SMT-Based Bisimulation Minimisation of Markov Models

Dave Parker

University of Birmingham

Joint work with: Joost-Pieter Katoen and Christian Dehnert
(presented at ATVA 2013)

HIERATIC meeting, Jena, May 2013

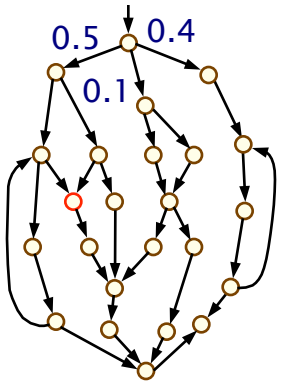
Overview

- Probabilistic model checking + PRISM
- Discrete-time Markov chains + PCTL
- Bisimulation minimisation
- SMT-based bisimulation minimisation

Probabilistic model checking

System

Probabilistic model
e.g. Markov chain



System require-ments

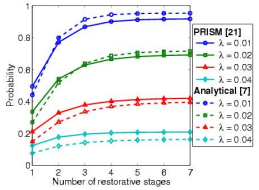
$P_{<0.1}$ [F fail]

Probabilistic temporal logic specification
e.g. PCTL, CSL, LTL

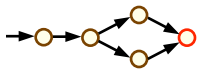
Probabilistic model checker
e.g. PRISM

Result

Quantitative results



Counter-example



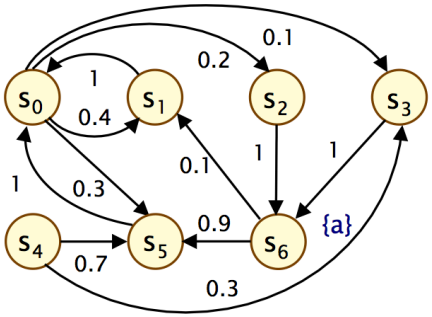
Probabilistic model checking with PRISM

System

PRISM model

```
module A
    a : [0..N] init N;
    ab : [0..N] init 0;
    [r1] a>0 → k1*a : (a' =a−1)&(ab' =ab+1);
    [r2] ab>0 → k2*ab : (a'=a+1)&(ab'=ab−1);
    [r3] a>0 → k3*a : (a'=a−1);
endmodule
```

Markov chain



Example PRISM model

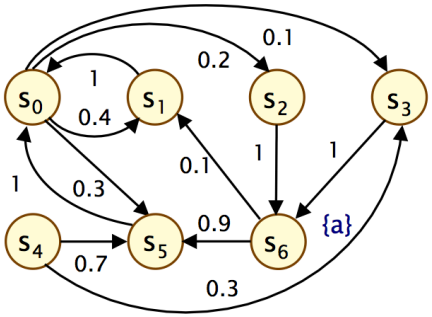
PRISM + bisimulation

System

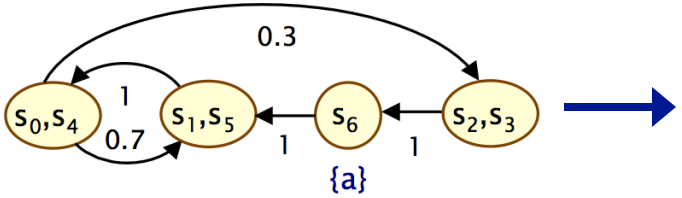
PRISM model

```
module A
   a : [0..N] init N;
   ab : [0..N] init 0;
   [r1] a>0 → k1*a : (a' =a−1)&(ab' =ab+1);
   [r2] ab>0 → k2*ab : (a'=a+1)&(ab'=ab−1);
   [r3] a>0 → k3*a : (a'=a−1);
endmodule
```

Markov chain



bisimulation minimisation



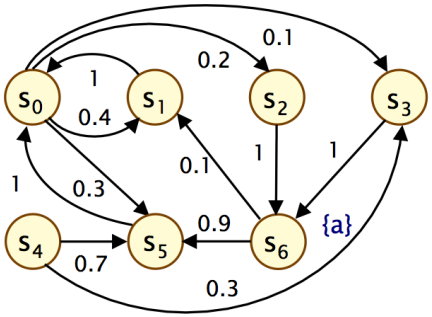
PRISM + bisimulation

System

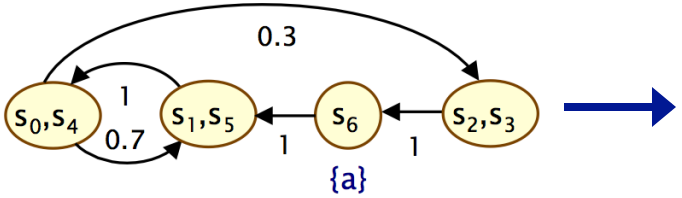
PRISM model

```
module A
    a : [0..N] init N;
    ab : [0..N] init 0;
    [r1] a>0 → k1*a : (a' =a−1)&(ab' =ab+1);
    [r2] ab>0 → k2*ab : (a'=a+1)&(ab'=ab−1);
    [r3] a>0 → k3*a : (a'=a−1);
endmodule
```

Markov chain

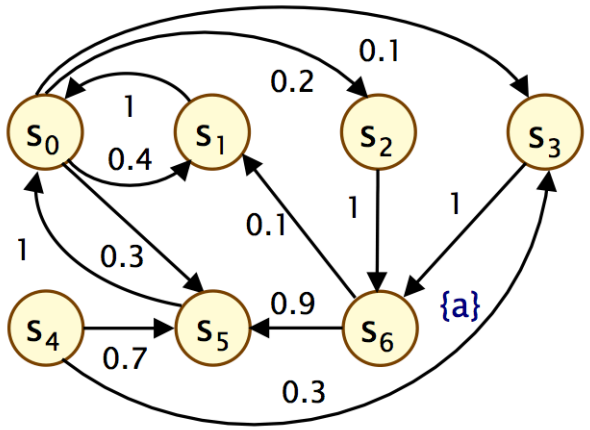


this talk



Discrete-time Markov chains (DTMCs)

- A DTMC D is a tuple (S, s_{init}, P, L) where:
  - S is a (finite) set of states ("state space")
  - $s_{init} \in S$ is the initial state
  - $P : S \times S \to [0,1]$ is the transition probability matrix where $\Sigma_{s' \in S} P(s,s') = 1$ for all $s \in S$
  - $L : S \to 2^{AP}$ is a function labelling states with atomic propositions

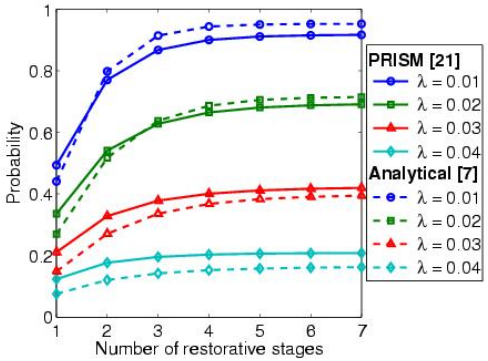


- Typical analysis: study probability of event occurrence
  - probability measure Pr_s over infinite paths (executions) in D
  - e.g. $Pr_s(F\ a) = Pr_s(\{ s_0 s_1 s_2 \ldots \in Path(s_{init}) \mid s_i \vDash a \text{ for some } i \})$

Probabilistic temporal logics

- PCTL = Probabilistic Computation Tree Logic [HJ94]
  - e.g. send → $P_{\geq 0.95}$ [$F^{\leq 10}$ deliver]
  - "if a message is sent, then the probability of it being delivered within 10 steps is at least 0.95"

- Quantitative (numerical) queries
  - $P_{=?}$ [$F^{\leq 10}$ deliver] – "what is the probability of
  - the message being delivered within 10 steps?"



- Extensions: Probabilistic LTL and PCTL*
  - e.g. $P_{\geq 1}$ [GF ready] – "with probability 1, the server always eventually returns to a ready-state"
  - e.g. $P_{<0.01}$ [FG error] – "with probability at most 0.01, an irrecoverable error occurs"
  - e.g. $P_{>0.5}$ [GF $crit_1$] ∧ $P_{>0.5}$ [GF $crit_2$]

Bisimulation on DTMCs

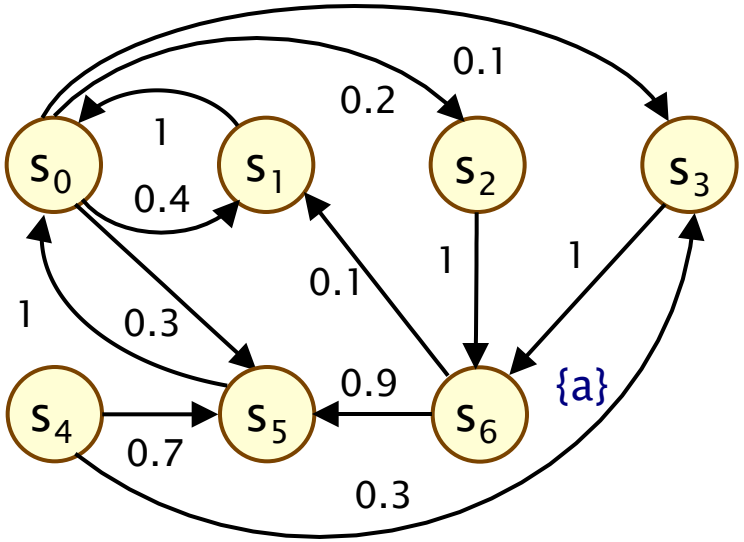
- Probabilistic bisimulation
  - preservation of stepwise behaviour (and labels)

- Equivalence relation R on S is a probabilistic bisimulation on D if and only if, for all s_1 R s_2:
  - $L(s_1) = L(s_2)$
  - $\mathbf{P}(s_1, T) = \mathbf{P}(s_2, T)$ for all $T \in S/R$ (i.e. for all equivalence classes of R)
  - where: $\mathbf{P}(s,T) = \Sigma_{s' \in T} \mathbf{P}(s,s')$ for $T \subseteq S$

- We write ~ for the coarsest possible bisimulation
  - we say states s_1 and s_2 are bisimilar if $s_1 \sim s_2$
  - and DTMCs D_1 and D_2 are bisimilar if their initial states are

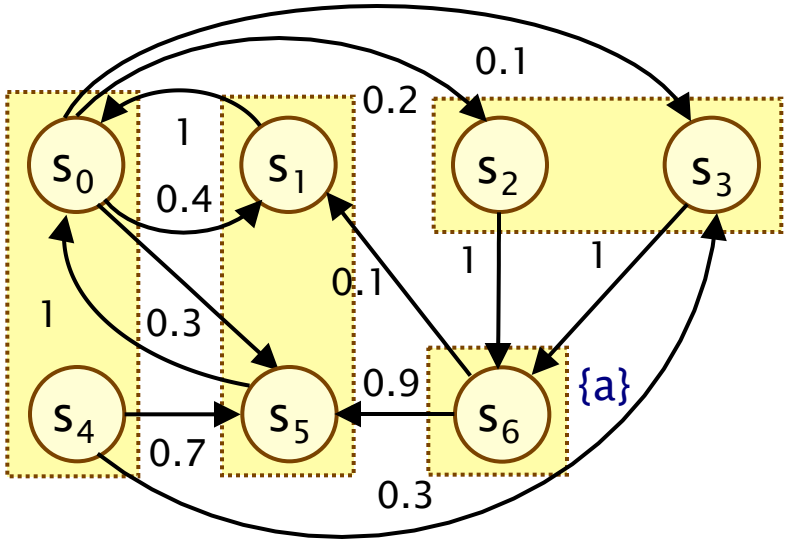
Example

- DTMC D with state space S



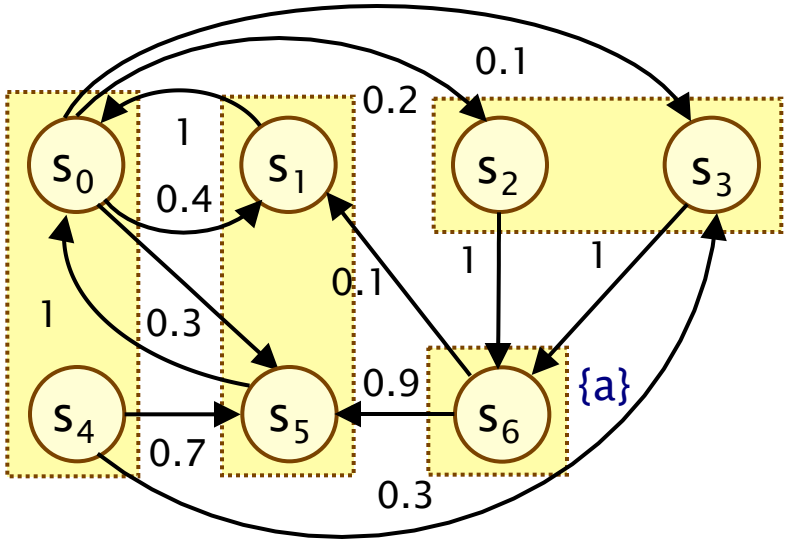
Example

- DTMC D with state space S
  - quotient state space $S/\sim = \{ \{s_1,s_5\}, \{s_6\}, \{s_2,s_3\}, \{s_0,s_4\} \}$

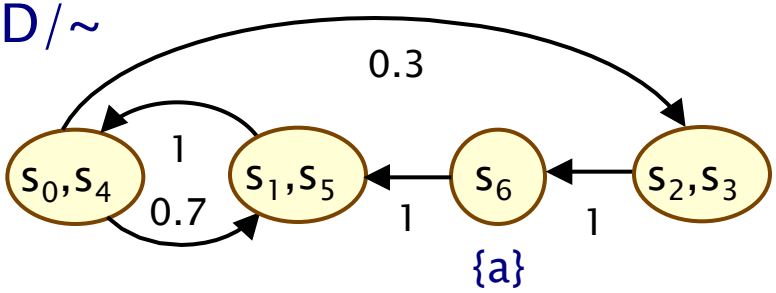


Example

- DTMC D with state space S
  - quotient state space $S/\sim = \{ \{s_1,s_5\}, \{s_6\}, \{s_2,s_3\}, \{s_0,s_4\} \}$



- Quotient DTMC D/~



Bisimulation and PCTL

- Probabilistic bisimulation preserves all PCTL formulae
- For all states s and s' :

$$s \sim s'$$

$$\Leftrightarrow$$

for all PCTL formulae Φ, $s \models \Phi$ if and only if $s' \models \Phi$

- Note also:
  - every pair of non-bisimilar states can be distinguished with some PCTL formula
  - ~ is the coarsest relation with this property
  - in fact, bisimulation also preserves all PCTL* formulae

Bisimulation minimisation

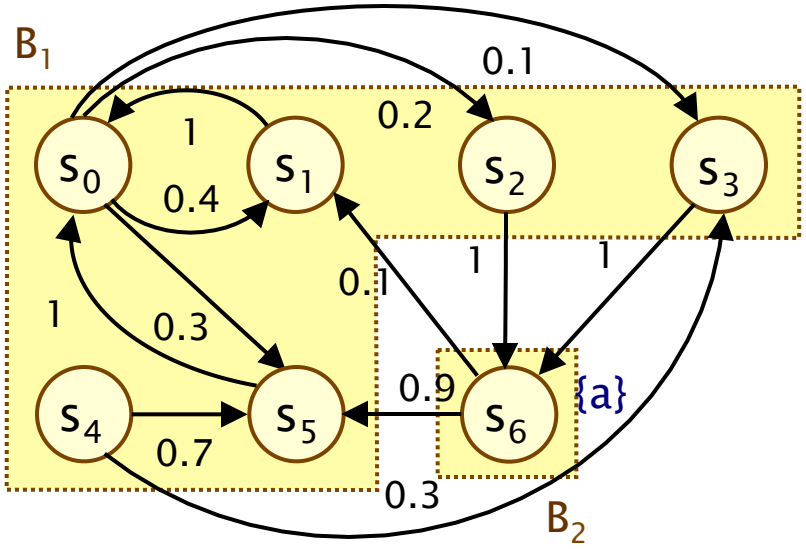
- Bisimulation minimisation
  - algorithm to compute the bisimulation relation
  - (and, in doing so, the quotient DTMC)
  - should yield more efficient PCTL model checking
  - (assuming quotient DTMC can be constructed efficiently)

- The basic algorithm
  - construct quotient DTMC based on partition refinement
  - repeated splitting of an initially coarse partition
  - final partition is coarsest bisimulation wrt. initial partition
  - complexity: $O(|\mathbf{P}| \cdot \log|S| + |AP| \cdot |S|)$ [DHS'03]
    - assuming suitable data structure used (splay trees)

Bisimulation minimisation

- 1. Start with initial partition
  - say $\Pi = \{ \{ s \in S \mid L(s)=lab \} \mid lab \in 2^{AP} \}$
- 2. Find a splitter $T \in \Pi$ for some block $B \in \Pi$
  - a splitter T is a block such that probability of going to T differs for some states in block B
  - i.e. $\exists s,s' \in B \,.\, \mathbf{P}(s,T) \neq \mathbf{P}(s',T)$
- 3. Split B into sub-blocks
  - such that $\mathbf{P}(s,T)$ is the same for all states in each sub-block
- 4. Repeat steps 2/3 until no more splitters exist
  - i.e. no change to partition Π

Example



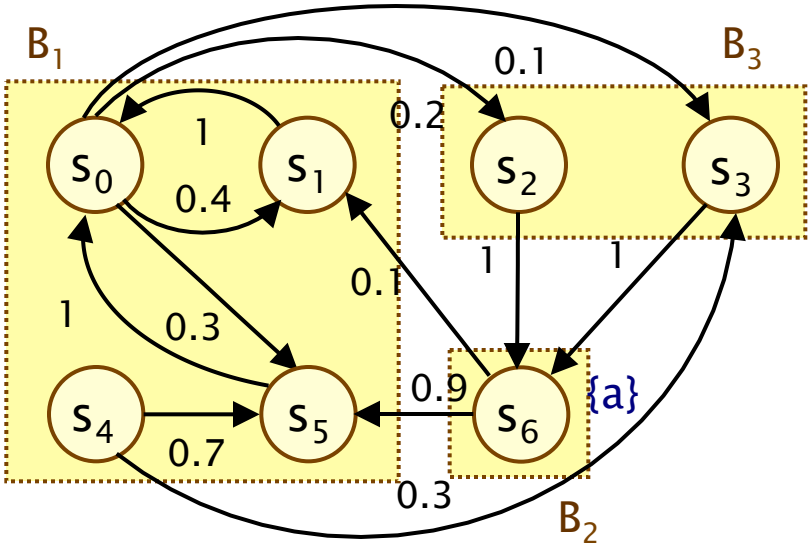
Minimisation:

Π_0: $B_1=\{s_0,s_1,s_2,s_3,s_4,s_5\}$, $B_2=\{s_6\}$

B_2 is a splitter for B_1

(since e.g. $R(s_1,B_2)=0\neq 1=R(s_2,B_2)$)

Example



Minimisation:

Π_0: $B_1=\{s_0,s_1,s_2,s_3,s_4,s_5\}$, $B_2=\{s_6\}$

B_2 is a splitter for B_1

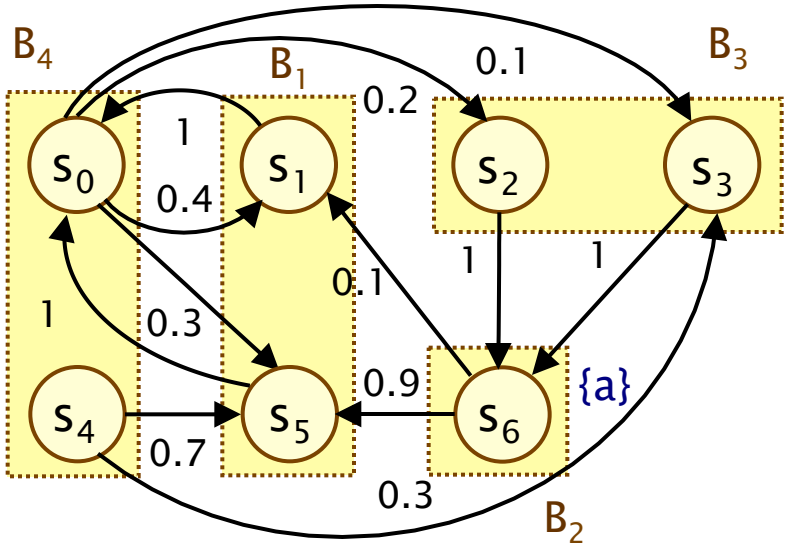
(since e.g. $R(s_1,B_2)=0\neq 1=R(s_2,B_2)$)

Π_1: $B_1=\{s_0,s_1,s_4,s_5\}$, $B_2=\{s_6\}$, $B_3=\{s_2,s_3\}$

B_3 is a splitter for B_1

(since e.g. $R(s_1,B_3)=0\neq 0.3=R(s_0,B_3)$)

Example



Minimisation:

Π_0: $B_1=\{s_0,s_1,s_2,s_3,s_4,s_5\}$, $B_2=\{s_6\}$

B_2 is a splitter for B_1

(since e.g. $R(s_1,B_2)=0\neq 1=R(s_2,B_2)$)

Π_1: $B_1=\{s_0,s_1,s_4,s_5\}$, $B_2=\{s_6\}$, $B_3=\{s_2,s_3\}$

B_3 is a splitter for B_1

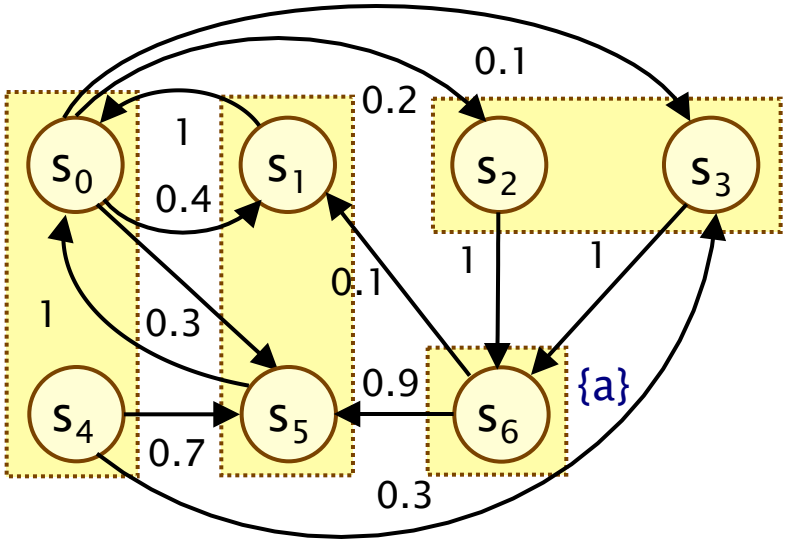
(since e.g. $R(s_1,B_3)=0\neq 0.3=R(s_0,B_3)$)

Π_2: $B_1=\{s_1,s_5\}$, $B_2=\{s_6\}$, $B_3=\{s_2,s_3\}$, $B_4=\{s_0,s_4\}$

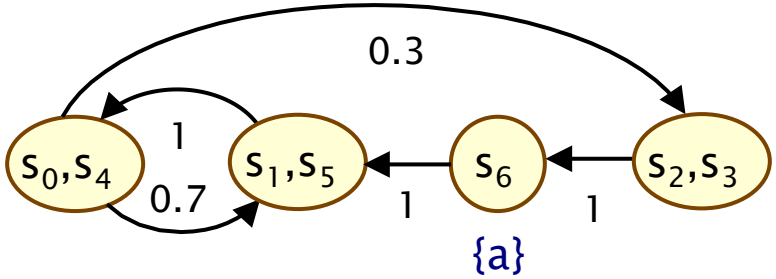
No more splitters…

$S/\sim = \{ \{s_1,s_5\}, \{s_6\}, \{s_2,s_3\}, \{s_0,s_4\} \}$

Example



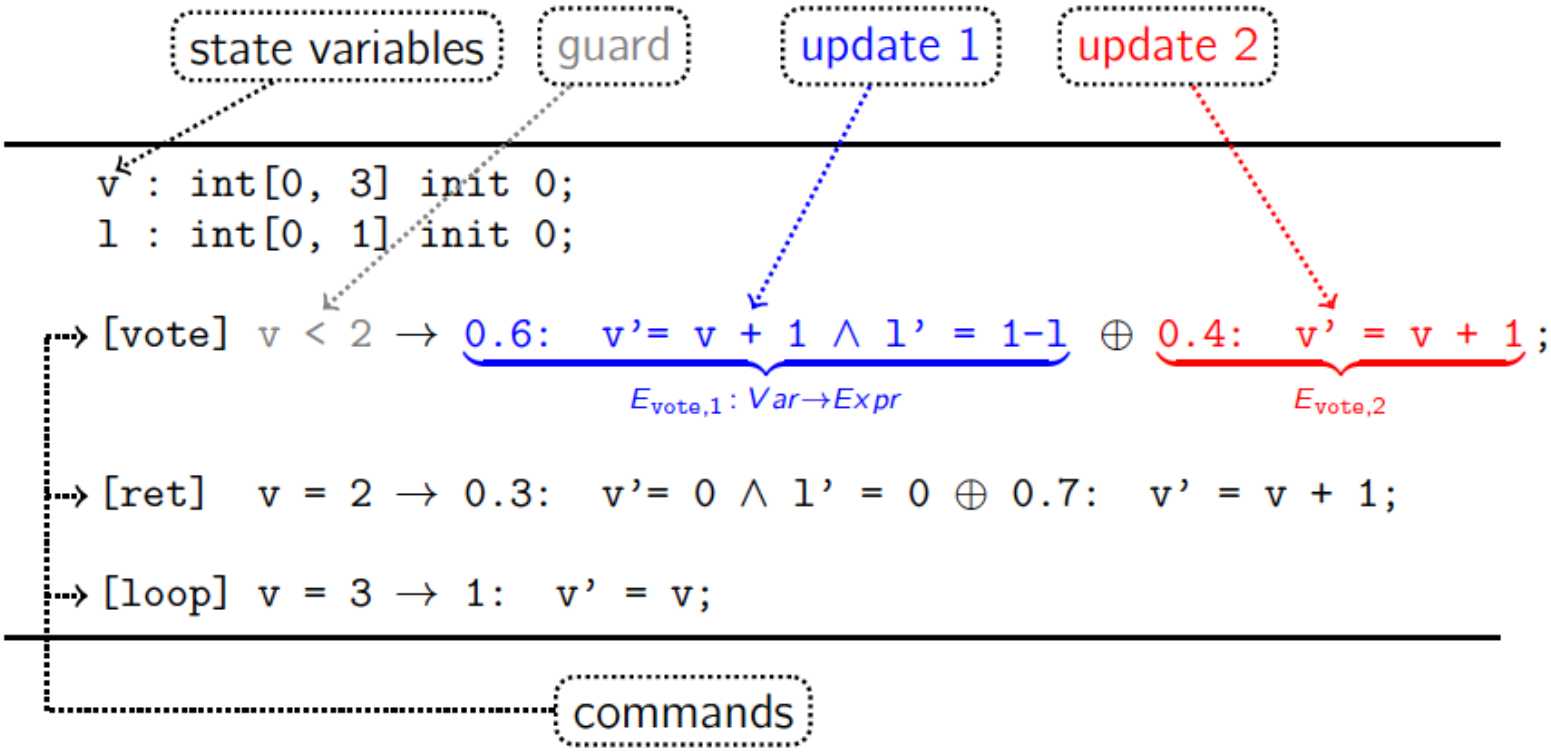
$S/\sim = \{ \{s_1,s_5\}, \{s_6\}, \{s_2,s_3\}, \{s_0,s_4\} \}$



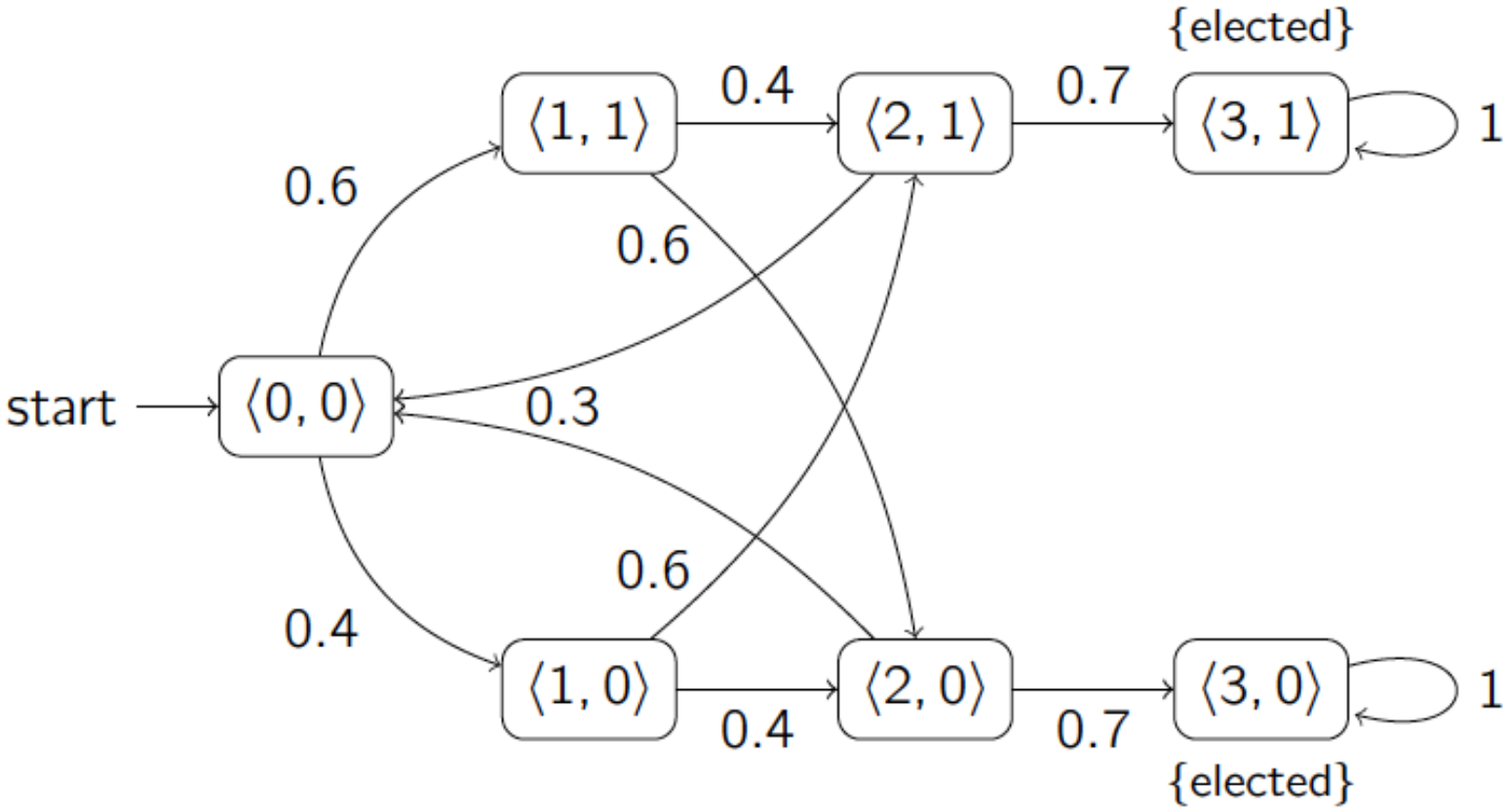
SAT and SMT

- SAT: Boolean satisfiability problem
  - e.g. do there exist values for Boolean variables x, y and z such that $(x \wedge y) \vee (\neg x \wedge z) \vee (y \wedge \neg z)$ is true?
  - classic NP-complete decision problem
  - many practical applications; many efficient SAT solvers exist
- SMT: Satisfiability modulo theories
  - SAT where (some) variables are predicates over some theory
  - (e.g. linear real/integer arithmetic, bit vectors, …)
  - e.g. do there exist values for integer variables a, b, c such that $(a+5b \leq 10) \vee (5a+c \geq 6) \wedge (b<c)$ is true?
  - again, many applications/solvers
- Variant: ALLSAT (for SAT or SMT)
  - extracts all satisfiable variable valuations

Example: PRISM model

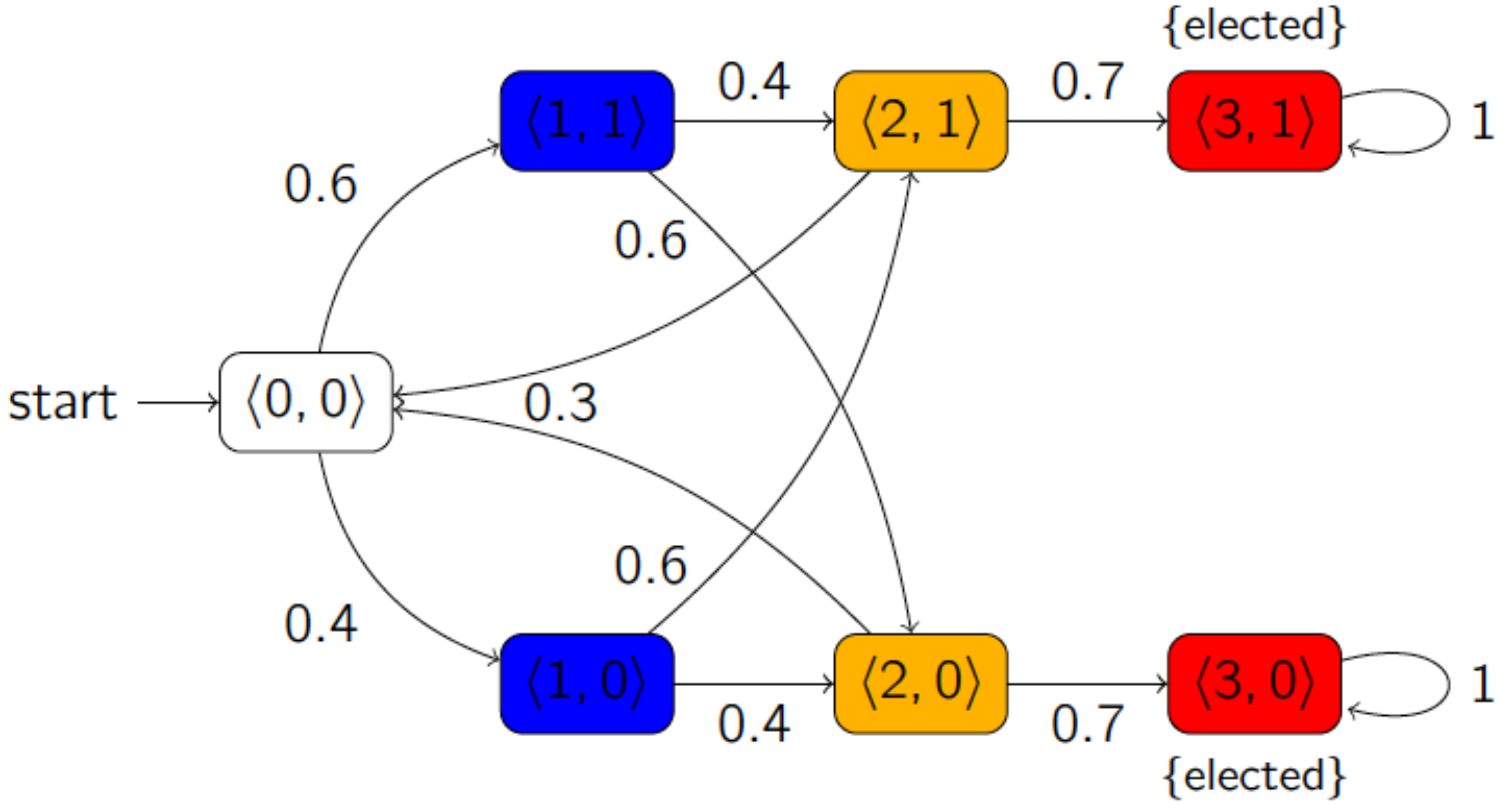


Example: DTMC



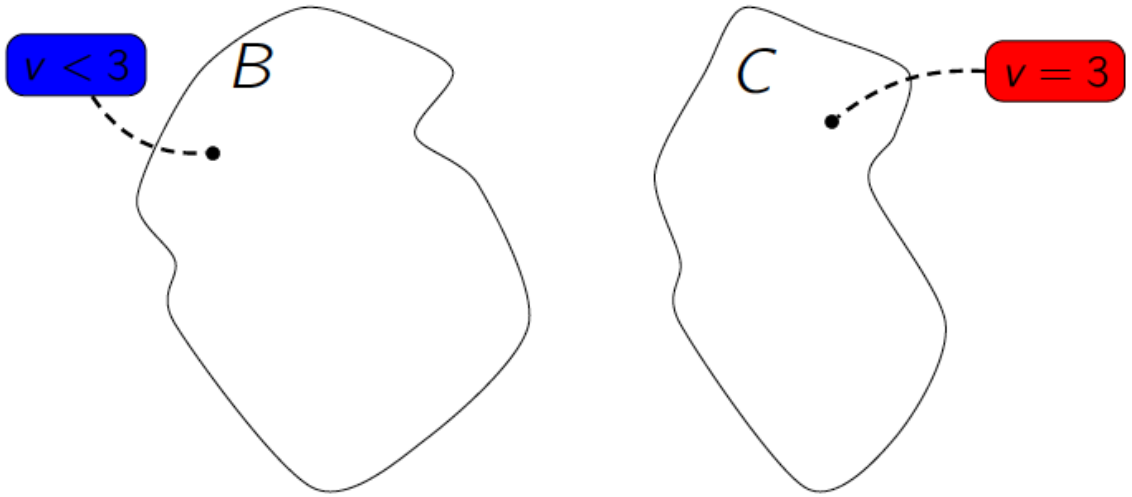
$s = \langle v, l \rangle$ $\{\text{elected}\} \mathrel{\widehat{=}} (v = 3)$

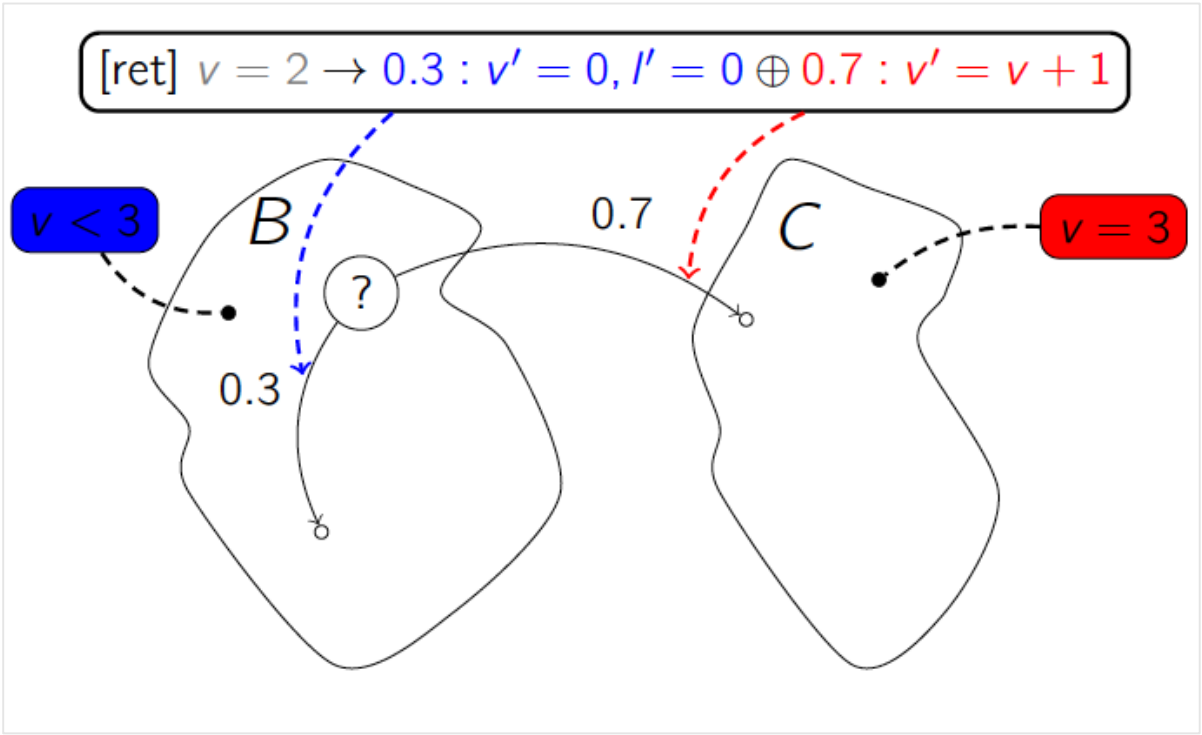
Example: bisimulation quotient



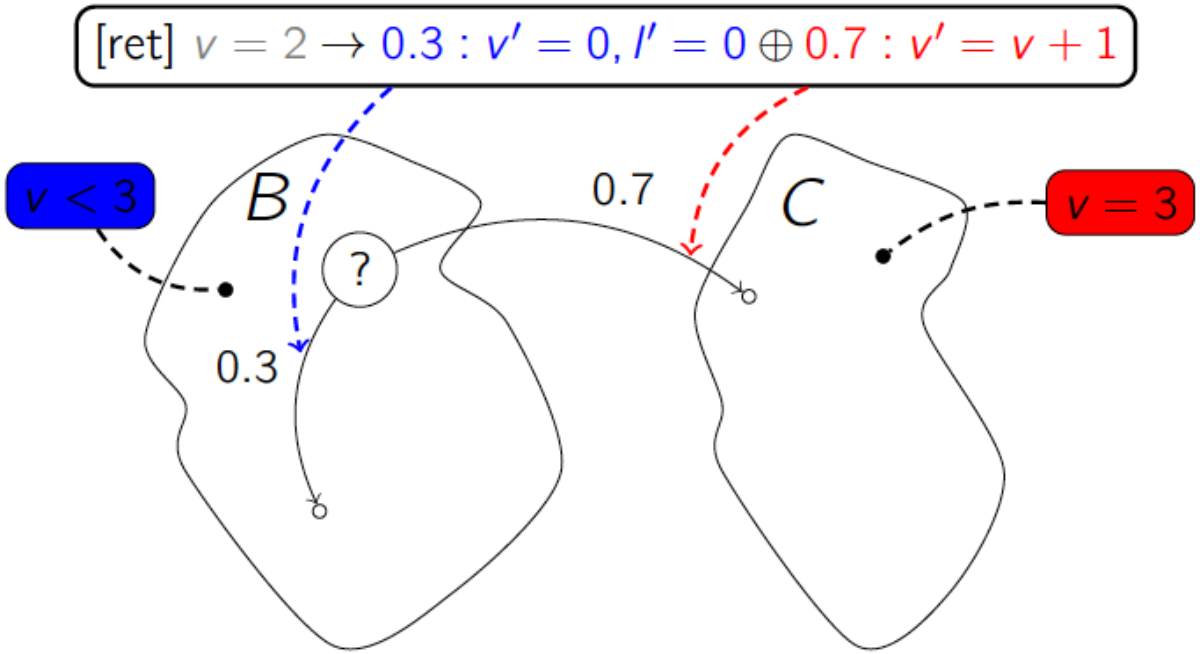
$B_{1,1} \mathrel{\widehat{=}} (v = 0)$ $B_{1,2} \mathrel{\widehat{=}} (v = 1)$ $B_2 \mathrel{\widehat{=}} (v = 2)$ $C \mathrel{\widehat{=}} (v = 3)$

Partition refinement via SMT



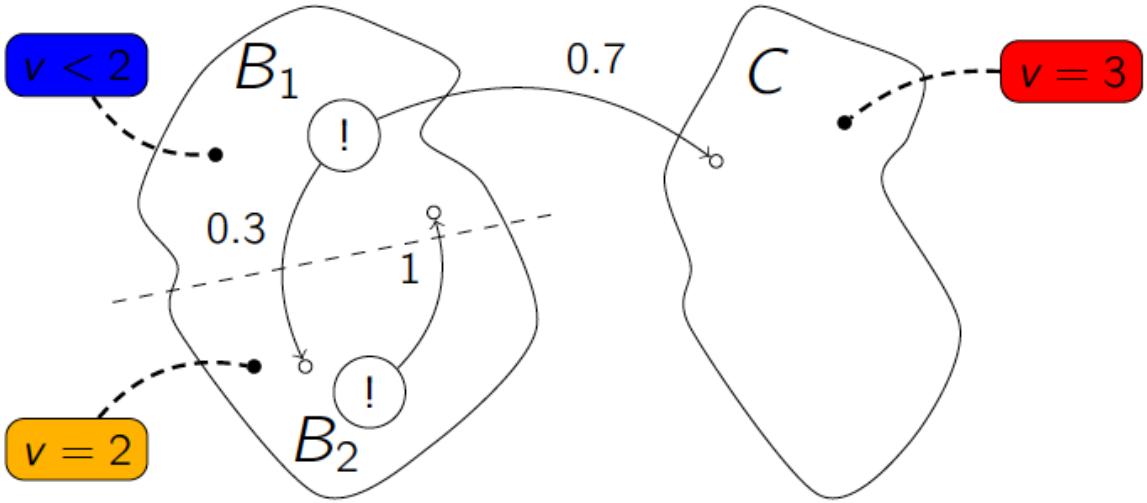

[ret] $v = 2 \rightarrow 0.3 : v' = 0, l' = 0 \oplus 0.7 : v' = v + 1$
$v < 3$
B
?
0.3
0.7
C
$v = 3$

Partition refinement via SMT



$$? \Leftrightarrow \underbrace{v < 3}_{B} \wedge \underbrace{v = 2}_{\text{guard}} \wedge \underbrace{0 < 3}_{wp(B, E_{\text{ret},1})} \wedge \underbrace{v + 1 = 3}_{wp(C, E_{\text{ret},2})} \text{ satisfiable}$$

Partition refinement via SMT



Implementation & results

- Prototype implementation using Z3 SMT solver
  - evaluated on 2 large DTMC benchmarks
- Example: Leader election protocol [Itai/Rodeh]

$$\text{speed-up} = \frac{\text{PRISM construct + verify}}{\text{SMT-based bisim. + verify}}$$

N	K		PRISM		quotient		speed-up
		states	const.	verif.	states	const.	
4	9	19817	6.00s	88.75s	10	11.40s	8.31
4	11	44107	41.74s	543.85s	10	26.57s	22.04
5	9	236745	414.91s	9456.78s	12	497.91s	19.82
5	11	644983	4083.82s	60784.22s	12	1945.99s	33.33
6	9	2659233	TO	TO	14	28548.85s	–

Conclusions

- SMT-based bisimulation minimisation
  - construction of quotient DTMC without building full model
  - good results on two benchmarks
- Also works (with some adaptation) for:
  - continuous-time Markov chains
  - Markov decision processes
  - infinite-state models
  - parameterised (symbolic) models
  - abstractions (not bisimulations)