The What, Why, and How of Probabilistic Verification

Part 3: Towards Verifying Gigantic Markov Models

Joost-Pieter Katoen

RWTH AACHEN UNIVERSITY

UNIVERSITY OF TWENTE.

CAV Invited Tutorial 2015, San Francisco

Overview

Treating Gigantic Markov Models

Overview

A Real-Life Case Study @ esa

Crash Course on Satellite Internals

Crash Course on Satellite Internals

Payload is mission-specific, e.g.:

- telecom transponders,
- navigation signals,
- earth observation telemetry

Crash Course on Satellite Internals

Payload is mission-specific, e.g.:

- telecom transponders,
- navigation signals,
- earth observation telemetry

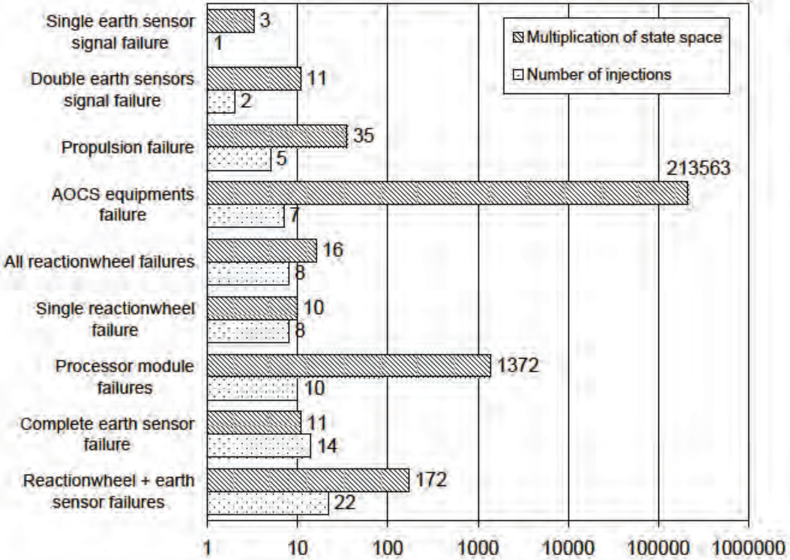
Platform keeps the satellite in space:

- attitude and orbital control
- power distribution
- data handling
- communication
- thermal regulation

AADL Model of Satellite Platform

Scope	Metric	Count
Model	Components	86
	Ports	937
	Modes	244
	Error models	20
	Recoveries	16
	Nominal state space	48421100
	LOC (without comments)	3831
Requirements	Propositional	25
	Absence	2
	Universality	1
	Response	14
	Probabilistic Invariance	1
	Probabilistic Existence	1

State Space Growth by Fault Injection



Conquering the State Space Explosion Problem

1. Symbolic approaches using (MT)BDDs PRISM
2. Bisimulation minimisation
3. Aggressive abstraction beyond bisimulation
4. Compositional abstraction
5. Confluence reduction (aka: partial-order reduction)
6. Exploit (multiple) multi-core processor(s)
7. ~~Resort to discrete event simulation~~[1]

[1] In modern terminology: statistical model checking.

Probabilistic Bisimulation [Larsen & Skou, 1989]

Probabilistic Bisimulation [Larsen & Skou, 1989]

Consider a DTMC with state space S and equivalence $R \subseteq S \times S$.
Then: R is a probabilistic bisimulation on S if for any $(s, t) \in R$:

1. $L(s) = L(t)$, and
2. $\mathbf{P}(s, C) = \mathbf{P}(t, C)$ for all equivalence classes $C \in S/R$

where $\mathbf{P}(s, C) = \sum_{s' \in C} \mathbf{P}(s, s')$.

Probabilistic Bisimulation [Larsen & Skou, 1989]

Consider a DTMC with state space S and equivalence $R \subseteq S \times S$.
Then: R is a probabilistic bisimulation on S if for any $(s, t) \in R$:

1. $L(s) = L(t)$, and
2. $\mathbf{P}(s, C) = \mathbf{P}(t, C)$ for all equivalence classes $C \in S/R$

where $\mathbf{P}(s, C) = \sum_{s' \in C} \mathbf{P}(s, s')$.

Let $\sim$ denote the largest possible probabilistic bisimulation.

Properties

Quotienting: using partition-refinement in $\mathcal{O}(|\mathbf{P}| \cdot \log |S|)$

Preservation: all probabilistic CTL*-formulas

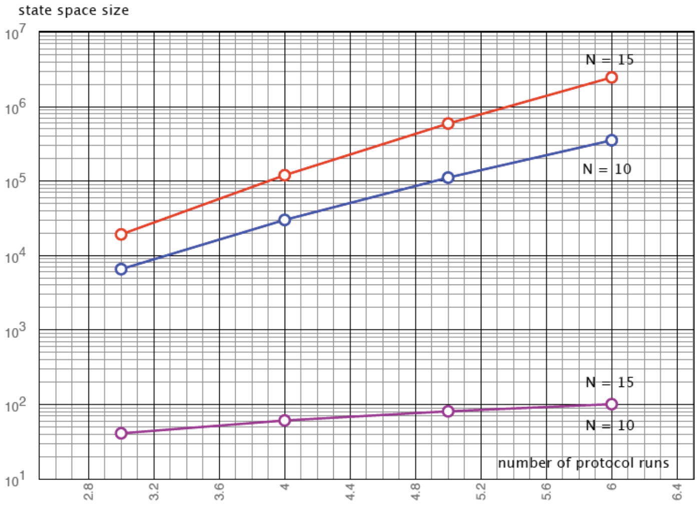
Congruence: with respect to parallel composition

Continuous: can all readily be adapted to CTMCs

Stuttering: weak variants are around and preserve PCTL* without next

Savings: potentially exponentially in time and space

Reducing Crowds Protocol [Reiter & Rubin, 1998]



Reducing IEEE 802.11 Group Communication Protocol

	original DTMC			quotient DTMC		red. factor	
OD	states	transitions	ver. time	blocks	total time	states	time
4	1125	5369	122	71	13	**15.9**	**9.00**
12	37349	236313	7180	1821	642	**20.5**	**11.2**
20	231525	1590329	50133	10627	5431	**21.8**	**9.2**
28	804837	5750873	195086	35961	24716	**22.4**	**7.9**
36	2076773	15187833	5103900	91391	77694	**22.7**	**6.6**
40	3101445	22871849	7725041	135752	127489	**22.9**	**6.1**

all times in milliseconds

Reducing BitTorrent-like P2P protocol

			symmetry reduction				
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

bisimulation may reduce a factor 66 after (manual) symmetry reduction

Principle of Compositional Minimisation

- Interactive Markov chains
  - mix of labeled transition systems and CTMCs
  - allow for compositional modeling
  - and non-determinism (aka: CTMDPs)

Principle of Compositional Minimisation

- Interactive Markov chains
  - mix of labeled transition systems and CTMCs
  - allow for compositional modeling
  - and non-determinism (aka: CTMDPs)

- Congruence property

$$(\mathcal{M}_1 \sim \mathcal{N}_1 \text{ and } \mathcal{M}_2 \sim \mathcal{N}_2) \quad \text{implies} \quad \mathcal{M}_1 \,||_A\, \mathcal{M}_2 \sim \mathcal{N}_1 \,||_A\, \mathcal{N}_2$$

Principle of Compositional Minimisation

- Interactive Markov chains
  - mix of labeled transition systems and CTMCs
  - allow for compositional modeling
  - and non-determinism (aka: CTMDPs)

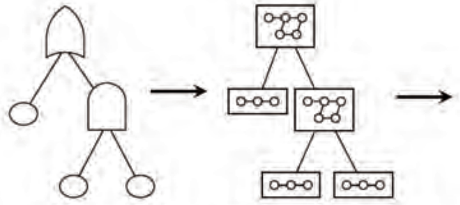
- Congruence property

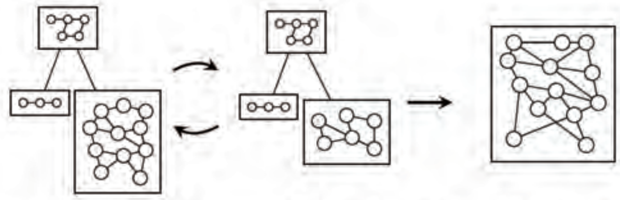
$$(\mathcal{M}_1 \sim \mathcal{N}_1 \text{ and } \mathcal{M}_2 \sim \mathcal{N}_2) \quad \text{implies} \quad \mathcal{M}_1 \,||_A\, \mathcal{M}_2 \sim \mathcal{N}_1 \,||_A\, \mathcal{N}_2$$

- Component-wise minimisation[a]
  1. Consider $\mathcal{M}_1 \,||_A \ldots ||_A\, \mathcal{M}_i \,||_A \ldots ||_A\, \mathcal{M}_k$
  2. Pick process $\mathcal{M}_i$ and consider its quotient under $\sim$
  3. Yielding $\mathcal{M}_1 \,||_A \ldots ||_A\, \mathcal{M}_i/\!\sim \,||_A \ldots ||_A\, \mathcal{M}_k$
  4. This can also be applied to groups of processes

[a] This paradigm is well-supported by the CADP tool.

Compositional Minimisation of DFTs


(a) DFT (b) Transformation


(c) Composition (d) Minimisation (e) IMC

Compositional Minimisation of DFTs

case study	peak # states	# transitions	unreliability	time (s)
CPS	4113	24608	.00135	490

CPS	133	465	.00135	67

Comparing Galileo DIFTree (top) to new approach (bottom)

Compositional Minimisation of DFTs

case study	peak # states	# transitions	unreliability	time (s)
CPS	4113	24608	.00135	490
CAS	8	10	.65790	1

CPS	133	465	.00135	67
CAS	36	119	.65790	94

Comparing Galileo DIFTree (top) to new approach (bottom)

Compositional Minimisation of DFTs

case study	peak # states	# transitions	unreliability	time (s)
CPS	4113	24608	.00135	490
CAS	8	10	.65790	1
CAS-PH	x	x	x	x
NDPS	x	x	x	x

case study	peak # states	# transitions	unreliability	time (s)
CPS	133	465	.00135	67
CAS	36	119	.65790	94
CAS-PH	40052	265442	.112	231
NDPS	61	169	[.00586, .00598]	266

Comparing Galileo DIFTree (top) to new approach (bottom)

Compositional Minimisation of DFTs

case study	peak # states	# transitions	unreliability	time (s)
CPS	4113	24608	.00135	490
CAS	8	10	.65790	1
CAS-PH	x	x	x	x
NDPS	x	x	x	x
FTTP-4	32757	426826	.01922	13111
FTTP-5	MO	MO	MO	MO

CPS	133	465	.00135	67
CAS	36	119	.65790	94
CAS-PH	40052	265442	.112	231
NDPS	61	169	[.00586, .00598]	266
FTTP-4	1325	13642	.01922	65
FTTP-6	11806565	22147378	.00045	1989

Comparing Galileo DIFTree (top) to new approach (bottom)

Compositional Minimisation of DFTs

case study	peak # states	# transitions	unreliability	time (s)
CPS	4113	24608	.00135	490
CAS	8	10	.65790	1
CAS-PH	x	x	x	x
NDPS	x	x	x	x
FTTP-4	32757	426826	.01922	13111
FTTP-5	MO	MO	MO	MO

CPS	133	465	.00135	67
CAS	36	119	.65790	94
CAS-PH	40052	265442	.112	231
NDPS	61	169	[.00586, .00598]	266
FTTP-4	1325	13642	.01922	65
FTTP-6	11806565	22147378	.00045	1989

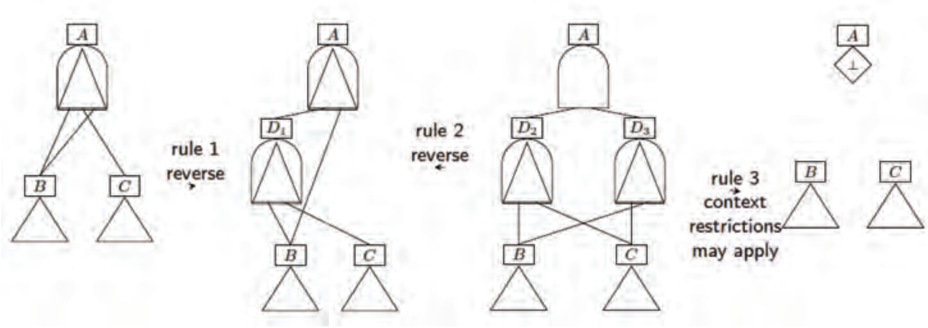
Comparing Galileo DIFTree (top) to new approach (bottom)

In practice, DFTs of >50 nodes are not an exception.

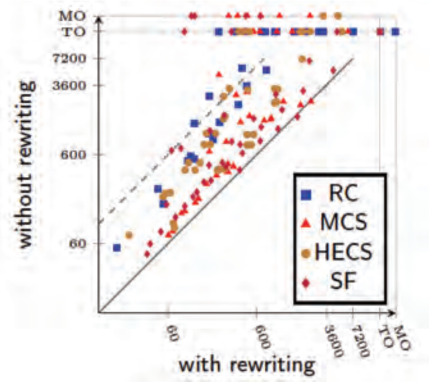
Tailored DFT Abstraction [Junges *et al.*, 2015]

Key idea

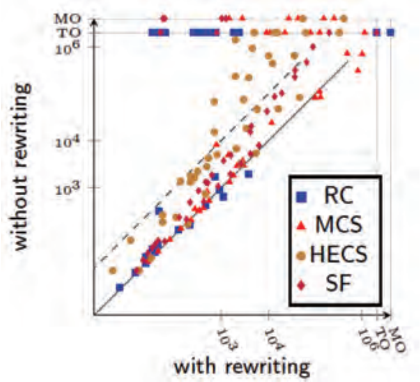
Simplify DFTs by graph rewriting prior to (compositional) state space generation.



Tailored DFT Abstraction



total verification and minimisation time



state space size of resulting CTMDP

49 out of 179 case studies could be treated now that could not be treated before

More Aggressive Abstraction [Katoen *et al.*, 2007]

- Partition the state space into groups of concrete states
  - allow any partitioning, not just grouping of bisimilar states

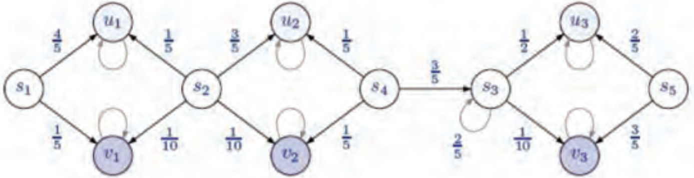
More Aggressive Abstraction [Katoen *et al.*, 2007]

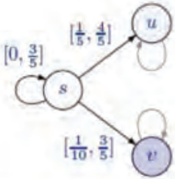
- Partition the state space into groups of concrete states
  - allow any partitioning, not just grouping of bisimilar states
- Use three-valued semantics
  - abstraction is conservative for both negative and positive results
  - if verification yields don't know, validity in concrete model is unknown

More Aggressive Abstraction [Katoen *et al.*, 2007]

- Partition the state space into groups of concrete states
  - allow any partitioning, not just grouping of bisimilar states
- Use three-valued semantics
  - abstraction is conservative for both negative and positive results
  - if verification yields don't know, validity in concrete model is unknown
- Important aspects:
  - ingredients of abstract probabilistic models
  - how to verify abstracts models?
  - how accurate are abstractions in practice?

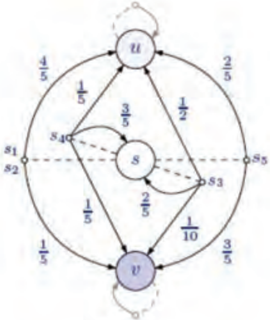
Intuition of Abstraction





⟵ Interval abstraction

CTMDP abstraction ⟶

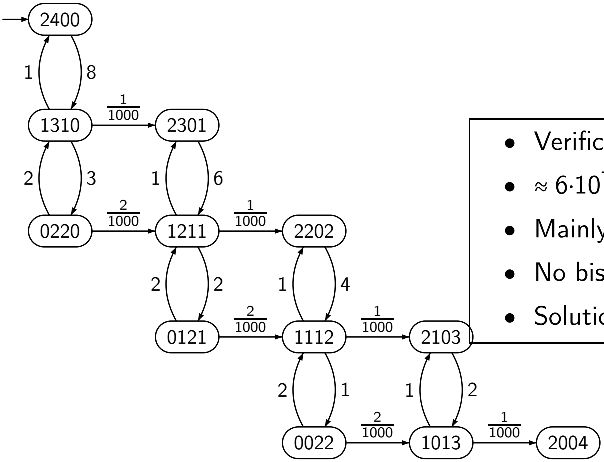


Theoretical Results on Abstraction

1. For a given state-space partitioning: abstract probabilistic model "simulates" concrete model (but not the converse)
2. If $s \sqsubseteq s'$ and $[\![\Phi]\!](s') \neq ?$ then: $[\![\Phi]\!](s') = [\![\Phi]\!](s)$ for any formula Φ in continuous stochastic logic (without next)
3. Extreme policies suffice for verifying interval-probabilistic models
4. Step-bounded and time-bounded reachability can be checked in polynomial time
5. Interval Markov chains + modal transition systems yields a useful and elegant framework for compositional abstraction
6. "Simulation" is a pre-congruence with respect to parallel composition, so:

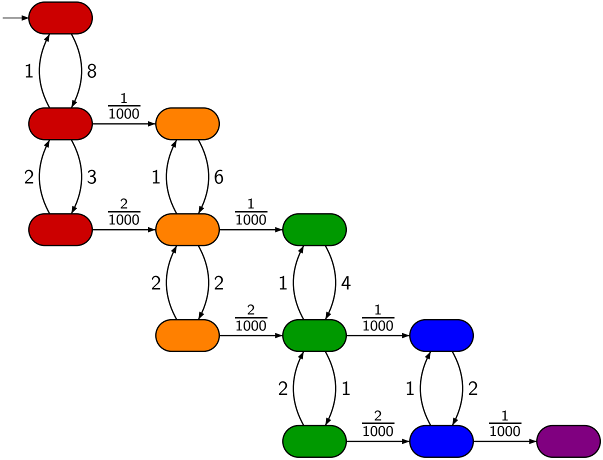
$$M_1 \sqsubseteq N_1 \text{ and } M_2 \sqsubseteq N_2 \implies M_1 \,||_A\, M_2 \sqsubseteq N_1 \,||_A\, N_2$$

Substrate Conversion



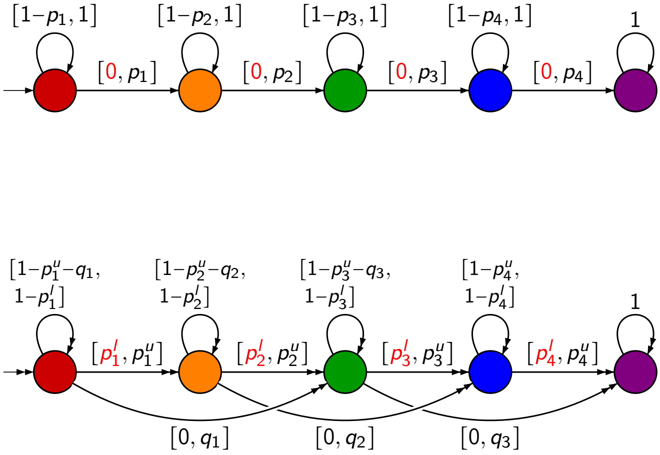
- Verification takes days
- $\approx 6{\cdot}10^7$ iterations needed
- Mainly due to stiffness
- No bisimilar states
- Solution: abstraction

Example: Substrate Conversion

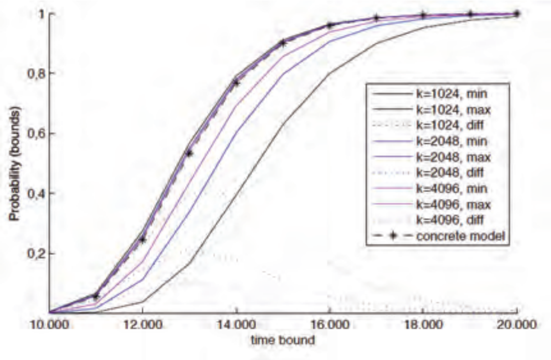


rule of thumb: group sets of "fast" connected states

Improving Lower Bounds



Model Checking The Abstraction



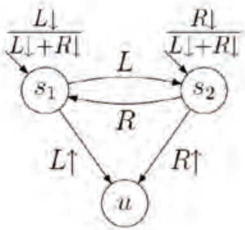
$\lvert A \rvert$	$\lvert S \rvert$	time
50	861	$0m$ $5s$
300	6111	$37m$ $36s$
500	10311	$70m$ $39s$
1000	20811	$144m$ $49s$
1500	31311	$214m$ $2s$
2000	41811	$322m$ $50s$

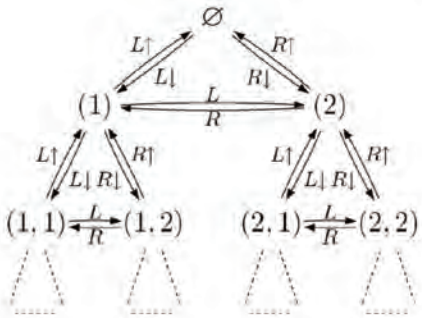
probability of only having products in deadline t (200 substrates, 20 enzymes)

results using Markov Chain Model Checker www.mrmc-tool.org

Example: Abstracting Queueing Networks

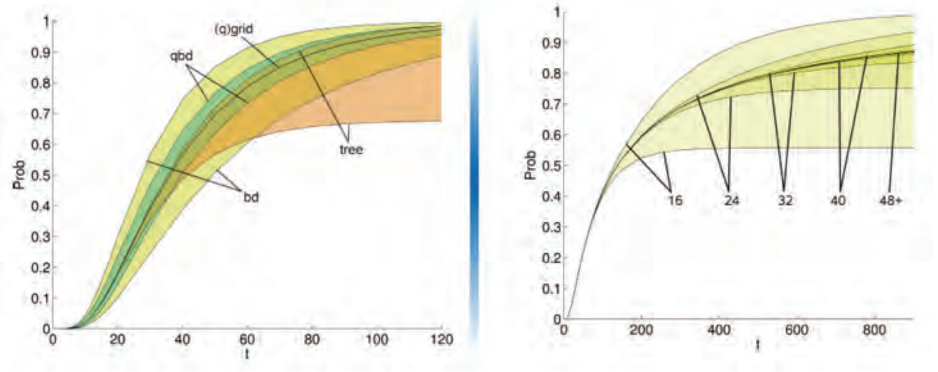
- Application: a M/PH$_n$/1 queueing station with preemptive scheduling
- Model: tree-based quasi-birth death (QBD) process
- Alternatively: a probabilistic push-down automaton
- Chance from a given configuration to serve up to k jobs within a deadline?





Experimental Results

Comparing different partitioning schemes and influence of cut level:



Experimental Results

Grid abstraction versus tree analysis techniques (error bound is 10^{-6}):

		grid abstraction							uniformization	
diff	grid 12	grid 16	grid 20	grid 24	grid 28	grid 32	grid 36	grid 40	trunc	≈ states
2.5	0.0224	0.001	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	185	10^{129}
t 10	0.3117	0.0580	0.0062	0.0004	10^{-5}	10^{-6}	10^{-6}	10^{-6}	270	10^{188}
15	0.4054	0.1345	0.0376	0.0086	0.0015	0.0002	$2 \cdot 10^{-5}$	$3 \cdot 10^{-6}$	398	10^{278}
states	6188	20349	53130	118755	237336	435894	749398	1221759		
distributions	28666	96901	256796	579151	1164206	2146761	3701296	6047091		
time (h:m:s)	0:00:26	0:01:33	0:04:15	0:09:50	0:20:14	0:38:13	1:07:57	2:06:04		

Experimental Results

Grid abstraction versus tree analysis techniques (error bound is 10^{-6}):

	grid abstraction								uniformization	
diff	grid 12	grid 16	grid 20	grid 24	grid 28	grid 32	grid 36	grid 40	trunc	$\approx$ states
t 2.5	0.0224	0.001	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	185	10^{129}
t 10	0.3117	0.0580	0.0062	0.0004	10^{-5}	10^{-6}	10^{-6}	10^{-6}	270	10^{188}
t 15	0.4054	0.1345	0.0376	0.0086	0.0015	0.0002	$2 \cdot 10^{-5}$	$3 \cdot 10^{-6}$	398	10^{278}
states	6188	20349	53130	118755	237336	435894	749398	1221759		
distributions	28666	96901	256796	579151	1164206	2146761	3701296	6047091		
time (h:m:s)	0:00:26	0:01:33	0:04:15	0:09:50	0:20:14	0:38:13	1:07:57	2:06:04		

$\Rightarrow$ Abstraction yields same accuracy by 1.2 million state as 10^{278} concrete ones

$\Rightarrow$ First time that tree-based QBDs of this size have been successfully analysed

Compositional Abstraction

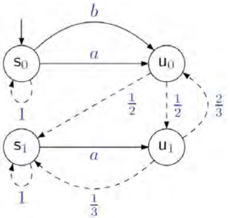
- Interactive Markov chains (IMCs)
  - mix of transition systems and CTMCs
  - allow for compositional modeling
  - and compositional minimisation

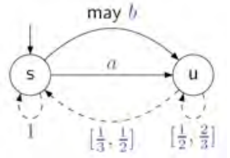
Compositional Abstraction

- Interactive Markov chains (IMCs)
  - mix of transition systems and CTMCs
  - allow for compositional modeling
  - and compositional minimisation
- Abstract IMCs
  - use interval abstraction
  - and modal transition systems (MTS)

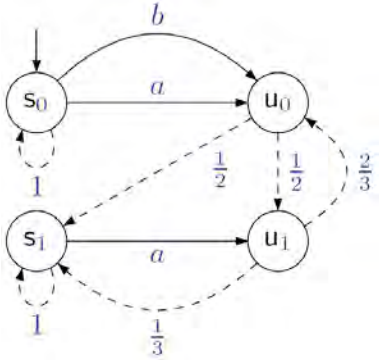
Compositional Abstraction

- Interactive Markov chains (IMCs)
  - mix of transition systems and CTMCs
  - allow for compositional modeling
  - and compositional minimisation
- Abstract IMCs
  - use interval abstraction
  - and modal transition systems (MTS)
- Aim: abstract component-wise
  - replace $\mathcal{M}_i$ by $\alpha(\mathcal{M}_i)$
  - then $\mathcal{M}_1 \| \ldots \| \mathcal{M}_n$ by $\alpha(\mathcal{M}_1) \| \ldots \| \alpha(\mathcal{M}_n)$

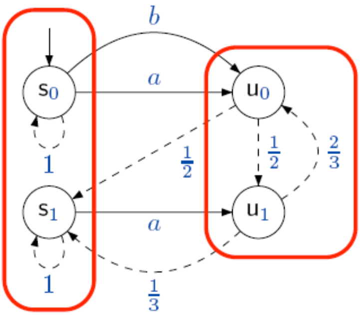




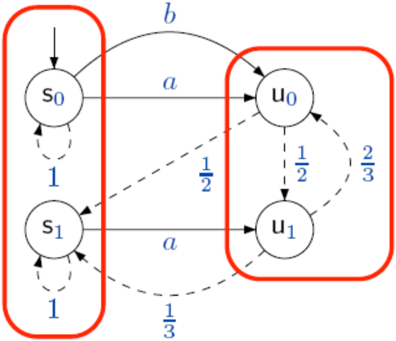
Compositional Abstraction

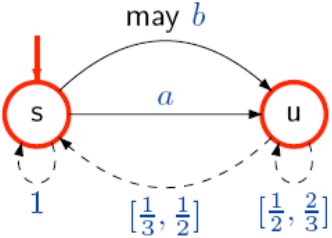


Compositional Abstraction

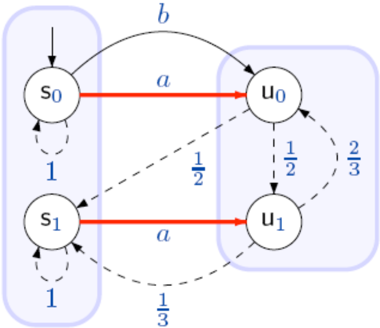


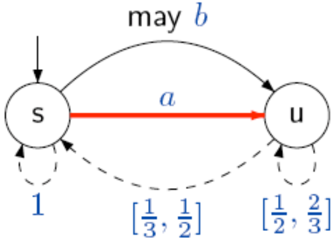
Compositional Abstraction



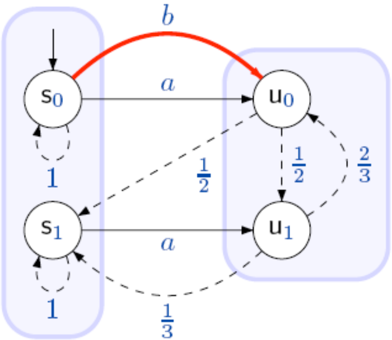


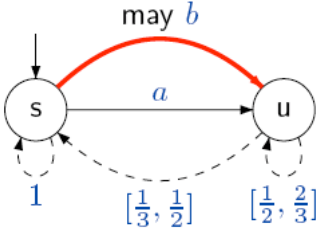
Compositional Abstraction



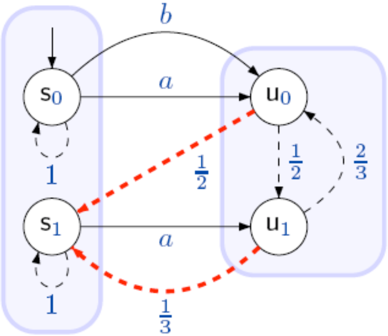


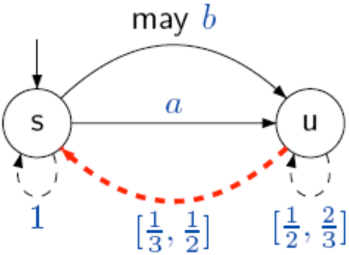
Compositional Abstraction



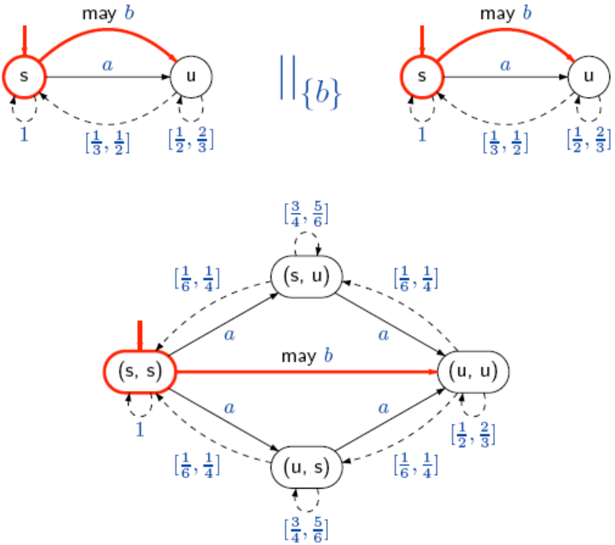


Compositional Abstraction

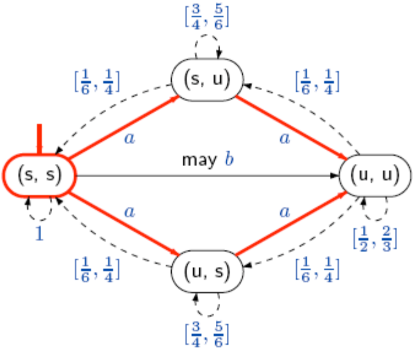




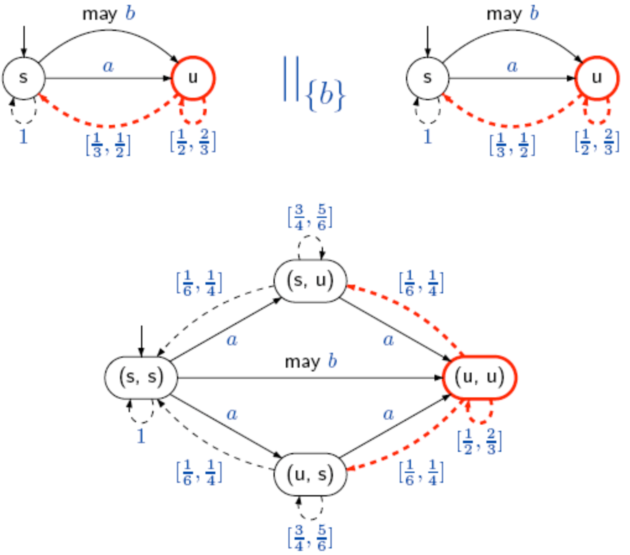
Parallel Composition



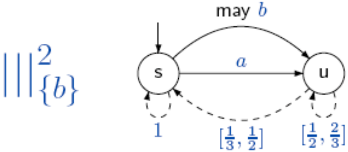
Parallel Composition

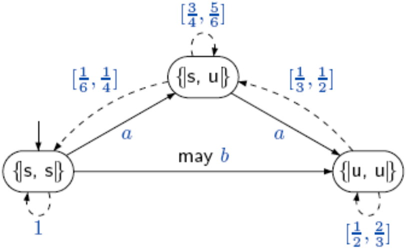


Parallel Composition



Symmetric Composition





Multisets representing tuples: $\{\!|s, u|\!\} \mathrel{\hat{=}} \{(s, u), (u, s)\}$

Theoretical Results

- Symmetric composition and parallel composition are bisimilar

$$|||_A^{n} \, \mathcal{M} \;\sim\; \underbrace{\mathcal{M} \,||_A \ldots ||_A \, \mathcal{M}}_{n \text{ times}}$$

Theoretical Results

- Symmetric composition and parallel composition are bisimilar

$$|||_A^n \mathcal{M} \sim \underbrace{\mathcal{M} \,||_A \ldots ||_A\, \mathcal{M}}_{n \text{ times}}$$

- Simulation is a pre-congruence wrt. $||$ and symmetric composition

$$\mathcal{M}_1 \sqsubseteq \mathcal{N}_1 \text{ and } \mathcal{M}_2 \sqsubseteq \mathcal{N}_2 \quad \text{implies} \quad \mathcal{M}_1 \,||_A\, \mathcal{M}_2 \sqsubseteq \mathcal{N}_1 \,||_A\, \mathcal{N}_2$$

Theoretical Results

- Symmetric composition and parallel composition are bisimilar

$$|||_A^{n}\, \mathcal{M} \;\sim\; \underbrace{\mathcal{M} \,||_A \ldots ||_A\, \mathcal{M}}_{n \text{ times}}$$

- Simulation is a pre-congruence wrt. $||$ and symmetric composition

$$\mathcal{M}_1 \sqsubseteq \mathcal{N}_1 \text{ and } \mathcal{M}_2 \sqsubseteq \mathcal{N}_2 \quad \text{implies} \quad \mathcal{M}_1 \,||_A\, \mathcal{M}_2 \sqsubseteq \mathcal{N}_1 \,||_A\, \mathcal{N}_2$$

- Bisimulation is a congruence wrt. $||$ and symmetric composition

Theoretical Results

- Symmetric composition and parallel composition are bisimilar

$$|||_A^{n} \mathcal{M} \sim \underbrace{\mathcal{M} \,||_A \ldots ||_A\, \mathcal{M}}_{n \text{ times}}$$

- Simulation is a pre-congruence wrt. $||$ and symmetric composition

$$\mathcal{M}_1 \sqsubseteq \mathcal{N}_1 \text{ and } \mathcal{M}_2 \sqsubseteq \mathcal{N}_2 \quad \text{implies} \quad \mathcal{M}_1 \,||_A\, \mathcal{M}_2 \sqsubseteq \mathcal{N}_1 \,||_A\, \mathcal{N}_2$$

- Bisimulation is a congruence wrt. $||$ and symmetric composition

- Abstracting many parallel "similar" components:

$$(\text{for all } i.\ \mathcal{M}_i \sqsubseteq \mathcal{N}) \quad \text{implies} \quad \mathcal{M}_1 \,||_A \ldots ||_A\, \mathcal{M}_n \sqsubseteq |||_A^{n} \mathcal{N}$$

A Production Example

- Workers $\mathcal{M}_i$ (8 states)
- Counting process $\mathcal{Q}$ (44 states)

$$(\mathcal{M}_1 \mid\mid_\emptyset \mathcal{M}_2 \mid\mid_\emptyset \mathcal{M}_3) \mid\mid_A \mathcal{Q} \qquad 22528 \text{ states}$$

- Replace $\mathcal{M}_i$ by abstract worker $\mathcal{N}$ (6 states)

$$(\mathcal{N} \mid\mid_\emptyset \mathcal{N} \mid\mid_\emptyset \mathcal{N}) \mid\mid_A \mathcal{Q} \qquad 9504 \text{ states}$$

- Exploit symmetry by using multisets:
  $\{\!|s, s, u|\!\}$ instead of (s, s, u), (s, u, s), (u, s, s)

$$(|||_\emptyset^3 \mathcal{N}) \mid\mid_A \mathcal{Q} \qquad 2464 \text{ states}$$

Confluence (aka: Partial-Order) Reduction [Timmer *et al.*, 2015]

- Confluence reduction in a nutshell
  - State space reduction technique based on commutativity of transitions
  - Remove spurious non-determinism resulting from independent $||$
  - Construct a subset of the invisible transitions satisfying the confluence restrictions.
  - Choose a representative state for each state in the original system.
  - Skip confluent transitions until reaching a representative state

Confluence (aka: Partial-Order) Reduction [Timmer *et al.*, 2015]

- Confluence reduction in a nutshell
  - State space reduction technique based on commutativity of transitions
  - Remove spurious non-determinism resulting from independent $||$
  - Construct a subset of the invisible transitions satisfying the confluence restrictions.
  - Choose a representative state for each state in the original system.
  - Skip confluent transitions until reaching a representative state

- Confluence links divergence-sensitive branching bisimilar states

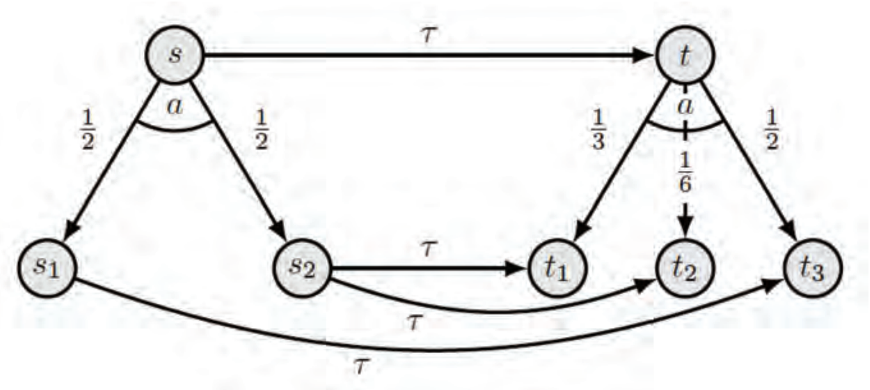
Confluence (aka: Partial-Order) Reduction [Timmer *et al.*, 2015]

- Confluence reduction in a nutshell
  - State space reduction technique based on commutativity of transitions
  - Remove spurious non-determinism resulting from independent $||$
  - Construct a subset of the invisible transitions satisfying the confluence restrictions.
  - Choose a representative state for each state in the original system.
  - Skip confluent transitions until reaching a representative state
- Confluence links divergence-sensitive branching bisimilar states
- Confluence is detected symbolically on model descriptions
  - based on conservative modelling-language-specific heuristics

Confluence (aka: Partial-Order) Reduction [Timmer *et al.*, 2015]

- Confluence reduction in a nutshell
  - State space reduction technique based on commutativity of transitions
  - Remove spurious non-determinism resulting from independent $||$
  - Construct a subset of the invisible transitions satisfying the confluence restrictions.
  - Choose a representative state for each state in the original system.
  - Skip confluent transitions until reaching a representative state
- Confluence links divergence-sensitive branching bisimilar states
- Confluence is detected symbolically on model descriptions
  - based on conservative modelling-language-specific heuristics
- On-the-fly reduction while generating the state space

Main Principle of Confluence Reduction



Experimental Results

	Original state space				Reduction		Impact	
Benchmark	$\lvert S \rvert$	$\lvert \mathbf{P} \rvert$	Gen.	Analysis	Gen.	Analysis	States	Time
le-3-7	25,505	34,257	4.7	103	5.1	9	**-78%**	-87%
le-3-9	52,465	71,034	9.7	212	10.4	18	**-79%**	-87%
le-3-11	93,801	127,683	18.0	429	19.2	32	**-79%**	-89%
le-4-3	35,468	50,612	9.0	364	8.7	33	**-78%**	-89%
le-4-4	101,261	148,024	25.8	1,310	24	94.4	**-79%**	-91%
poll-2-2-6	27,651	51,098	12.7	91	5.4	49	**-40%**	-48%
poll-2-5-2	27,659	47,130	4.0	1,572	4.0	1,054	**-29%**	-33%
poll-4-6-1	15,439	29,506	3.1	331	3.0	109	**-61%**	-66%
poll-5-4-1	21,880	43,760	5.1	816	5.1	318	**-71%**	-61%
proc-3	10,852	20,872	3.1	66	3.3	23	**-45%**	-62%
proc-4	31,832	62,356	10.8	925	10.3	366	**-45%**	-60%

2.4 GHz 4 GB Intel Core 2 Duo MacBook

CR removes 90% of states that are probabilistically branching bisimilar[2]

[2] Checked using the tool CADP.