bowtie p}[\Phi]$	$\mathcal{P}_{\bowtie p}[\Phi \mathcal{U} \Psi]$	$\mathcal{P}_{\bowtie p}[\Phi \mathcal{U}^{\leq k} \Psi]$	$\mathcal{P}_{\bowtie p}[\Phi \mathcal{U}^{[k1, k2]} \Psi]$
MRMC	✓	✓	✓	✓
PRISM		✓	✓	
VESTA		✓	✓	

MRMC versus PRISM, Ymer, VESTA, ETMCC

The supported operators of CSL

Tool	$S_{\Delta\phi}[\phi]$	$P_{\Delta\phi}[\phi \ \mathcal{U} \ \psi]$	$P_{\Delta\phi}[\phi \ \mathcal{U}_{\leq t} \ \psi]$	$P_{\Delta\phi}[\phi \ \mathcal{U}_{\geq t} \ \psi]$	$P_{\Delta\phi}[\phi \ \mathcal{U}_{[t_1, t_2]} \ \psi]$	$P_{\Delta\phi}[\chi_{\leq t} \ \psi]$	$P_{\Delta\phi}[\chi_{[t_1, t_2]} \ \psi]$
MRMC	✓	✓	✓		✓	✓	✓
PRISM	✓	✓	✓	✓	✓		
Ymer			✓		✓	✓	✓
VESTA		✓	✓			✓	
ETMCC	✓	✓	✓				

Implementation metrics

MRMC metrics	Value	Tool
Lines of code	6738	<i>Understand C/C++</i>
Lines of comments	8287	
McCabes cyclomatic complexity	1399	<i>CCCC</i>
Development effort estimate	20.31 MM	<i>SLOCCount</i>

Test-suite metrics	Value	Tool
Test coverage	83.46%	<i>GCov</i>
Lines of code	1474	<i>SLOCCount</i>
Development effort estimate	3.61 MM	

The third-party projects

GreatSPN v2.0

- ▶ Department information, Università di Torino, Italy
- ▶ Modeling, validation, and performance evaluation of distributed systems
- ▶ **MRMC as a backend for CSL model checking** [CDDS06]

Heuristics-Guided Dependability Analysis

- ▶ The chair of Software Engineering, Universität des Konstanz, Germany
- ▶ Generating diagnostics information for stochastic models [AL06]
- ▶ **A prototype tool called DiPro is being linked to MRMC**

Reachability analysis in uniform CTMDPs

- ▶ Dependable Systems & Software group, Universität des Saarlandes, Germany
- ▶ New timed-reachability algorithms for uCTMDPs [BHH⁺06]
- ▶ **The tool chain: STATEMATE – extended MRMC**

The next release

What to expect from MRMC v1.3?

- ▶ Model checking via discrete-event simulation (PCTL, CSL)
- ▶ Model checking of uCTMDPs (reachability properties)
- ▶ Optimized performance and memory usage
- ▶ Improved command-prompt interface:
 - ▶ Access to the intermediate model-checking results
 - ▶ Context help and run-time settings
 - ▶ *Et cetera ...*
- ▶ Simplified internal interfaces for the external developers

Conclusions and future work

Conclusions

- ▶ **MRMC** is small and fast
- ▶ It is the only tool supporting:
 - ▶ PCTL, CSL, PRCTL and CSRL
 - ▶ Bisimulation minimization
 - ▶ Precise steady-state detection
- ▶ It is available for: Windows, Linux and Mac OS X
- ▶ There are several third-party projects that use **MRMC**

Future work

- ▶ State-space abstractions
- ▶ Counter examples
- ▶ MDPs, CTMCPs, etc.



Husain Aljazzar and Stefan Leue.

Extended directed search for probabilistic timed reachability.

In *Formal Modeling and Analysis of Timed Systems (FORMATS'06)*, pages 33–51, 2006.



Eckard Bode, Marc Herbstritt, Holger Hermanns, Sven Johr, Thomas Peikenkamp, Reza Pulungan, Ralf Wimmer, and Bernd Becker.

Compositional Performability Evaluation for STATEMATE.

In *QEST '06: Proceedings of the 3rd international conference on the Quantitative Evaluation of Systems*, pages 167–178, Washington, DC, USA, 2006. IEEE Computer Society.



D. Cerotti, D. D'Aprile, S. Donatelli, and J. Sproston.

Verifying stochastic well-formed nets with csl model-checking tools.

In *ACSD '06: Proceedings of the Sixth International Conference on Application of Concurrency to System Design*, pages 143–152, Washington, DC, USA, 2006. IEEE Computer Society.



A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker.

PRISM: A tool for automatic verification of probabilistic systems.

In H. Hermanns and J. Palsberg, editors, *Proc. 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06)*, volume 3920 of *LNCS*, pages 441–444. Springer, 2006.



Joost-Pieter Katoen, Tim Kemna, Ivan Zapreev, and David N. Jansen.

Bisimulation Minimisation Mostly Speeds Up Probabilistic Model Checking.

In Orna Grumberg and Michael Huth, editors, *Tools and Algorithms for the Construction and Analysis of Systems, TACAS'07*, volume 4424 of *LNCS*, pages 87–101. Springer, 2007.



J.-P. Katoen and I. S. Zapreev.

Safe On-The-Fly Steady-State Detection for Time-Bounded Reachability.

Technical Report TR-CTIT-05-52, CTIT, University of Twente, 2005.



Mirco Tribastone and Stephen Gilmore.

A New Generation PEPA Workbench.

In *Process Algebra and Stochastically Timed Activities (PASTA'06)*, pages 1820–1845, 2006.