Bisimulation minimisation mostly speeds up probabilistic model checking

Joost-Pieter Katoen [1,2], Tim Kemna [1], Ivan Zapreev [1,2] and David N. Jansen [1,2]

University of Twente [1]
RWTH-Aachen [2]

March 28, 2007

Probabilistic model checking

1. Enjoys a rapid increase of interest
2. Case studies:
   - Biological process modeling
   - Communication protocols
   - Randomised algorithms
   - Quantum computing
   - Planning and AI
   - Security
3. Formalisms that use probabilistic model checking:
   - Probabilistic extension of Promela (Baier et al., 2005a)
   - Stochastic process algebra PEPA (Hillston, 1996)
   - Stochastic Petri nets (D'Aprile et al., 2004)
   - Statemate (Bode et al., 2006)
4. Model checking tools:
   - LiQuor (Baier et al., 2005a)
   - PRISM (Kwiatkowska et al., 2004)
   - MRMC (Katoen et al., 2005)

Probabilistic model checking

1. Enjoys a rapid increase of interest
2. Case studies:
   - Biological process modeling
   - Communication protocols
   - Randomised algorithms
   - Quantum computing
   - Planning and AI
   - Security
3. Formalisms that use probabilistic model checking:
   - Probabilistic extension of Promela (Baier et al., 2005a)
   - Stochastic process algebra PEPA (Hillston, 1996)
   - Stochastic Petri nets (D'Aprile et al., 2004)
   - Statemate (Bode et al., 2006)
4. Model checking tools:
   - LiQuor (Baier et al., 2005a)
   - PRISM (Kwiatkowska et al., 2004)
   - MRMC (Katoen et al., 2005)

Probabilistic model checking

1. Enjoys a rapid increase of interest
2. Case studies:
   - Biological process modeling
   - Communication protocols
   - Randomised algorithms
   - Quantum computing
   - Planning and AI
   - Security
3. Formalisms that use probabilistic model checking:
   - Probabilistic extension of Promela (Baier et al., 2005a)
   - Stochastic process algebra PEPA (Hillston, 1996)
   - Stochastic Petri nets (D'Aprile et al., 2004)
   - Statemate (Bode et al., 2006)
4. Model checking tools:
   - LiQuor (Baier et al., 2005a)
   - PRISM (Kwiatkowska et al., 2004)
   - MRMC (Katoen et al., 2005)

Probabilistic model checking

1. Enjoys a rapid increase of interest
2. Case studies:
   - Biological process modeling
   - Communication protocols
   - Randomised algorithms
   - Quantum computing
   - Planning and AI
   - Security
3. Formalisms that use probabilistic model checking:
   - Probabilistic extension of Promela (Baier et al., 2005a)
   - Stochastic process algebra PEPA (Hillston, 1996)
   - Stochastic Petri nets (D'Aprile et al., 2004)
   - Statemate (Bode et al., 2006)
4. Model checking tools:
   - LiQuor (Baier et al., 2005a)
   - PRISM (Kwiatkowska et al., 2004)
   - MRMC (Katoen et al., 2005)

Motivation

Probabilistic model checking

1. State-space explosion
2. State-space reduction techniques:
    - Symmetry reduction (Kwiatkowska et al., 2006)
    - Binary decision diagrams (Kwiatkowska et al., 2004)
    - Abstraction refinement (D'Argenio et al., 2001)
    - Bisimulation equivalences (Baier et al., 2005b)

Bisimulation minimization

- Huge state-space reduction
- Is fully automated
- Drastic time penalty for LTL model checking (Fisler and Vardi, 1998; Fisler and Vardi, 1999; Fisler and Vardi, 2002)

Motivation

Probabilistic model checking

1. State-space explosion
2. State-space reduction techniques:
   - Symmetry reduction (Kwiatkowska et al., 2006)
   - Binary decision diagrams (Kwiatkowska et al., 2004)
   - Abstraction refinement (D'Argenio et al., 2001)
   - Bisimulation equivalences (Baier et al., 2005b)

Bisimulation minimization

- Huge state-space reduction
- Is fully automated
- Drastic time penalty for LTL model checking (Fisler and Vardi, 1998; Fisler and Vardi, 1999; Fisler and Vardi, 2002)

Motivation

Probabilistic model checking

1. State-space explosion
2. State-space reduction techniques:
   - Symmetry reduction (Kwiatkowska et al., 2006)
   - Binary decision diagrams (Kwiatkowska et al., 2004)
   - Abstraction refinement (D'Argenio et al., 2001)
   - Bisimulation equivalences (Baier et al., 2005b)

Bisimulation minimization

- Huge state-space reduction
- Is fully automated
- Drastic time penalty for LTL model checking (Fisler and Vardi, 1998; Fisler and Vardi, 1999; Fisler and Vardi, 2002)

Motivation

Probabilistic model checking

1. State-space explosion
2. State-space reduction techniques:
   - Symmetry reduction (Kwiatkowska et al., 2006)
   - Binary decision diagrams (Kwiatkowska et al., 2004)
   - Abstraction refinement (D'Argenio et al., 2001)
   - Bisimulation equivalences (Baier et al., 2005b)

Bisimulation minimization

- Huge state-space reduction
- Is fully automated
- Drastic time penalty for LTL model checking (Fisler and Vardi, 1998; Fisler and Vardi, 1999; Fisler and Vardi, 2002)

Motivation

Probabilistic model checking

1. State-space explosion
2. State-space reduction techniques:
    - Symmetry reduction (Kwiatkowska et al., 2006)
    - Binary decision diagrams (Kwiatkowska et al., 2004)
    - Abstraction refinement (D'Argenio et al., 2001)
    - Bisimulation equivalences (Baier et al., 2005b)

Bisimulation minimization

- Huge state-space reduction
- Is fully automated
- Drastic time penalty for LTL model checking (Fisler and Vardi, 1998; Fisler and Vardi, 1999; Fisler and Vardi, 2002)

Motivation

Probabilistic model checking

1. State-space explosion
2. State-space reduction techniques:
    - Symmetry reduction (Kwiatkowska et al., 2006)
    - Binary decision diagrams (Kwiatkowska et al., 2004)
    - Abstraction refinement (D'Argenio et al., 2001)
    - Bisimulation equivalences (Baier et al., 2005b)

Bisimulation minimization

- Huge state-space reduction
- Is fully automated
- Drastic time penalty for LTL model checking
  (Fisler and Vardi, 1998; Fisler and Vardi, 1999; Fisler and Vardi, 2002)

What is our contribution?

An empirical study

We did an empirical study on the effect of bisimulation minimization on probabilistic model checking.

Our main result

Bisimulation minimization often pays off.

What is our contribution?

An empirical study

We did an empirical study on the effect of bisimulation minimization on probabilistic model checking.

Our main result

Bisimulation minimization often pays off.

What is our contribution?

Consider

- Known theory
- Discrete and continuous time Markov Chains
- Reward extensions

An empirical study

- Use benchmark problems in the field (Kwiatkowska et al., 2007)
- Investigate 7 case studies
- Perform about 1870 experiments

Focus on

- The state-space reduction
- Time of lumping + verification
- Peak-memory consumption

What is our contribution?

Consider

- Known theory
- Discrete and continuous time Markov Chains
- Reward extensions

An empirical study

- Use benchmark problems in the field (Kwiatkowska et al., 2007)
- Investigate 7 case studies
- Perform about 1870 experiments

Focus on

- The state-space reduction
- Time of lumping + verification
- Peak-memory consumption

What is our contribution?

Consider

- Known theory
- Discrete and continuous time Markov Chains
- Reward extensions

An empirical study

- Use benchmark problems in the field (Kwiatkowska et al., 2007)
- Investigate 7 case studies
- Perform about 1870 experiments

Focus on

- The state-space reduction
- Time of lumping + verification
- Peak-memory consumption

What is our contribution?

Consider

- Known theory
- Discrete and continuous time Markov Chains
- Reward extensions

An empirical study

- Use benchmark problems in the field (Kwiatkowska et al., 2007)
- Investigate 7 case studies
- Perform about 1870 experiments

Focus on

- The state-space reduction
- Time of lumping + verification
- Peak-memory consumption

What is our contribution?

Consider

- Known theory
- Discrete and continuous time Markov Chains
- Reward extensions

An empirical study

- Use benchmark problems in the field (Kwiatkowska et al., 2007)
- Investigate 7 case studies
- Perform about 1870 experiments

Focus on

- The state-space reduction
- Time of lumping + verification
- Peak-memory consumption

What is our contribution?

Consider

- Known theory
- Discrete and continuous time Markov Chains
- Reward extensions

An empirical study

- Use benchmark problems in the field (Kwiatkowska et al., 2007)
- Investigate 7 case studies
- Perform about 1870 experiments

Focus on

- The state-space reduction
- Time of lumping + verification
- Peak-memory consumption

What is our contribution?

Consider

- Known theory
- Discrete and continuous time Markov Chains
- Reward extensions

An empirical study

- Use benchmark problems in the field (Kwiatkowska et al., 2007)
- Investigate 7 case studies
- Perform about 1870 experiments

Focus on

- The state-space reduction
- Time of lumping + verification
- Peak-memory consumption

What is our contribution?

Consider

- Known theory
- Discrete and continuous time Markov Chains
- Reward extensions

An empirical study

- Use benchmark problems in the field (Kwiatkowska et al., 2007)
- Investigate 7 case studies
- Perform about 1870 experiments

Focus on

- The state-space reduction
- Time of lumping + verification
- Peak-memory consumption

What is our contribution?

Consider

- Known theory
- Discrete and continuous time Markov Chains
- Reward extensions

An empirical study

- Use benchmark problems in the field (Kwiatkowska et al., 2007)
- Investigate 7 case studies
- Perform about 1870 experiments

Focus on

- The state-space reduction
- Time of lumping + verification
- Peak-memory consumption

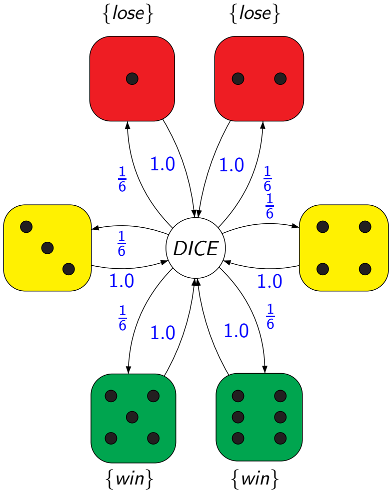
The considered models

Definition (Discrete time Markov chain)

A (labelled) DTMC is a tuple $(S, \mathcal{P}, AP, L)$:

- S - a finite set of *states*,
- AP - a finite set of *atomic propositions*,
- $L : S \rightarrow 2^{AP}$ - a *labelling* function,
- $\mathcal{P} : S \times S \rightarrow [0, 1]$ - a *probability matrix*,

$$\sum_{s' \in S} \mathcal{P}(s, s') = 1 \text{ for all } s \in S$$



Plus

- Continuous time Markov chains
- Reward extentions of both

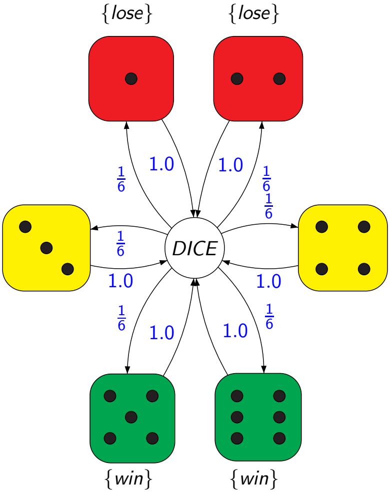
The considered models

Definition (Discrete time Markov chain)

A (labelled) DTMC is a tuple $(S, \mathcal{P}, AP, L)$:

- S - a finite set of *states*,
- AP - a finite set of *atomic propositions*,
- $L : S \rightarrow 2^{AP}$ - a *labelling* function,
- $\mathcal{P} : S \times S \rightarrow [0, 1]$ - a *probability matrix*,

$$\sum_{s' \in S} \mathcal{P}(s, s') = 1 \text{ for all } s \in S$$



Plus

- Continuous time Markov chains
- Reward extentions of both

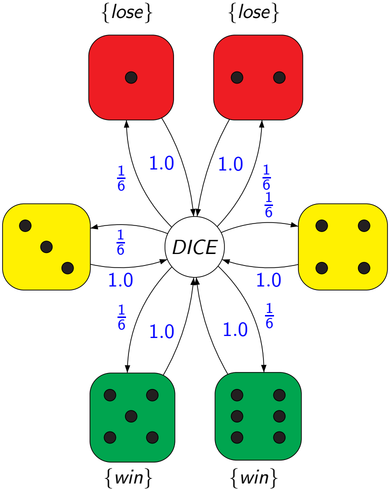
Probabilistic time-bounded reachability

Example

Determine states from which *win* states may be reached with a probability at least 0.9, within 10 time steps.

$$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 10} win)$$

Model	Example
DTMC	$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 10} win)$
CTMC	$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 3.5} win)$
Rewards	$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 15}_{\leq 13.7} win)$



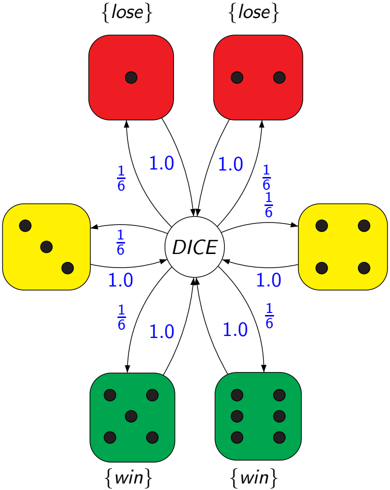
Probabilistic time-bounded reachability

Example

Determine states from which *win* states may be reached with a probability at least 0.9, within 10 time steps.

$$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 10} win)$$

Model	Example
DTMC	$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 10} win)$
CTMC	$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 3.5} win)$
Rewards	$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 15}_{\leq 13.7} win)$



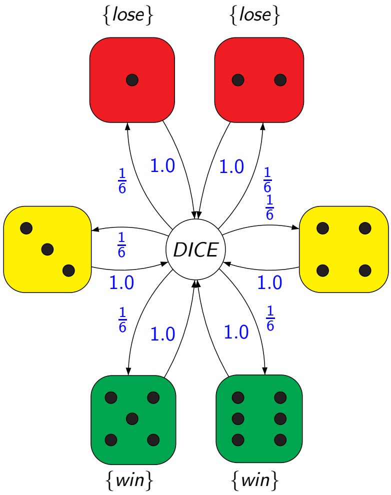
Probabilistic time-bounded reachability

Example

Determine states from which *win* states may be reached with a probability at least 0.9, within 10 time steps.

$$\mathcal{P}_{\geq 0.9}(\lozenge^{\leq 10} win)$$

Model	Example
DTMC	$\mathcal{P}_{\geq 0.9}(\lozenge^{\leq 10} win)$
CTMC	$\mathcal{P}_{\geq 0.9}(\lozenge^{\leq 3.5} win)$
Rewards	$\mathcal{P}_{\geq 0.9}(\lozenge^{\leq 15}_{\leq 13.7} win)$



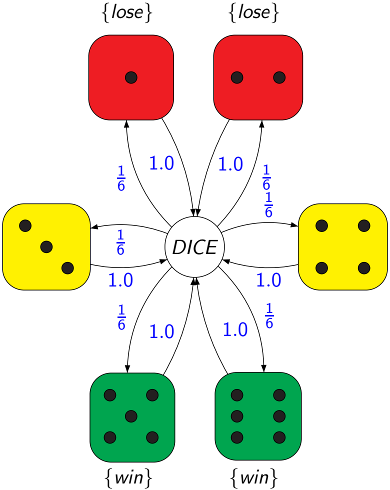
Probabilistic time-bounded reachability

Example

Determine states from which *win* states may be reached with a probability at least 0.9, within 10 time steps.

$$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 10} win)$$

Model	Example
DTMC	$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 10} win)$
CTMC	$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 3.5} win)$
Rewards	$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 15}_{\leq 13.7} win)$



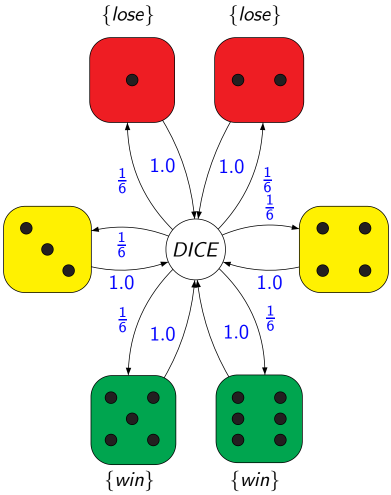
Probabilistic time-bounded reachability

Example

Determine states from which *win* states may be reached with a probability at least 0.9, within 10 time steps.

$$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 10} win)$$

Model	Example
DTMC	$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 10} win)$
CTMC	$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 3.5} win)$
Rewards	$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 15}_{\leq 13.7} win)$



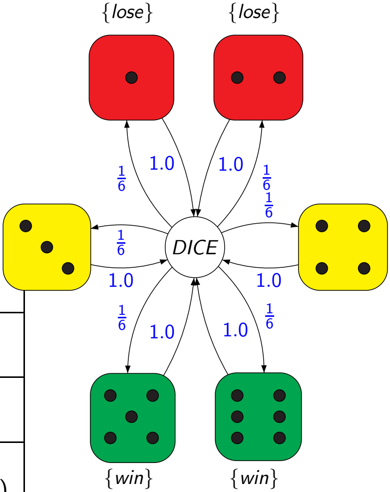
Probabilistic time-bounded reachability

Example

Determine states from which *win* states may be reached with a probability at least 0.9, within 10 time steps.

$$\mathcal{P}_{\geq 0.9}(\Diamond^{\leq 10} win)$$

Model	**Logic**
DTMC	PCTL (Hansson and Jonsson, 1994)
CTMC	CSL (Baier et al., 2003)
Rewards	PRCTL/CSRL (Andova et al., 2003; Baier et al., 2000)

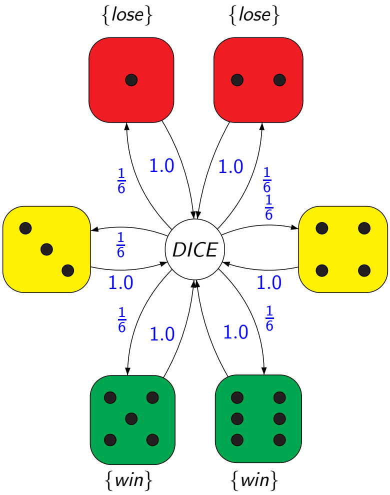


Bisimulation minimization

Definition (Strong bisimulation (Buchholz, 1994; Hillston, 1996))

- Let $D = (S, \mathcal{P}, AP, L)$ be a DTMC.
- Δ an equivalence relation on S.
- S/Δ is the *quotient* of S under Δ.
- Δ is a *strong bisimulation*, if $s_1 \Delta s_2 \Rightarrow$

$$L(s_1) = L(s_2)$$
$$\forall B \in S/\Delta \ : \ \mathcal{P}(s_1, B) = \mathcal{P}(s_2, B)$$

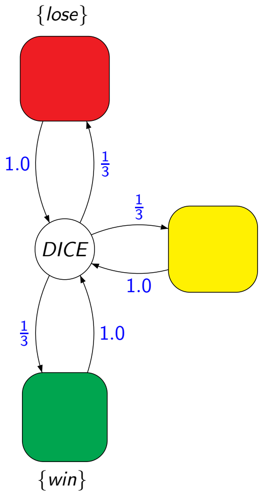


Bisimulation minimization

Definition (Strong bisimulation (Buchholz, 1994; Hillston, 1996))

- Let $D = (S, \mathcal{P}, AP, L)$ be a DTMC.
- Δ an equivalence relation on S.
- S/Δ is the *quotient* of S under Δ.
- Δ is a *strong bisimulation*, if $s_1 \Delta s_2 \Rightarrow$

$$L(s_1) = L(s_2)$$
$$\forall B \in S/\Delta \; : \; \mathcal{P}(s_1, B) = \mathcal{P}(s_2, B)$$



Preservation results

Theorem (1, (Aziz et al., 1995))

Let D be a DTMC, Δ a bisimulation and $s \in S$. Then $\forall \Phi \in PCTL^*$

$$s \models_D \Phi \iff [s]_\Delta \models_{D/\Delta} \Phi$$

Note

- Probabilistic bisimulation is the coarsest relation for Theor. 1.
- Since $s \sim [s]_\Delta$, verify properties on a bisimulation quotient.

Preservation results

Theorem (1, (Aziz et al., 1995))

Let D be a DTMC, Δ a bisimulation and $s \in S$. Then $\forall \Phi \in PCTL^*$

$$s \models_D \Phi \iff [s]_\Delta \models_{D/\Delta} \Phi$$

Note

- Probabilistic bisimulation is the coarsest relation for Theor. 1.
- Since $s \sim [s]_\Delta$, verify properties on a bisimulation quotient.

Preservation results

Theorem (1, (Aziz et al., 1995))

Let D be a DTMC, Δ a bisimulation and $s \in S$. Then $\forall \Phi \in PCTL^*$

$$s \models_D \Phi \iff [s]_\Delta \models_{D/\Delta} \Phi$$

Note

- Probabilistic bisimulation is the coarsest relation for Theor. 1.
- Since $s \sim [s]_\Delta$, verify properties on a bisimulation quotient.

Preservation results

Theorem (1, (Aziz et al., 1995))

Let D be a DTMC, Δ a bisimulation and $s \in S$. Then $\forall \Phi \in PCTL^*$

$$s \models_D \Phi \iff [s]_\Delta \models_{D/\Delta} \Phi$$

Note

- Probabilistic bisimulation is the coarsest relation for Theor. 1.
- Since $s \sim [s]_\Delta$, verify properties on a bisimulation quotient.

Measure-driven bisimulation

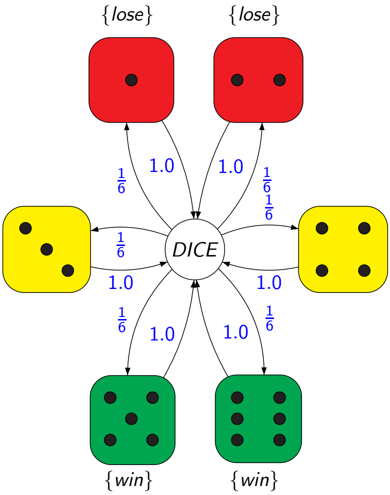
Definition (F-bisimulation (Baier et al., 2000))

- Let $D = (S, \mathcal{P}, AP, L)$ be a DTMC.
- F is a subset of PCTL formulas.
- Δ an equivalence relation on S.
- S/Δ is the *quotient* of S under Δ.
- Δ is an *F-bisimulation* on S, if $s_1 \Delta s_2$:

$$\forall \Phi \in F \ : \ s_1 \models \Phi \iff s_2 \models \Phi$$
$$\forall B \in S/\Delta \ : \ \mathcal{P}(s_1, B) = \mathcal{P}(s_2, B)$$

Example (F-bisimulation)

Let us take $F = \{win\}$.



Measure-driven bisimulation

Definition (F-bisimulation (Baier et al., 2000))

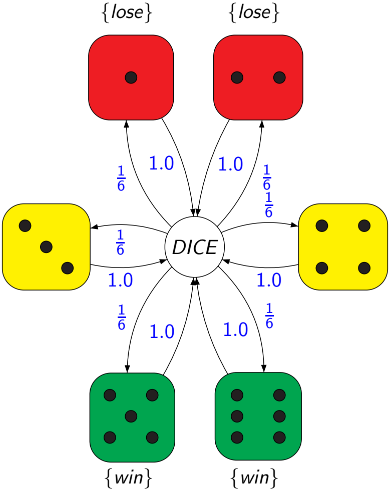
- Let $D = (S, \mathcal{P}, AP, L)$ be a DTMC.
- F is a subset of PCTL formulas.
- Δ an equivalence relation on S.
- S/Δ is the *quotient* of S under Δ.
- Δ is an *F-bisimulation* on S, if $s_1 \Delta s_2$:

$$\forall \Phi \in F \ : \ s_1 \models \Phi \iff s_2 \models \Phi$$

$$\forall B \in S/\Delta \ : \ \mathcal{P}(s_1, B) = \mathcal{P}(s_2, B)$$

Example (F-bisimulation)

Let us take $F = \{win\}$.



Measure-driven bisimulation

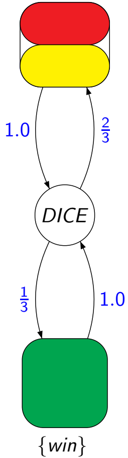
Definition (F-bisimulation (Baier et al., 2000))

- Let $D = (S, \mathcal{P}, AP, L)$ be a DTMC.
- F is a subset of PCTL formulas.
- Δ an equivalence relation on S.
- S/Δ is the *quotient* of S under Δ.
- Δ is an *F-bisimulation* on S, if $s_1 \Delta s_2$:

$$\forall \Phi \in F \ : \ s_1 \models \Phi \iff s_2 \models \Phi$$
$$\forall B \in S/\Delta \ : \ \mathcal{P}(s_1, B) = \mathcal{P}(s_2, B)$$

Example (F-bisimulation)

Let us take $F = \{win\}$.



Preservation results

Theorem ((Baier et al., 2003))

Let D be a DTMC, Δ an F-bisimulation and $s \in S$. Then $\forall \Phi \in PCTL_F$

$$s \models_D \Phi \iff [s]_\Delta \models_{D/\Delta} \Phi$$

Strong bisimulation vs. F-bisimulation

- Strong bisimilarity is F-bisimilarity for $F = AP$
- F-bisimulation is coarser than strong bisimulation
- Verify properties on F-bisimulation quotient

Preservation results

Theorem ((Baier et al., 2003))

Let D be a DTMC, Δ an F-bisimulation and $s \in S$. Then $\forall \Phi \in PCTL_F$

$$s \models_D \Phi \iff [s]_\Delta \models_{D/\Delta} \Phi$$

Strong bisimulation vs. F-bisimulation

- Strong bisimilarity is F-bisimilarity for $F = AP$
- F-bisimulation is coarser than strong bisimulation
- Verify properties on F-bisimulation quotient

Obtaining bisimulation quotient

Strong bisimulation (Derisavi et al., 2003)

- Partition refinement algorithm
- The worst-time complexity is $O(|P|log|S|)$

F-bisimulation

- A slight modification of the partition refinement algorithm.

Obtaining bisimulation quotient

Strong bisimulation (Derisavi et al., 2003)

- Partition refinement algorithm
- The worst-time complexity is $O(|P| log |S|)$

F-bisimulation

- A slight modification of the partition refinement algorithm.

Initial partitioning for $\mathcal{P}_{\trianglelefteq p}(\Phi \, \mathrm{U} \, \Psi)$ and $\mathcal{P}_{\trianglelefteq p}(\Phi \, \mathrm{U}^{[0,t]} \, \Psi)$

Note

- Strong bisimulation: Atomic propositions
- F – bisimulation: Formulas Φ, Ψ

$\mathcal{P}_{\trianglelefteq p}(\Phi \, \mathrm{U} \, \Psi)$

- Define $U_0 = Sat(\mathcal{P}_{\leq 0}(\Phi \, \mathrm{U} \, \Psi))$.
- Define $U_1 = Sat(\mathcal{P}_{\geq 1}(\Phi \, \mathrm{U} \, \Psi))$.
- Choose $F = \{U_0, U_1, S \setminus (U_0 \cup U_1)\}$.
- Apply F-bisimulation.

S_1 vs. U_1

A finer initial partitioning

$\mathcal{P}_{\trianglelefteq p}(\Phi \, \mathrm{U}^{[0,t]} \, \Psi)$

- Define $U_0 = Sat(\mathcal{P}_{\leq 0}(\Phi \, \mathrm{U} \, \Psi))$.
- Define $S_1 = Sat(\Psi)$.
- Choose $F = \{U_0, S_1, S \setminus (U_0 \cup S_1)\}$.
- Apply F-bisimulation.

Initial partitioning for $\mathcal{P}_{\trianglelefteq p}(\Phi \, \mathrm{U} \, \Psi)$ and $\mathcal{P}_{\trianglelefteq p}(\Phi \, \mathrm{U}^{[0,t]} \, \Psi)$

Note

- Strong bisimulation: Atomic propositions
- F – bisimulation: Formulas Φ, Ψ

$\mathcal{P}_{\trianglelefteq p}(\Phi \, \mathrm{U} \, \Psi)$

- Define $U_0 = Sat(\mathcal{P}_{\leq 0}(\Phi \, \mathrm{U} \, \Psi))$.
- Define $U_1 = Sat(\mathcal{P}_{\geq 1}(\Phi \, \mathrm{U} \, \Psi))$.
- Choose $F = \{U_0, U_1, S \setminus (U_0 \cup U_1)\}$.
- Apply F-bisimulation.

S_1 vs. U_1

A finer initial partitioning

$\mathcal{P}_{\trianglelefteq p}(\Phi \, \mathrm{U}^{[0,t]} \, \Psi)$

- Define $U_0 = Sat(\mathcal{P}_{\leq 0}(\Phi \, \mathrm{U} \, \Psi))$.
- Define $S_1 = Sat(\Psi)$.
- Choose $F = \{U_0, S_1, S \setminus (U_0 \cup S_1)\}$.
- Apply F-bisimulation.

Initial partitioning for $\mathcal{P}_{\trianglelefteq p}(\Phi \, \mathrm{U} \, \Psi)$ and $\mathcal{P}_{\trianglelefteq p}(\Phi \, \mathrm{U}^{[0,t]} \, \Psi)$

Note

- Strong bisimulation: Atomic propositions
- F – bisimulation: Formulas Φ, Ψ

$\mathcal{P}_{\trianglelefteq p}(\Phi \, \mathrm{U} \, \Psi)$

- Define $U_0 = Sat\,(\mathcal{P}_{\leq 0}(\Phi \, \mathrm{U} \, \Psi))$.
- Define $U_1 = Sat\,(\mathcal{P}_{\geq 1}(\Phi \, \mathrm{U} \, \Psi))$.
- Choose $F = \{U_0,\ U_1,\ S \setminus (U_0 \cup U_1)\}$.
- Apply F-bisimulation.

S_1 vs. U_1

A finer initial partitioning

$\mathcal{P}_{\trianglelefteq p}(\Phi \, \mathrm{U}^{[0,t]} \, \Psi)$

- Define $U_0 = Sat\,(\mathcal{P}_{\leq 0}(\Phi \, \mathrm{U} \, \Psi))$.
- Define $S_1 = Sat\,(\Psi)$.
- Choose $F = \{U_0,\ S_1,\ S \setminus (U_0 \cup S_1)\}$.
- Apply F-bisimulation.

Initial partitioning for $\mathcal{P}_{\trianglelefteq p}(\Phi \mathrm{U} \Psi)$ and $\mathcal{P}_{\trianglelefteq p}(\Phi \mathrm{U}^{[0,t]} \Psi)$

Note

- Strong bisimulation: Atomic propositions
- F – bisimulation: Formulas Φ, Ψ

S_1 vs. U_1

A finer initial partitioning

$\mathcal{P}_{\trianglelefteq p}(\Phi \mathrm{U} \Psi)$

- Define $U_0 = \mathit{Sat}(\mathcal{P}_{\leq 0}(\Phi \mathrm{U} \Psi))$.
- Define $U_1 = \mathit{Sat}(\mathcal{P}_{\geq 1}(\Phi \mathrm{U} \Psi))$.
- Choose $F = \{U_0, U_1, S \setminus (U_0 \cup U_1)\}$.
- Apply F-bisimulation.

$\mathcal{P}_{\trianglelefteq p}(\Phi \mathrm{U}^{[0,t]} \Psi)$

- Define $U_0 = \mathit{Sat}(\mathcal{P}_{\leq 0}(\Phi \mathrm{U} \Psi))$.
- Define $S_1 = \mathit{Sat}(\Psi)$.
- Choose $F = \{U_0, S_1, S \setminus (U_0 \cup S_1)\}$.
- Apply F-bisimulation.

Initial partitioning for $\mathcal{P}_{\trianglelefteq p}(\Phi \ \mathrm{U} \ \Psi)$ and $\mathcal{P}_{\trianglelefteq p}(\Phi \ \mathrm{U}^{[0,t]} \ \Psi)$

Note

- Strong bisimulation: Atomic propositions
- F – bisimulation: Formulas Φ, Ψ

S_1 vs. U_1

A finer initial partitioning

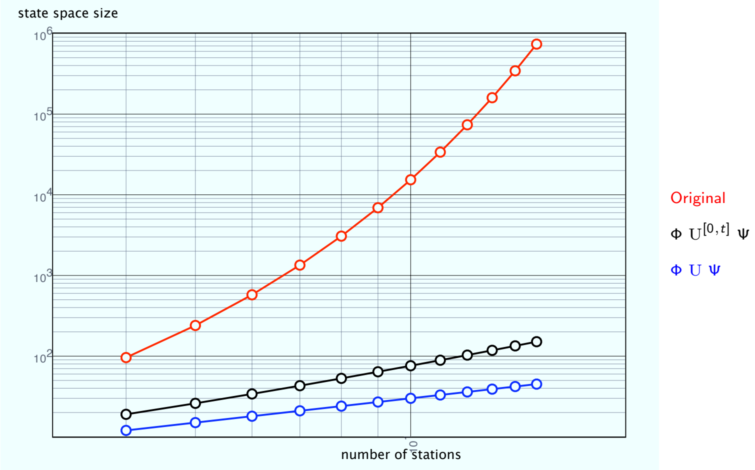
$\mathcal{P}_{\trianglelefteq p}(\Phi \ \mathrm{U} \ \Psi)$

- Define $U_0 = Sat\,(\mathcal{P}_{\leq 0}(\Phi \ \mathrm{U} \ \Psi))$.
- Define $U_1 = Sat\,(\mathcal{P}_{\geq 1}(\Phi \ \mathrm{U} \ \Psi))$.
- Choose $F = \{U_0,\ U_1,\ S \setminus (U_0 \cup U_1)\}$.
- Apply F-bisimulation.

$\mathcal{P}_{\trianglelefteq p}(\Phi \ \mathrm{U}^{[0,t]} \ \Psi)$

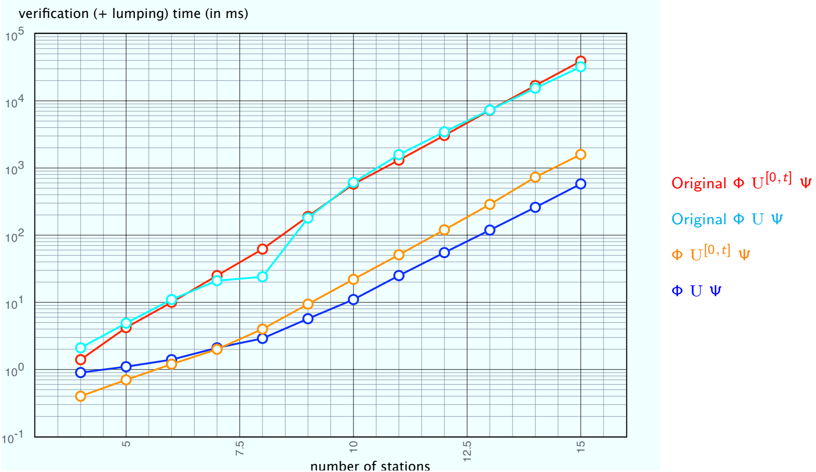
- Define $U_0 = Sat\,(\mathcal{P}_{\leq 0}(\Phi \ \mathrm{U} \ \Psi))$.
- Define $S_1 = Sat\,(\Psi)$.
- Choose $F = \{U_0,\ S_1,\ S \setminus (U_0 \cup S_1)\}$.
- Apply F-bisimulation.

Cyclic polling server (Ibe and Trivedi, 1990)



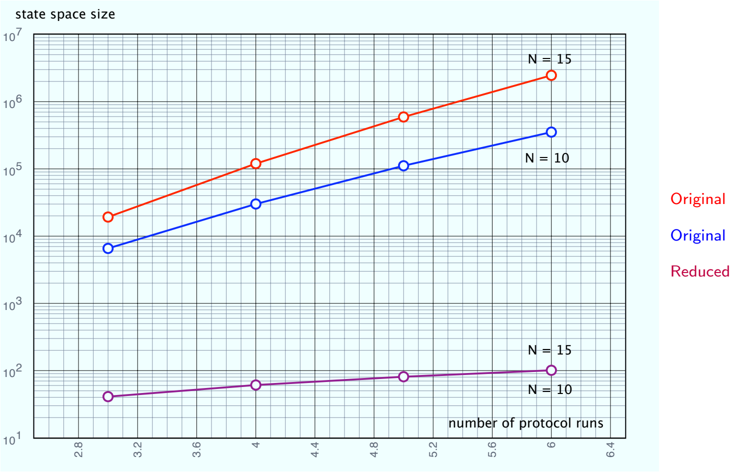
State-space reductions for $\mathcal{P}_{\leq q}(\neg serve_1 \ \mathrm{U}^{[0,1010]} \ serve_1)$ and $\mathcal{P}_{\leq q}(\neg serve_1 \ \mathrm{U} \ serve_1)$

Cyclic polling server (Ibe and Trivedi, 1990)



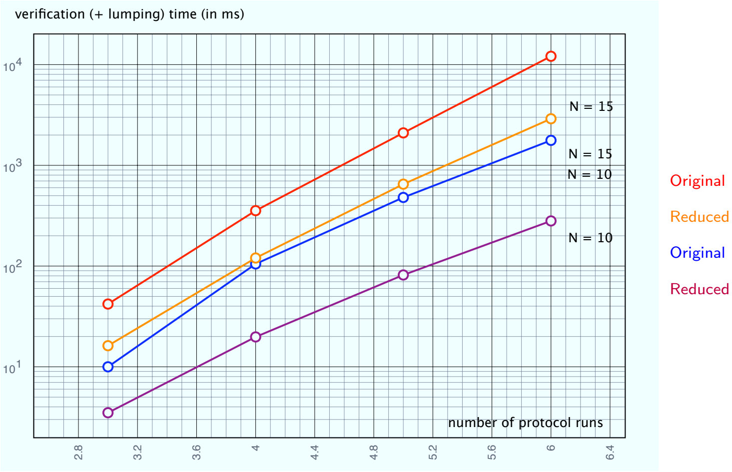
Run times for $\mathcal{P}_{\leq q}(\neg \mathit{serve}_1\ \mathrm{U}^{[0,1010]}\ \mathit{serve}_1)$ and $\mathcal{P}_{\leq q}(\neg \mathit{serve}_1\ \mathrm{U}\ \mathit{serve}_1)$

Crowds protocol (Reiter and Rubin, 1998)



State-space reductions for eventually observing the real sender more than once

Crowds protocol (Reiter and Rubin, 1998)



Run times for eventually observing the real sender more than once

Simple P2P protocol (Kwiatkowska et al., 2006)

			symmetry reduction (Kwiatkowska et al., 2006)				
original CTMC			reduced CTMC			red. factor	
N	states	ver. time	states	red. time	ver. time	states	time
2	1024	5.6	528	12	2.9	1.93	0.38
3	32768	410	5984	100	59	5.48	2.58
4	1048576	22000	52360	360	820	20.0	18.3

			bisimulation minimisation				
original CTMC			lumped CTMC			red. factor	
N	states	ver. time	blocks	lump time	ver. time	states	time
2	1024	5.6	56	1.4	0.3	18.3	3.3
3	32768	410	252	170	1.3	130	2.4
4	1048576	22000	792	10200	4.8	1324	2.2

Simple P2P protocol (Kwiatkowska et al., 2006)

			symmetry reduction (Kwiatkowska et al., 2006)				
original CTMC			reduced CTMC			red. factor	
N	states	ver. time	states	red. time	ver. time	states	time
2	1024	5.6	528	12	2.9	1.93	0.38
3	32768	410	5984	100	59	5.48	2.58
4	1048576	22000	52360	360	820	20.0	18.3

			bisimulation minimisation				
original CTMC			lumped CTMC			red. factor	
N	states	ver. time	blocks	lump time	ver. time	states	time
2	1024	5.6	56	1.4	0.3	18.3	3.3
3	32768	410	252	170	1.3	130	2.4
4	1048576	22000	792	10200	4.8	1324	2.2

Simple P2P protocol (Kwiatkowska et al., 2006)

			symmetry reduction (Kwiatkowska et al., 2006)				
original CTMC			reduced CTMC			red. factor	
N	states	ver. time	states	red. time	ver. time	states	time
2	1024	5.6	528	12	2.9	1.93	0.38
3	32768	410	5984	100	59	5.48	2.58
4	1048576	22000	52360	360	820	20.0	18.3

			bisimulation minimisation				
original CTMC			lumped CTMC			red. factor	
N	states	ver. time	blocks	lump time	ver. time	states	time
2	1024	5.6	56	1.4	0.3	18.3	3.3
3	32768	410	252	170	1.3	130	2.4
4	1048576	22000	792	10200	4.8	1324	2.2

Simple P2P protocol (Kwiatkowska et al., 2006)

			symmetry reduction (Kwiatkowska et al., 2006)				
original CTMC			reduced CTMC			red. factor	
N	states	ver. time	states	red. time	ver. time	states	time
2	1024	5.6	528	12	2.9	1.93	0.38
3	32768	410	5984	100	59	5.48	2.58
4	1048576	22000	52360	360	820	20.0	18.3

			bisimulation minimisation				
original CTMC			lumped CTMC			red. factor	
N	states	ver. time	blocks	lump time	ver. time	states	time
2	1024	5.6	56	1.4	0.3	18.3	3.3
3	32768	410	252	170	1.3	130	2.4
4	1048576	22000	792	10200	4.8	1324	2.2

The end

Concluding remarks

- Significant, up to logarithmic, state-space reduction.
- The abstraction technique is fully automated.
- Strong bisimulation:
  - Sometimes, a substantial model-checking time reduction.
  - Sometimes, an increase of peak memory (by 50%).
- F-bisimulation:
  - Sometimes, a substantial model-checking time reduction.
  - The peak memory use is typically unchanged.
  - For reward case a decrease of peak memory (by 20-40%).

Future work

- Combine symmetry reduction with bisimulation.
- Extend experiments towards MDPs and simulation preorders.

Andova, S., Hermanns, H., and Katoen, J.-P.: 2003,
in *Formal Modeling and Analysis of Timed Systems (FORMATS)*, LNCS, Marseille, France

Aziz, A., Sanwal, K., Singhal, V., Brayton, R. K., and Sangiovanni-Vincentelli: 1995,
in *Computer Aided Verification (CAV)*, pp 155–165, Berlin, Germany

Baier, C., Ciesinski, F., and Größer, M.: 2005a,
SIGMETRICS Perform. Eval. Rev. **32(4)**, 22

Baier, C., Haverkort, B., Hermanns, H., and Katoen, J.-P.: 2003,
IEEE Trans. on Softw. Eng. **29(6)**, 524

Baier, C., Haverkort, B. R., Hermanns, H., and Katoen, J.-P.: 2000,
in *International Colloquium on Automata, Languages and Programming (ICALP)*, pp 780–792, London, UK

Baier, C., Katoen, J.-P., Hermanns, H., and Wolf, V.: 2005b,
Inf. Comput. **200(2)**, 149

Bode, E., Herbstritt, M., Hermanns, H., Johr, S., Peikenkamp, T., Pulungan, R., Wimmer, R., and Becker, B.: 2006,
in *QEST '06: Proceedings of the Third International Conference on the Quantitative Evaluation of Systems - (QEST'06)*, pp 167–178, IEEE Computer Society, Washington, DC, USA

Buchholz, P.: 1994,
Journal of Applied Probability **31**, 59

D'Aprile, D., Donatelli, S., and Sproston, J.: 2004,
in *Int. Symp. on Computer and Information Sciences*, Vol. 3280 of *LNCS*, pp 543–552

D'Argenio, P. R., Jeannet, B., Jensen, H. E., and Larsen, K. G.: 2001,
in *PAPM-PROBMIV '01: Proceedings of the Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification*, pp 39–56, Springer-Verlag, London, UK

Derisavi, S., Hermanns, H., and Sanders, W. H.: 2003,
Inf. Process. Lett. **87(6)**, 309

Fisler, K. and Vardi, M. Y.: 1998,

in *FMCAD*, Vol. 1522 of *LNCS*, pp 115–132

Fisler, K. and Vardi, M. Y.: 1999,
in *CHARME*, Vol. 1703 of *LNCS*, pp 338–342

Fisler, K. and Vardi, M. Y.: 2002,
in *Formal Methods in System Design*, Vol. 21, pp 39–78

Hansson, N. and Jonsson, B.: 1994,
Formal Aspects of Computing **6**, 512

Hillston, J.: 1996,
A compositional approach to performance modelling,
Cambridge University Press, New York, NY, USA

Ibe, O. C. and Trivedi, K. S.: 1990,
in *IEEE J. on Selected Areas in Communications*, Vol. 8, pp 1649–1657

Katoen, J.-P., Khattri, M., and Zapreev, I. S.: 2005,
in *Quantitative Evaluation of Systems (QEST)*, pp 243–244

Kwiatkowska, M., Norman, G., and Parker, D.: 2004,
International Journal on Software Tools for Technology Transfer (STTT) **6(2)**, 128

Kwiatkowska, M., Norman, G., and Parker, D.: 2006,
in T. Ball and R. Jones (eds.), *Proc. 18th International Conference on Computer Aided Verification (CAV'06)*, Vol. 4114 of *LNCS*, pp 234–248, Springer-Verlag

Kwiatkowska, M., Norman, G., and Parker, D.: 2007,
http:www.cs.bham.ac.uk dxpprismcasestudies

Reiter, M. K. and Rubin, A.: 1998,
in *ACM Transactions on Information and System Security*, Vol. 1, pp 66–92