

A Markov Reward Model Checker

Joost-Pieter Katoen^{‡,†}, Maneesh Khattri[†] and Ivan S. Zapreev[†]

[‡] RWTH Aachen, Germany

[†] University of Twente, The Netherlands

October 25, 2006

Verification of Markov Reward Models

MRLMs are underneath:

- Reward extensions of stochastic process algebras
- Stochastic reward nets
- Etc.

Rewards:

- **State** rewards
- **Impulse** rewards

[▶ Example](#)

Logics:

PRCTL - Probabilistic Reward Computation Tree Logic,
 $PCTL \subset PRCTL$.

CSRL - Continuous Stochastic Reward Logic,
 $CSL \subset CSRL$.

Allowed properties

PRCTL extends PCTL with:

- 1 The expected reward rate at a time instant
- 2 The long-run expected reward rate per time unit
- 3 The instantaneous reward at a time instant
- 4 The expected accumulated reward at a time instant

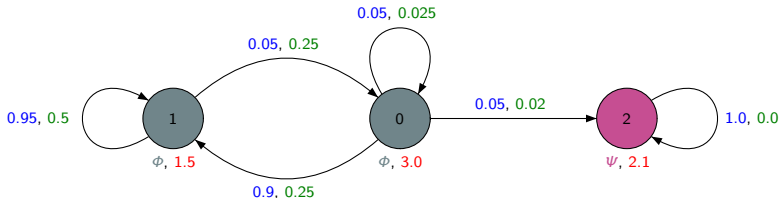
CSRL extends CSL with

- 1 **PRCTL**
- 2 The probability to reach one of the goal states (via indicated allowed states) within time t while having earned an accumulated reward that does not exceed r is larger than p

Examples

Example: PRCTL

$\mathcal{Y}_{[2,5]}^3 \phi$ - the expected accumulated cost in ϕ -states within 3 hops is between 2 and 5



Example: CSRL

$\mathcal{P}_{\geq 0.3}(\phi \mathcal{U}_{(23, \infty)}^{\leq 4} \psi)$ - a ψ -state is reached with probability at least 0.3 in at most 4 time units along an ϕ -path with total cost > 23

Implementation

Algorithms:

Sparse matrix - compressed row, compressed column

PCTL - (Hansson and Jonsson, 1994)

PRCTL - (Andova et al., 2003)

CSL - (Baier et al., 2003)

CSRL - Discretization (Tijms and Veldman, 2000),
Uniformization (Qureshi and Sanders, 1996)

Improvements:

- Search for bottom strongly connected components
- On-the-fly steady state detection
- Path graph representation

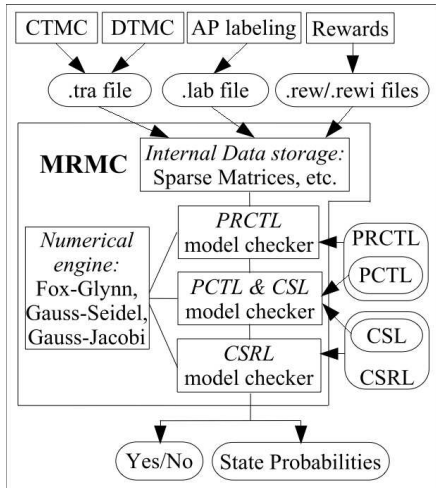
A command-line tool

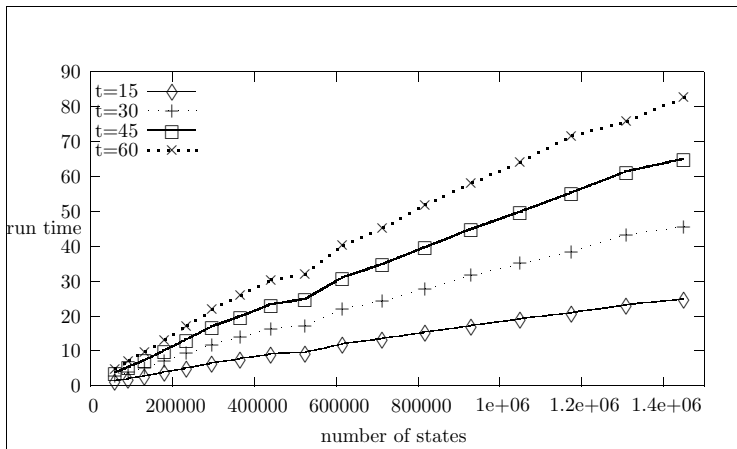
Input files:

- `.tra` - the probability/rate matrix
- `.lab` - the state-labeling
- `.rew` - the state rewards
- `.rewi` - the impulse rewards

Example

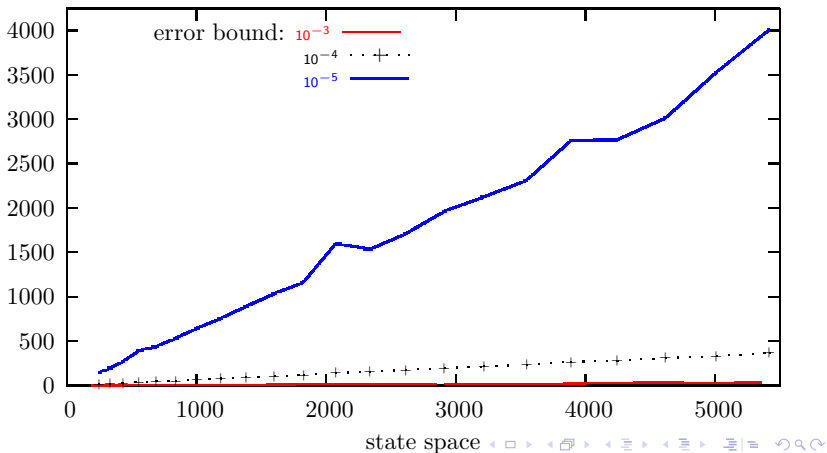
```
46:jewi410:/> ../../bin/mrmc csrl
csrl_unif_state_impulse_01.tra csrl_unif_state_impulse_01.lab
csrl_unif_state_impulse_01.rew csrl_unif_state_impulse_01.rewi
Running in the CSRL model checking mode.
States=5, Transitions=11
Space Occupied:: 1424 Bytes.
Type 'help' to get help.
>>P{>0.1}[ Sup U[0,50][0,3000] failed ]
State = 0, Probability = 5.087386e-03, Error Bound = 5.325909e-10
State = 1, Probability = 1.230380e-02, Error Bound = 4.466197e-10
State = 2, Probability = 1.000000e-00, Error Bound = 4.147793e-12
State = 3, Probability = 1.000000e-00, Error Bound = 4.147793e-12
State = 4, Probability = 1.000000e-00, Error Bound = 4.147793e-12
Result: ( 5.087386125019004e-03, 1.230379607517677e-02,
9.99999999999958522e-01, 9.99999999999958522e-01,
9.99999999999958522e-01 )
time to compute: 953 micro sec(s)
states = {3, 4, 5, }
>>quit
```



CSL logic: $\Phi \cup^{[0,t]} \psi$ 

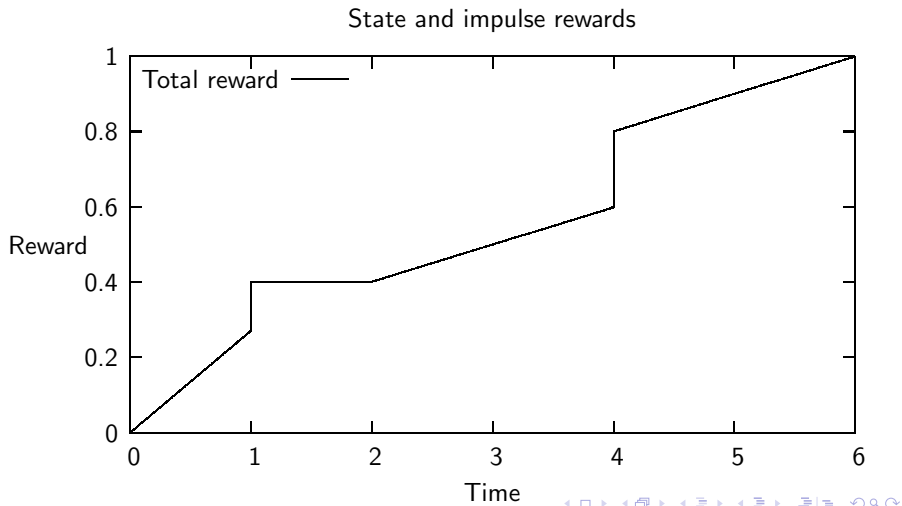
CSRL logic: $\Phi U_{[0,r]}^{[0,t]} \Psi$

computation time (in s)



- Andova, S., Hermanns, H., and Katoen, J.-P.: 2003,
Discrete-time rewards model-checked,
in *Formal Modeling and Analysis of Timed Systems (FORMATS)*, LNCS, Marseille, France
- Baier, C., Haverkort, B., Hermanns, H., and Katoen, J.-P.: 2003,
Model-checking algorithms for continuous-time Markov chains,
IEEE Trans. on Softw. Eng. **29(6)**, 524
- Hansson, N. and Jonsson, B.: 1994,
A logic for reasoning about time and probability,
Formal Aspects of Computing **6**, 512
- Qureshi, M. A. and Sanders, W. H.: 1996,
A new methodology for calculating distributions of reward accumulated during a finite interval.,
in *International Symposium on Fault-Tolerant Computing*, pp 116–125, Sendai, Japan
- Tijms, H. C. and Veldman, R.: 2000,
A fast algorithm for the transient reward distribution in continuous-time Markov chains,
in *Oper. Res. Lett.*, Vol. 26, pp 155–158

Verification of Markov Reward Models



A command-line tool

```
46:)ewi410:/> ../../../../bin/mrmc csrl
csrl_unif_state_impulse_01.tra csrl_unif_state_impulse_01.lab
csrl_unif_state_impulse_01.rew csrl_unif_state_impulse_01.rewi
Running in the CSRL model checking mode.
States=5, Transitions=11
Space Occupied:: 1424 Bytes.
Type 'help' to get help.
>>P{>0.1}[ Sup U[0,50][0,3000] failed ]
State = 0, Probabilty = 5.087386e-03, Error Bound = 5.325909e-10
State = 1, Probabilty = 1.230380e-02, Error Bound = 4.466197e-10
State = 2, Probabilty = 1.000000e-00, Error Bound = 4.147793e-12
State = 3, Probabilty = 1.000000e-00, Error Bound = 4.147793e-12
State = 4, Probabilty = 1.000000e-00, Error Bound = 4.147793e-12
Result: ( 5.087386125019004e-03, 1.230379607517677e-02,
9.999999999958522e-01, 9.999999999958522e-01,
9.999999999958522e-01 )
time to compute: 953 micro sec(s)
states = {3, 4, 5, }
>>quit
```