Verifying Large (Infinite) Markov Chains

Joost-Pieter Katoen

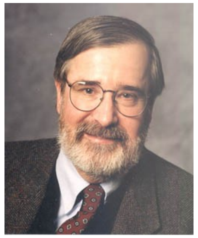
Software Modeling and Verification Group

RWTH Aachen University

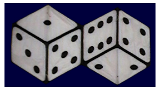
Nordic Workshop on Programming Theory, October 14, 2009

Model checking

- Automated model-based verification and debugging technique
  - model of system = Kripke structure $\approx$ labeled transition system
  - properties expressed in temporal logic like LTL or CTL
  - provides counterexamples in case of property refutation
- Various striking examples
  - Needham-Schroeder protocol, cache coherence, storm surge barrier, C code
- 2008: Pioneers awarded prestigious ACM Turing Award



- Today: model checking of probabilistic models



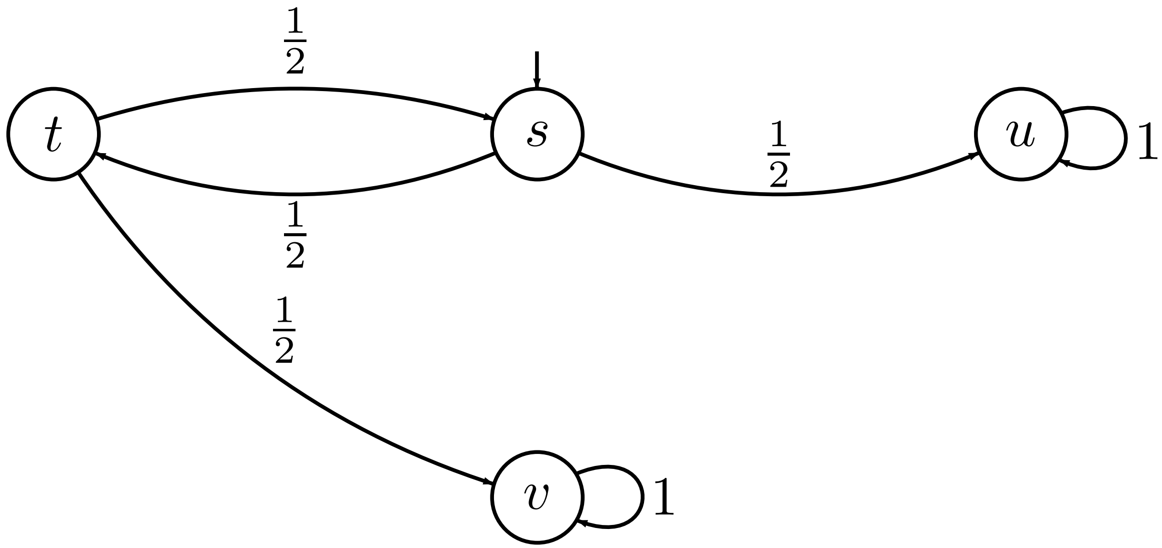
Probabilities help

- When analysing system performance and dependability
  - to quantify arrivals, waiting times, time between failure, QoS, ...
- When modelling uncertainty in the environment
  - to quantify imprecisions in system inputs
  - to quantify unpredictable delays, express soft deadlines, ...
- When building protocols for networked embedded systems
  - randomized algorithms
- When problems are undecidable deterministically
  - reachability of channel systems, ...

Probabilistic models

	Nondeterminism no	Nondeterminism yes
Discrete time	discrete-time Markov chain (DTMC)	Markov decision process (MDP)
Continuous time	CTMC	CTMDP

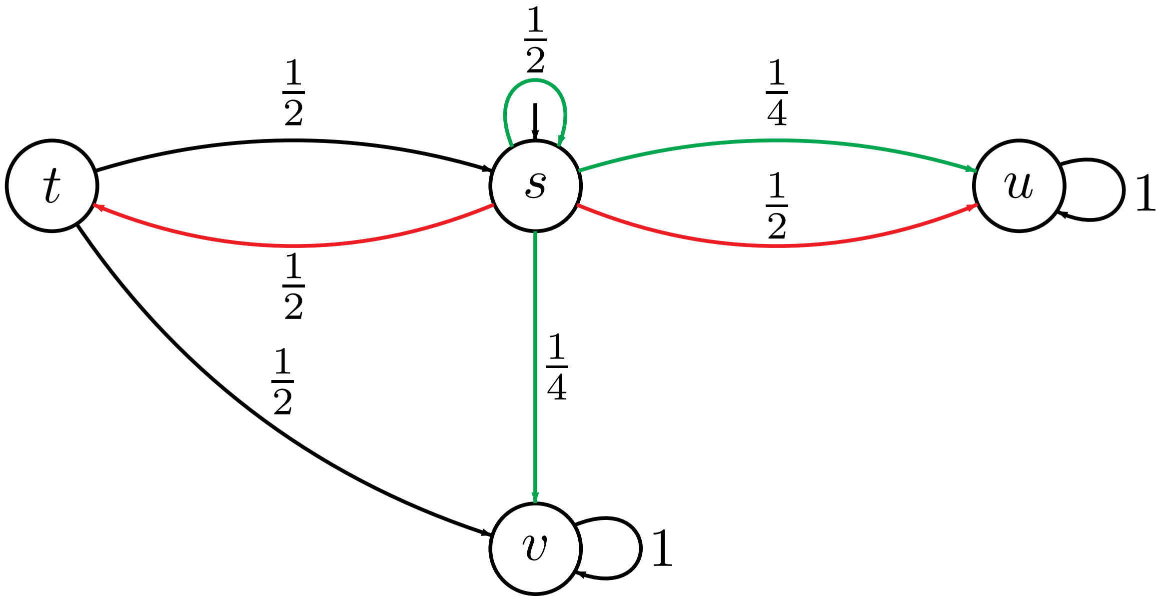
Discrete-time Markov chain



a DTMC is a triple $(S, \mathbf{P}, L)$ with state space S and state-labelling L

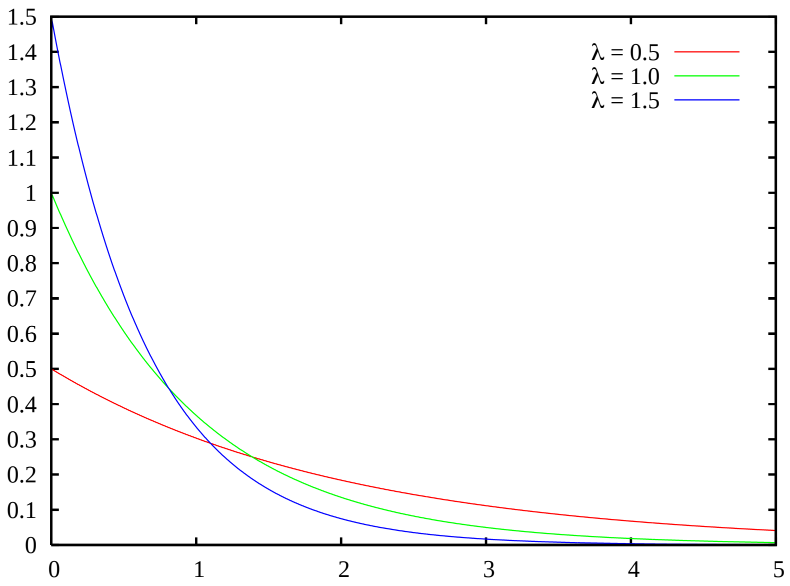
and $\mathbf{P}$ a stochastic matrix with $\mathbf{P}(s, s') =$ one-step probability to jump from s to s'

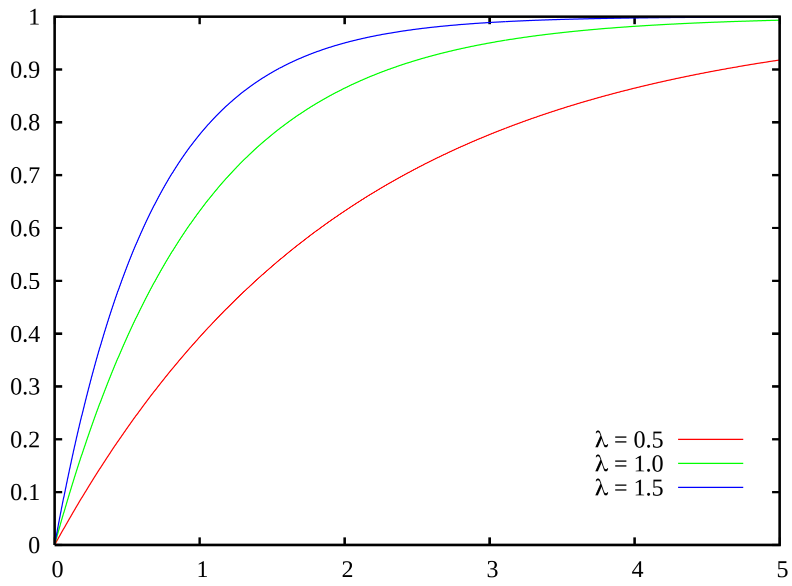
Markov decision process



an MDP is a DTMC if in each state there is only one color to choose

Exponential pdf and cdf





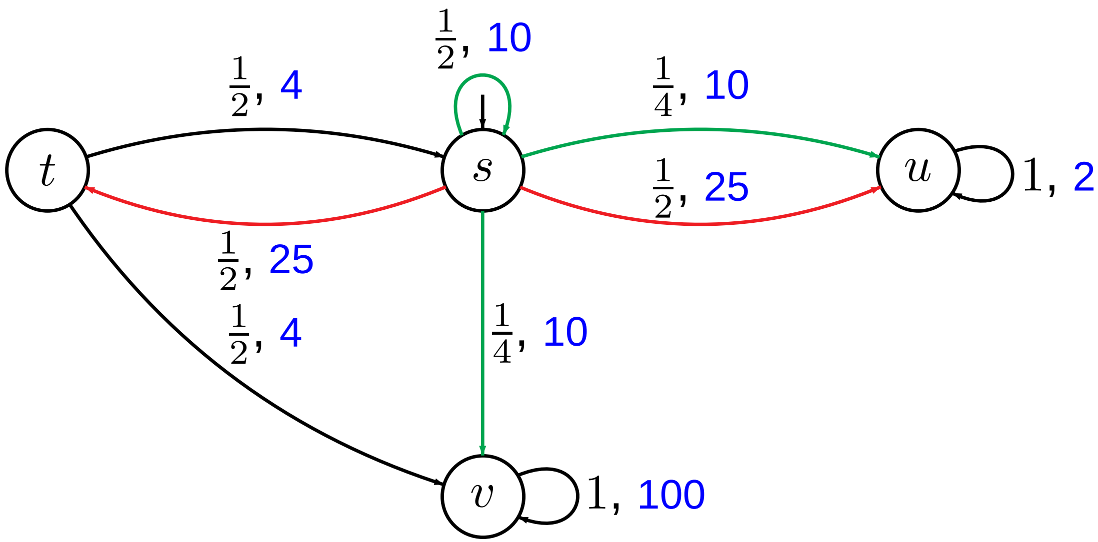
$$F_X(d) \;=\; \int_0^d \lambda \cdot e^{-\lambda \cdot x}\, dx \;=\; \left[-e^{-\lambda \cdot x}\right]_0^d \;=\; 1 - e^{-\lambda \cdot d}$$

Exponential distributions

- Are *adequate* for many real-life phenomena
  - inter-arrival times of jobs, telephone calls, and so on
- Are *memoryless*: $\Pr\{X > t + d \mid X > t\} \; = \; \Pr\{X > d\}$
- Elementary *properties*:
  - $\min(X, Y)$ is exponentially distributed with rate $\lambda + \mu$
  - $\Pr\{X = \min(X, Y)\} = \frac{\lambda}{\lambda + \mu}$
- Can *approximate* general distributions arbitrarily closely
- *Maximal entropy* if only the mean is known

Continuous-time MDP

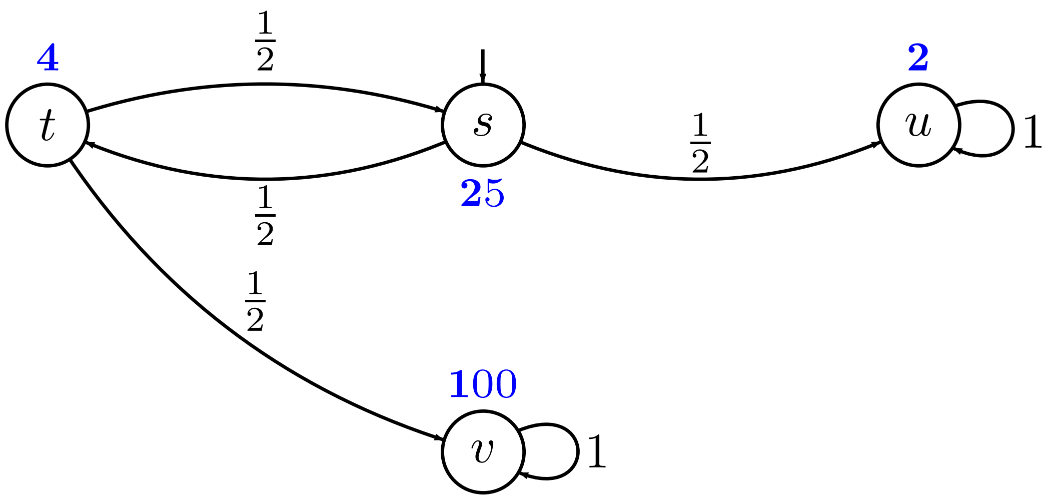
a CTMDP is an MDP plus an exit-rate function $r : S \times \mathit{Act} \longrightarrow \mathbb{R}_{\geqslant 0}$



note: when removing exit rates, an embedded MDP is obtained

Continuous-time Markov chain

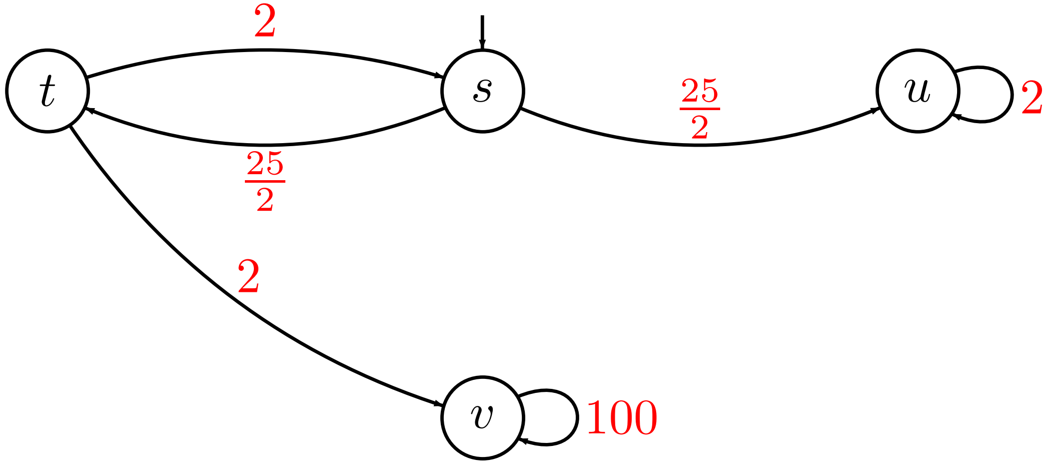
a CTMC $(S, \mathbf{P}, r, L)$ is a DTMC plus an exit-rate function $r : S \rightarrow \mathbb{R}_{\geqslant 0}$



the average residence time in state s is $\dfrac{1}{r(s)}$

An alternative perspective

a CTMC is a triple $(S, \mathbf{R}, L)$ with $\mathbf{R}(s, s') = \mathbf{P}(s, s') \cdot r(s)$



Modeling techniques for CTMCs

- Stochastic Petri nets (Molloy 1977)
- Markovian queueing networks (Kleinrock 1975)
- Stochastic activity networks (Meyer & Sanders 1985)
- Stochastic process algebra (Herzog *et al.*, Hillston 1993)
- Probabilistic input/output automata (Smolka *et al.* 1994)
- Calculi for biological systems (Priami *et al.*, Cardelli 2002)

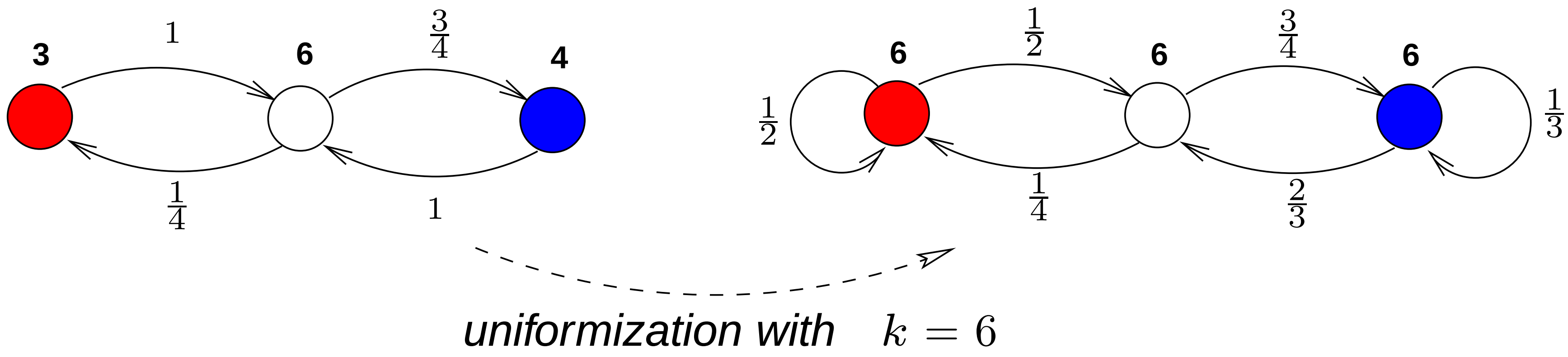
CTMCs are one of the most prominent models in performance analysis

Uniform CTMCs

- A CTMC is uniform if $r(s) = r$ for all s for some $r \in \mathbb{R}_{>0}$
- Any CTMC can be changed into a weak bisimilar uniform CTMC
- Let $r \in \mathbb{R}_{>0}$ such that $r \geqslant \mathit{max}_{s \in S}\, r(s)$
  - $\frac{1}{r}$ is at most the shortest mean residence time in CTMC $\mathcal{C}$
- Then $u_r(\mathcal{C}) = (S, \overline{\mathbf{P}}, \overline{r}, L)$ with $\overline{r}(s) = r$ for any s, and:

$$\overline{\mathbf{P}}(s,s') = \frac{r(s)}{r} \cdot \mathbf{P}(s,s') \text{ if } s' \neq s \quad \text{and} \quad \overline{\mathbf{P}}(s,s) = \frac{r(s)}{r} \cdot \mathbf{P}(s,s') + 1 - \frac{r(s)}{r}$$

Uniformization



all state transitions in $u_r(\mathcal{C})$ occur at an average pace of r per time unit

Timed reachability (Baier, Katoen & Hermanns, 1999)

- $[\![\, \mathbb{P}_J(\Diamond^{\leqslant t}\, \Psi)\,]\!](s) = \top$ if and only if $\Pr\{s \models \Diamond^{\leqslant t}\, \Psi\} \in J$
- $\Pr(s \models \Diamond^{\leqslant t}\, \Psi)$ is the least solution of:
  - 1 if $s \models \Psi$
  - otherwise:

$$\int_0^t \sum_{s' \in S} \mathbf{P}(s, s', x) \cdot \Pr(s' \models \Diamond^{\leqslant t-x}\, \Psi)\, dx$$

- Reduction to well-studied problem allows efficient, stable computation

Reachability probabilities

	Nondeterminism no	Nondeterminism yes
Reachability	linear equation system DTMC	linear programming MDP
Timed reachability	transient analysis (+ uniformization) CTMC	greedy backward reachability uniform CTMDP

Probabilistic bisimulation

- . . . coincides with CSL equivalence
  - $s \sim s' \;\Leftrightarrow\; \big(\forall \Phi \in \textit{CSL} : s \models \Phi$ if and only if $s' \models \Phi\big)$
- . . . its coarsest quotient can be obtained in $\mathcal{O}(|\mathbf{P}| \cdot \log |S|)$
- . . . may be tailored to property of interest

$\Rightarrow$ . . . offers fully automated and efficient abstraction

- . . . but for LTL/CTL minimization effort $\gg$ verification time

Probabilistic bisimulation (Kemeny & Snell, 1962), (Larsen & Skou, 1989)

- Let $\mathcal{C} = (S, \mathbf{P}, r, L)$ be a CTMC and R an equivalence on S
- R is a *strong bisimulation* on S if for any $(s, s') \in R$:

$$L(s) \; = \; L(s') \quad \text{and} \quad r(s) \; = \; r(s') \quad \text{and}$$

$$\mathbf{P}(s, C) \; = \; \mathbf{P}(s', C) \quad \text{for all} \quad C \in S/R$$

where $\mathbf{P}(s, C) \; = \; \sum_{u \in C} \mathbf{P}(s, u)$

- $s \sim s'$ iff $\exists$ a strong bisimulation R on S with $(s, s') \in R$

IEEE 802.11 group communication protocol

	original CTMC			lumped CTMC		red. factor	
OD	states	transitions	ver. time	blocks	lump + ver. time	states	time
4	1125	5369	121.9	71	13.5	**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**

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

Can we abstract more?

- Partition the state space into groups of concrete states
  - allow any partitioning, not just grouping of bisimilar states
- Use a three-valued semantics
  - abstraction is conservative for *both* negative and positive verification results
  - if verification yields *don't know*, validity in concrete model is unknown
- Challenges:
  - what are abstract probabilistic models?
  - how to interpret PCTL/CSL on these abstract models?
  - how to verify abstractions?
  - how accurate are abstractions in practice?

The discrete-time setting

An abstract MC (AMC) is a quintuple $\mathcal{D} = (S, \mathbf{P}^l, \mathbf{P}^u, L)$ with:

- $\mathbf{P}^l, \mathbf{P}^u : S \times S \mapsto [0,1]$, transition probability bounds where

$$\mathbf{P}^l(s, S) \leqslant 1 \leqslant \mathbf{P}^u(s, S) < \infty \quad \text{for all } s \in S$$

- $L : S \times \mathit{AP} \mapsto \{\, \top, \bot, ? \,\}$, the labeling function

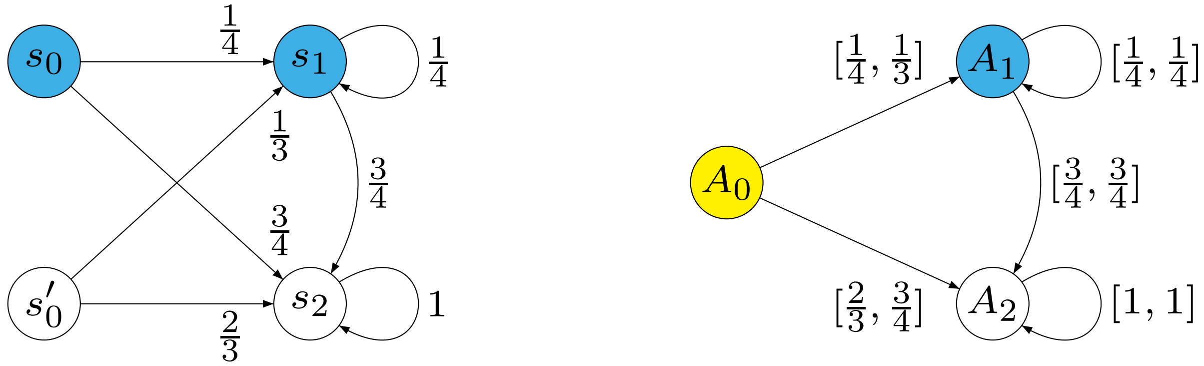
This is also known as interval Markov chains (Kozine & Utkin, 2002)

Abstraction

For $\mathcal{A} = \{ A_1, \ldots, A_n \}$ let AMC $\alpha(\mathcal{A}, \mathcal{D}) := (\mathcal{A}, \tilde{\mathbf{P}}^l, \tilde{\mathbf{P}}^u, \tilde{L})$ with:

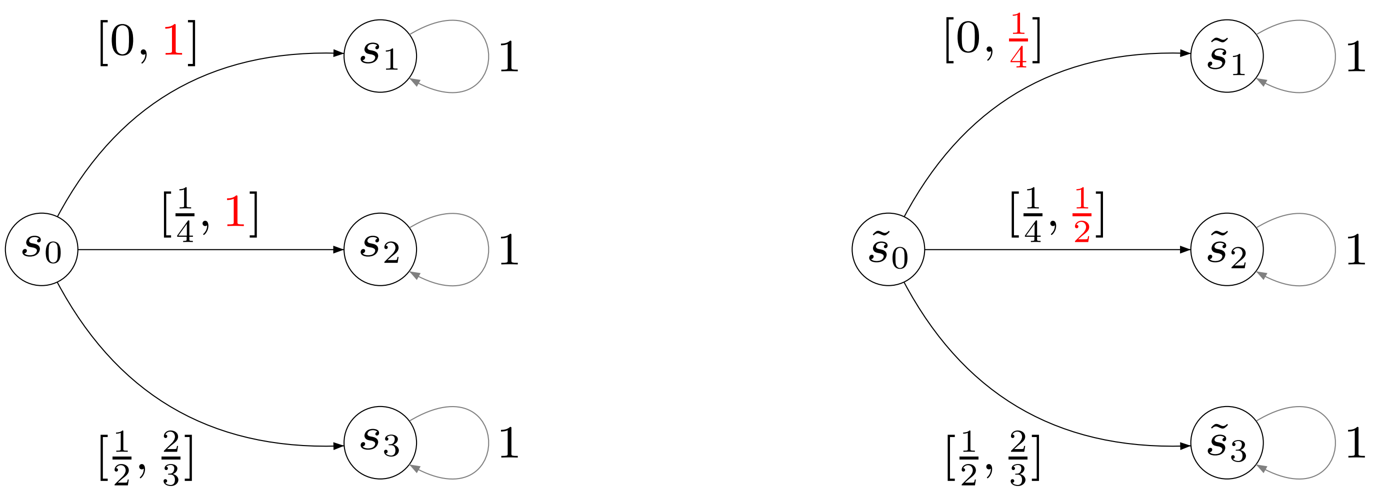
$$\tilde{\mathbf{P}}^l(A_i, A_j) = \inf_{s \in A_i} \mathbf{P}^l(s, A_j) \quad \text{and} \quad \tilde{\mathbf{P}}^u(A_i, A_j) = \min\{ 1, \sup_{s \in A_i} \mathbf{P}^u(s, A_j) \}$$

and $\tilde{L}(A_i, a) = \begin{cases} \top & \text{if } L(s, a) = \top \text{ for all } s \in A_i \\ \perp & \text{if } L(s, a) = \perp \text{ for all } s \in A_i \\ ? & \text{otherwise} \end{cases}$



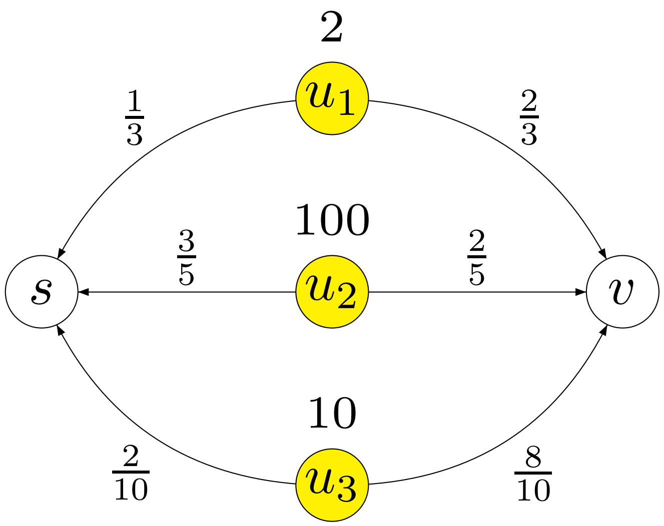
Normalization

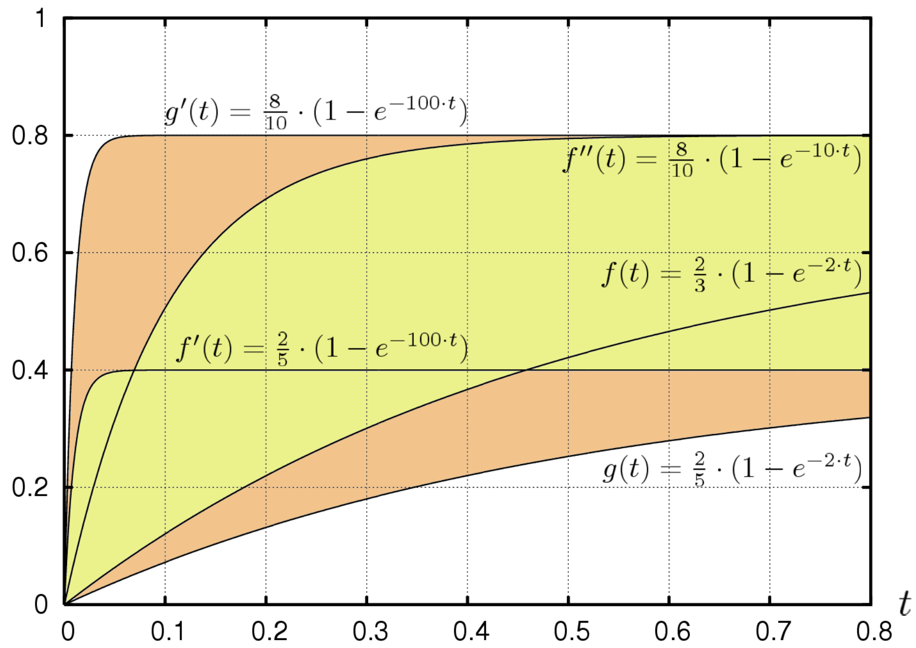
removes illegal probability combinations



an AMC is normalized if for each pair (s, s') and $p \in [\mathbf{P}^l(s, s'), \mathbf{P}^u(s, s')]$ there exists a distribution μ with $\mu(s') = p$

The continuous-time setting





$\mathbf{P}(A_u, A_v) = \underbrace{\{ f, f', f'' \}}_{F}$; but $\sup F$ and $\inf F$ are unequal $p \cdot (1 - e^{-k \cdot t})$

Uniformize before abstraction!

- For uniform CTMCs all exit rates are equal to r (say), and thus:

$$\begin{array}{rcl} \inf_{s \in A} \mathbf{P}(s, B, t) & = & (1 - e^{-r \cdot t}) \cdot \inf_{s \in A} \mathbf{P}(s, B) = (1 - e^{-r \cdot t}) \cdot p_l \\ & \leqslant & (1 - e^{-r \cdot t}) \cdot \sup_{s \in A} \mathbf{P}(s, B) = (1 - e^{-r \cdot t}) \cdot p_u \\ & = & \sup_{s \in A} \mathbf{P}(s, B, t) \end{array}$$

  - p_l, p_u are lower and upper bounds of time-independent transition probabilities

- Recall that any CTMC $\mathcal{C}$ can be turned into a uniform CTMC $u_r(\mathcal{C})$
  - in linear time while preserving CSL (no next) formulas as $\mathcal{C} \approx u_r(\mathcal{C})$

Correctness: simulation relation

- Let $\mathcal{C} = (S, \mathbf{P}, r, L)$ be a CTMC and R a binary relation on S
- R is a *simulation* relation on S if for all $(s, s') \in R$:

  $L(s) = L(s')$ and $\mathbf{P}(s, \cdot) \sqsubseteq_R \mathbf{P}(s', \cdot)$ and $r(s) \leqslant r(s')$

- s' simulates s, denoted $s \sqsubseteq s'$, if

  there exists a simulation relation R on S such that $(s, s') \in R$

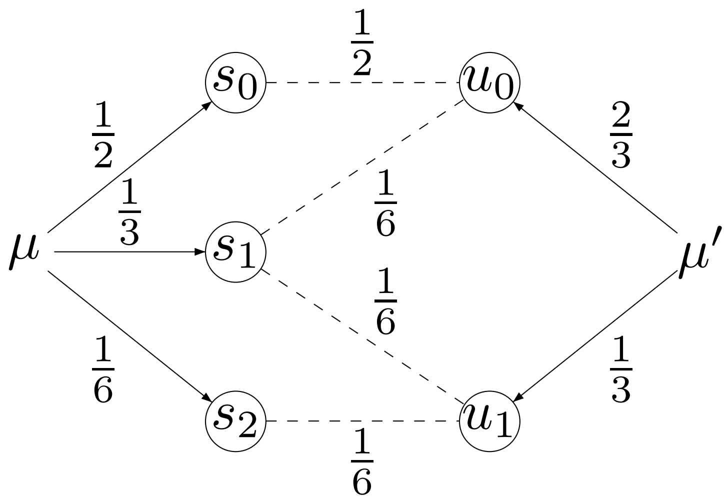
it remains to show how $\sqsubseteq_R$ between distributions is defined

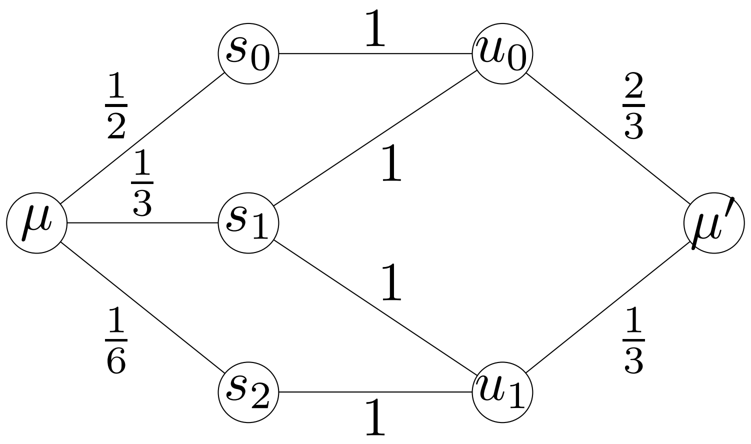
Weight function (Jones & Plotkin, 1990)

- Let S be a countable set, $R \subseteq S \times S$, and $\mu, \mu' \in$ *Dist*(S)
- $\Delta \in$ *Dist*$(S \times S)$ is a *weight function* for μ and μ' wrt. R if:
  - $\Delta(s, s') > 0$ implies $(s, s') \in R$
  - $\mu(s) = \sum_{s' \in S} \Delta(s, s')$ for any $s \in S$
  - $\mu'(s') = \sum_{s \in S} \Delta(s, s')$ for any $s' \in S$
- $\mu \sqsubseteq_R \mu'$ iff there exists a weight function for (μ, μ') wrt. R
  - or, equivalently, the maximal flow in a flow network equals one

$\sqsubseteq_R$ is the lifting of R (on states) to distributions

Simulation as maximal flow





Correctness (Katoen et al., 2007)

For AMC $\mathcal{C}$ with state space S, and partitioning $\mathcal{A}$ of S:

$$\mathcal{C} \sqsubseteq \alpha(\mathcal{A}, \mathcal{C})$$

For states s and s' of AMC $\mathcal{C}$ with $s \sqsubseteq s'$:

$$\forall \Phi \in \text{CSL} : [\![\Phi]\!](s') \neq ? \quad \text{implies} \quad [\![\Phi]\!](s) = [\![\Phi]\!](s')$$

Policies

- A policy resolves the nondeterminism as given by the intervals
  - consider time-abstract, history-dependent deterministic policies
  - there are infinitely many of such policies
  - on an AMC, such policies induce an (infinite-state) continuous-time Markov chain
- Extreme policies only select bounds of intervals
  - there are finitely many (possibly exponentially many) of such policies

For any measurable event E (in the σ-algebra on infinite paths):

$$\inf_{\text{extreme } \mathfrak{S}} \Pr^{\mathfrak{S}}(E) = \inf_{\text{any } \mathfrak{S}} \Pr^{\mathfrak{S}}(E) \quad \text{and} \quad \sup_{\text{extreme } \mathfrak{S}} \Pr^{\mathfrak{S}}(E) = \sup_{\text{any } \mathfrak{S}} \Pr^{\mathfrak{S}}(E)$$

Reachability probabilities

For $\mathcal{C} \subseteq \mathcal{C}'$ and compatible sets $G \subseteq S$, $G' \subseteq S'$
there exists for any policy $\mathfrak{S}$ on $\mathcal{C}$ a policy $\mathfrak{S}'$ on $\mathcal{C}'$ such that:

$$\Pr^{\mathfrak{S}}(\Diamond^{\leqslant k} G) = \Pr^{\mathfrak{S}'}(\Diamond^{\leqslant k} G') \quad \text{for any } k \in \mathbb{N}$$

$$\Pr^{\mathfrak{S}}(\Diamond^{\leqslant t} G) = \Pr^{\mathfrak{S}'}(\Diamond^{\leqslant t} G') \quad \text{for any } t \in \mathbb{R}_{\geqslant 0}$$

computing (time-)bounded probabilities is as in (CT)MDPs

Application 1

Systems biology: substrate conversion

Enzyme-catalysed substrate conversion

Enzyme - Wikipedia, the free encyclopedia - Mozilla Firefox

File Edit View History Bookmarks Tools Help

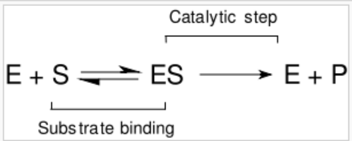
http://en.wikipedia.org/wiki/Enzyme

reaction, the reaction is *effectively* irreversible. Under these conditions the enzyme will, in fact, only catalyze the reaction in the thermodynamically allowed direction.

stabilizes the transition state, reducing the energy needed to form this species and thus reducing the energy required to form products.

Kinetics

Main article: Enzyme kinetics



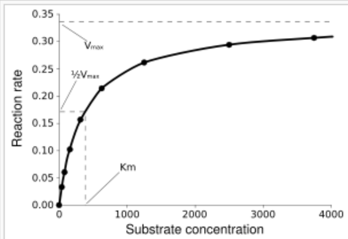
Mechanism for a single substrate enzyme catalyzed reaction. The enzyme (E) binds a substrate (S) and produces a product (P).

Enzyme kinetics is the investigation of how enzymes bind substrates and turn them into products. The rate data used in kinetic analyses are obtained from enzyme assays.

In 1902 Victor Henri [45] proposed a quantitative theory of enzyme kinetics, but his experimental data were not useful because the significance of the hydrogen ion concentration was not yet appreciated. After Peter Lauritz Sørensen had defined the logarithmic pH-scale and introduced the concept of buffering in 1909[46] the German chemist Leonor Michaelis and his Canadian postdoc Maud Leonora Menten repeated Henri's experiments and confirmed his equation which is referred to as Henri-Michaelis-Menten kinetics (sometimes also Michaelis-Menten kinetics).[47] Their work was further developed by G. E. Briggs and J. B. S. Haldane, who derived kinetic equations that are still widely used today.[48]

The major contribution of Henri was to think of enzyme reactions in two stages. In the first, the substrate binds reversibly to the enzyme, forming the enzyme-substrate complex. This is sometimes called the Michaelis complex. The enzyme then catalyzes the chemical step in the reaction and releases the product.

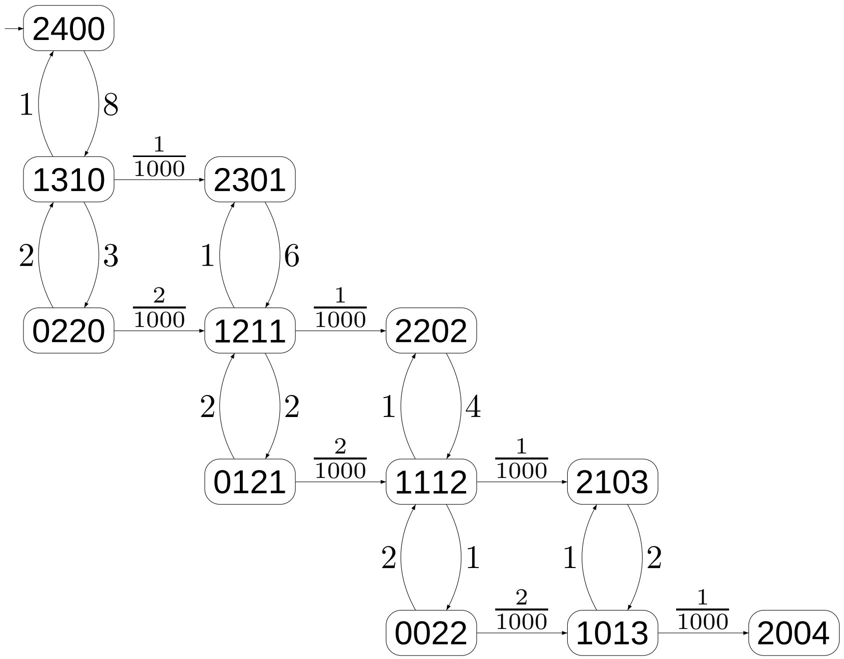
Enzymes can catalyze up to several million reactions per second. For example, the reaction catalyzed by orotidine 5'-phosphate decarboxylase will consume half of its substrate in 78 million years if no enzyme is present. However, when the decarboxylase is added, the same process takes just 25 milliseconds.[49] Enzyme rates depend on solution conditions and substrate concentration. Conditions that denature the protein abolish enzyme activity, such as high temperatures, extremes of pH or high salt concentrations, while raising substrate concentration tends to increase activity. To find the maximum speed of an enzymatic reaction, the substrate concentration is increased until a constant rate of product formation is seen. This is shown in the saturation curve on the right. Saturation happens because, as substrate concentration increases, more and more of the free enzyme is converted into the substrate-bound ES form. At the maximum velocity (V_{max}) of the enzyme, all the enzyme active sites are bound to substrate, and the amount of ES complex is the same as the total amount of enzyme. However, V_{max} is only one kinetic constant of enzymes. The amount of substrate needed to achieve a given rate of reaction is also important. This is given by the Michaelis-Menten constant (K_m), which is the substrate concentration required for an enzyme to reach one-half its maximum velocity. Each enzyme has a characteristic K_m for a given substrate, and this can show how tight the binding of the substrate is to the enzyme. Another useful constant is



Saturation curve for an enzyme reaction showing the relation between the substrate concentration (S) and rate (v).

Done

A Markov chain model

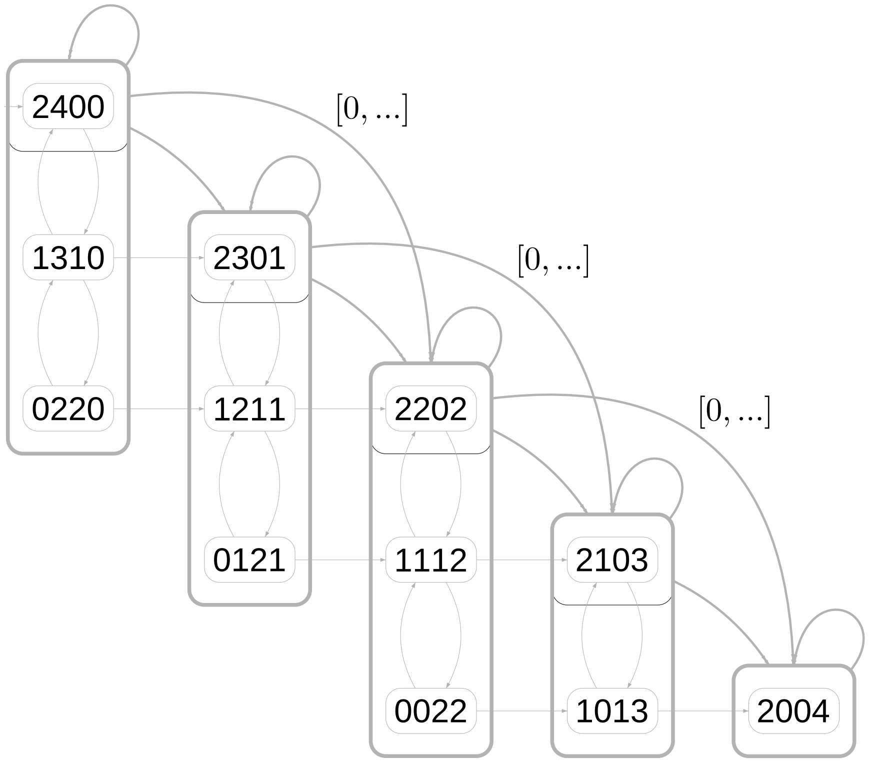


		init	*goal*
States:	enzymes	2	2
	substrate molecules	4	0
	complex molecules	0	0
	product molecules	0	4

Transitions: $E + S \underset{1}{\overset{1}{\rightleftharpoons}} C \xrightarrow{0.001} E + P$

e.g., $(x_E, x_S, x_C, x_P) \xrightarrow{0.001 \cdot x_C} (x_E + 1, x_S, x_C - 1, x_P + 1)$ for $x_C > 0$

Abstraction



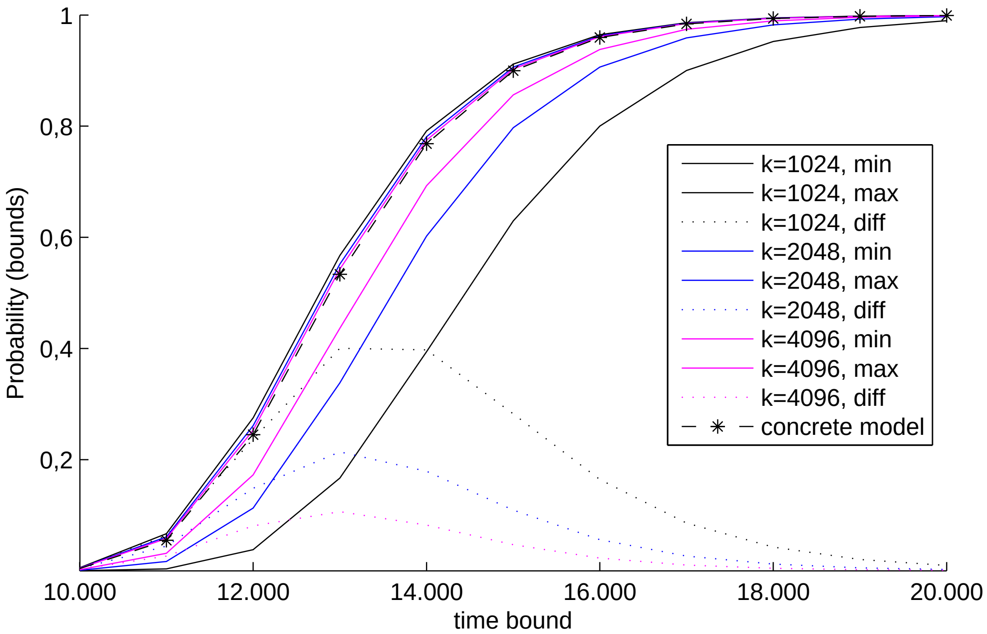
States:

		init	*goal*
	enzymes	2	2
	substrate molecules	4	0
	complex molecules	0	0
	product molecules	0	4

Transitions: $E + S \underset{1}{\overset{1}{\rightleftharpoons}} C \xrightarrow{0.001} E + P$

e.g., $(x_E, x_S, x_C, x_P) \xrightarrow{0.001 \cdot x_C} (x_E + 1, x_S, x_C - 1, x_P + 1)$ for $x_C > 0$

Verification times (200 substrates, 20 enzymes)



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

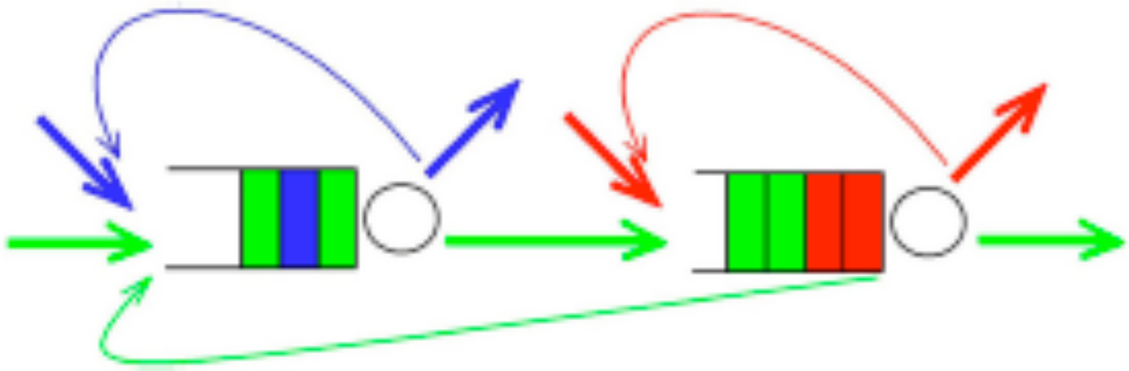
exploiting a generalized abstraction collapsing k transitions

Results

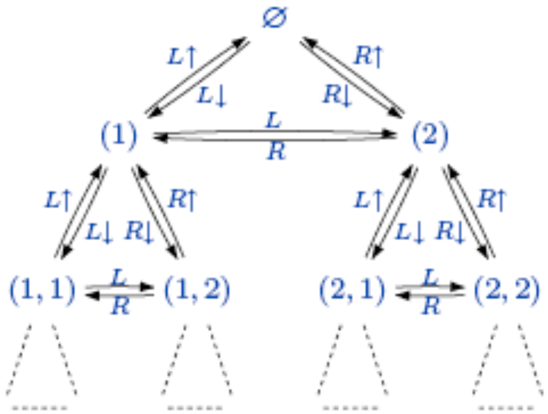
- Abstraction of enzyme-catalyzed substrate conversion
  - bisimulation does not yield any reduction
  - three-valued abstraction: reduction of state space of factor 20
  - reduction of verification time with an order of magnitude
- Difficult analysis due to stiffness of Markov chain
  - standard approach needs about $6 \cdot 10^7$ iterations
- Approximation is rather close to exact results
  - but approximation error cannot be estimated a priori

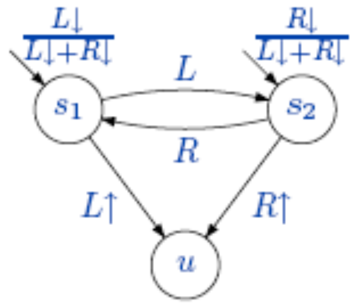
Application 2

Queuing networks: tree-based QBDs



Tree-based QBDs: M/PH$_2$/1 queue





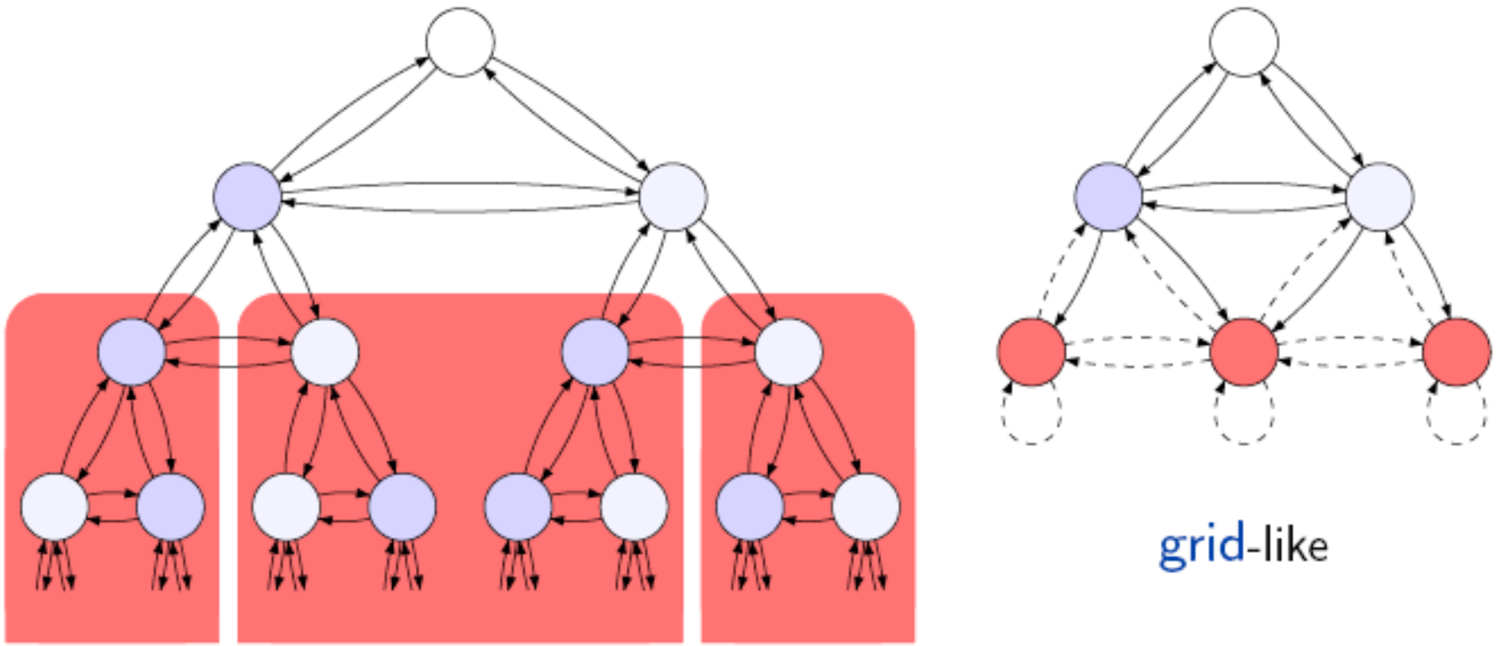
Queueing station

- preemptive LIFO scheduling
- arrival distribution: exponentially with rate $L\downarrow + R\downarrow$
- service distribution: phase-type distributed (see CTMC right)
- one service station

A grid-like abstraction

Group all states with the same number of jobs in each service phase

(in the first 2 queued jobs)



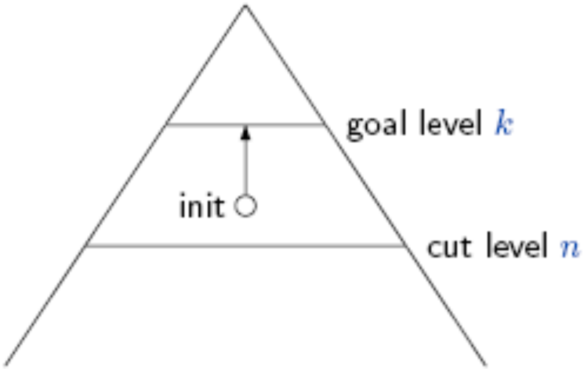
Experiments

Measure of interest

Given an initial queue, what is the probability to serve all but k jobs within t time units?

Overview

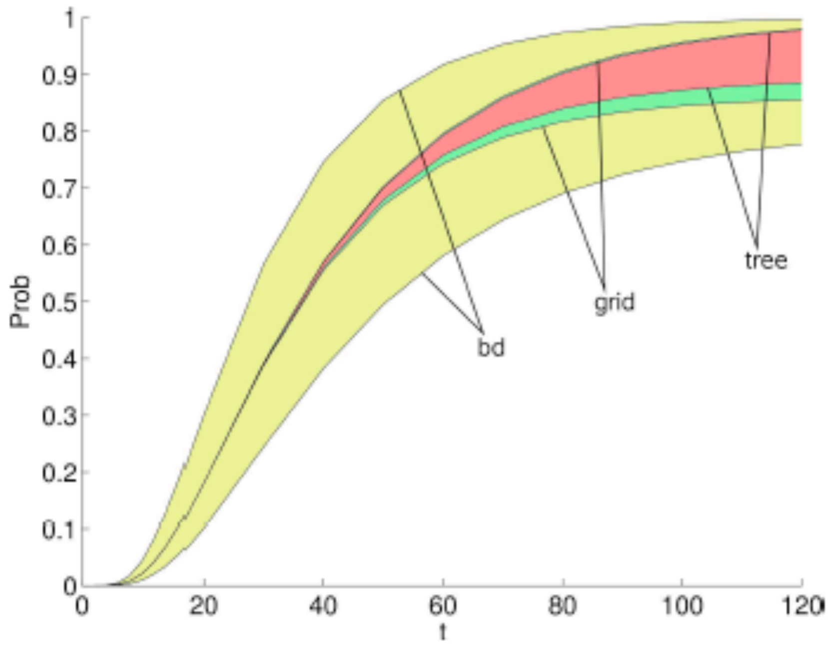
1. Precision w.r.t. partitioning schemes
2. Influence of the cut level
3. Influence of the goal level
4. Refinement for the grid scheme
5. Phase-type 5 service distribution



Comparing different partitioning schemes

Parameters: $L{\downarrow} = 2,\ R{\downarrow} = 3,\ L = 4,\ R = 5,\ L{\uparrow} = 7.5,\ R{\uparrow} = 10 \quad \Rightarrow \quad \rho_q = 0.57$

Initial state: $(1, 2, 1, 2, 1, 2, 1, 2)$ – Goal level: 0 – Cut level: 12



from now on:

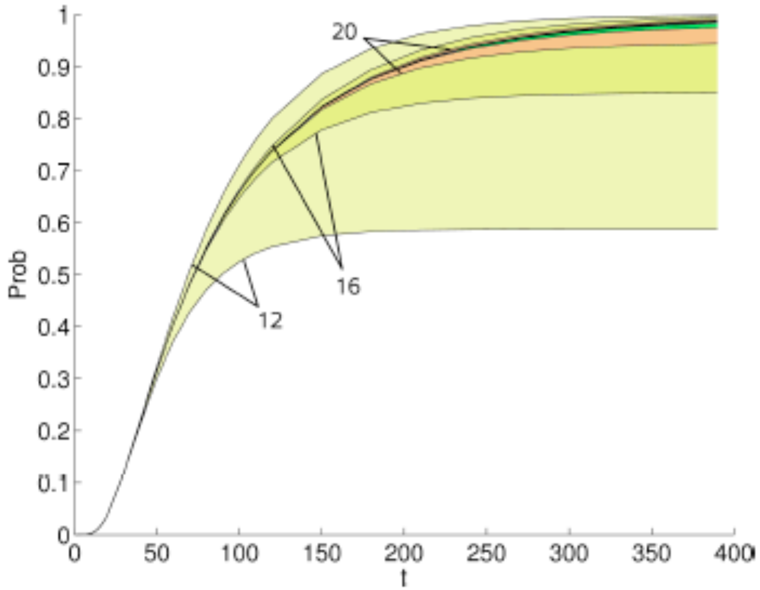
focus on **grid**-scheme

Experiments with a M/PH$_5$/1 queue

Parameters: ...with $\rho_q = 0.77$

Initial state: $(1, 2, 3, 4, 5, 1, 2, 3)$

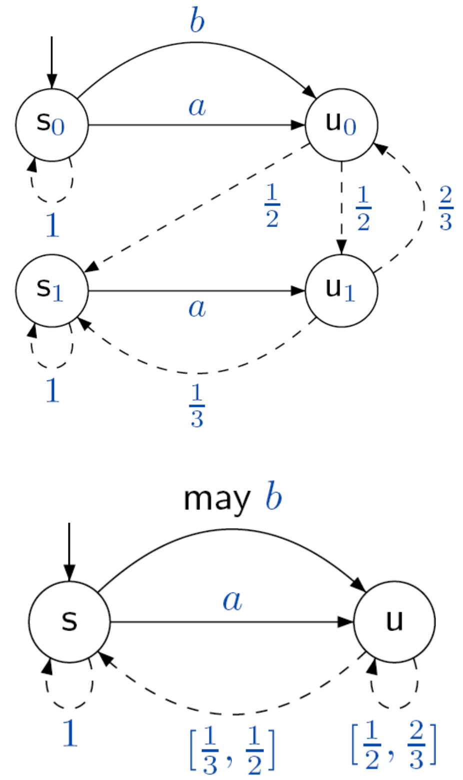
Goal level: 0



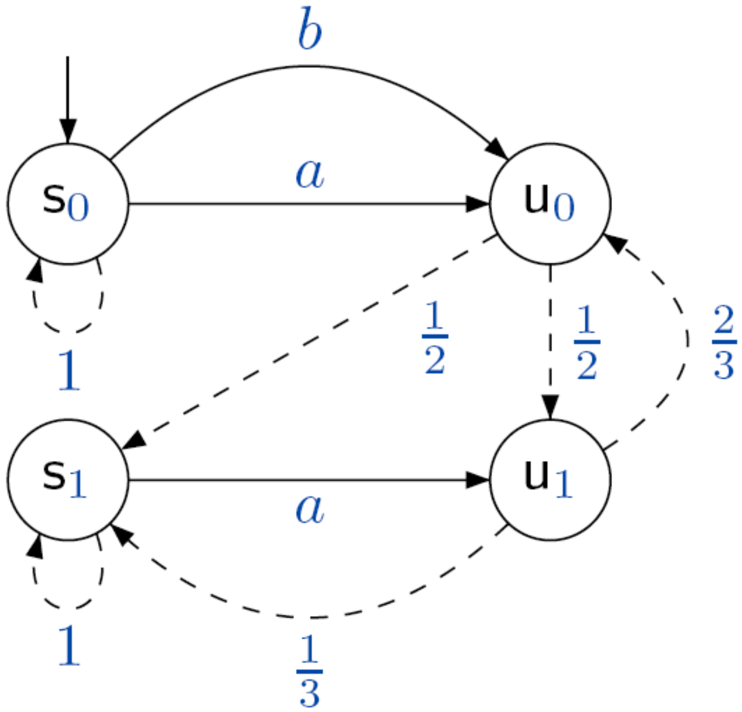
	grid abstraction								uniformization	
diff	grid 12	grid 16	grid 20	grid 24	grid 28	grid 32	grid 36	grid 40	trunc	≈ states
t 50	0.0224	0.001	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	1058	10^{740}
150	0.3117	0.0580	0.0062	0.0004	10^{-5}	10^{-6}	10^{-6}	10^{-6}	2909	10^{2033}
300	0.4054	0.1345	0.0376	0.0086	0.0015	0.0002	$2 \cdot 10^{-5}$	$3 \cdot 10^{-6}$	5607	10^{3919}
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		

Compositional abstraction

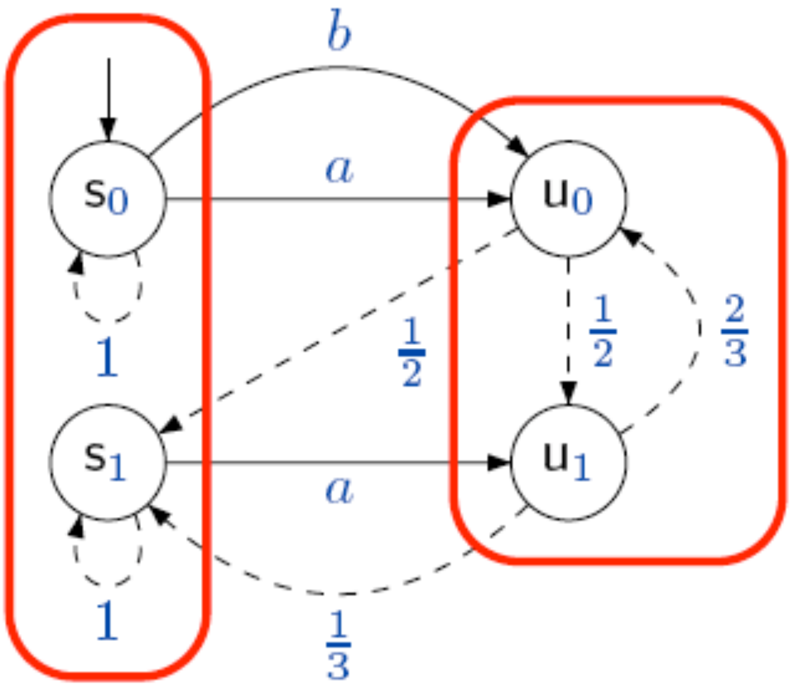
- Interactive Markov chains (IMCs)
  - mixture of labeled transition systems and CTMCs
  - allow for compositional modeling and minimisation
- Abstract IMCs = AMC + MTS
  - use interval abstraction (AMC)
  - and modal transition systems (MTS)
- Aim: abstract component-wise
  - replace $\mathcal{M}_i$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



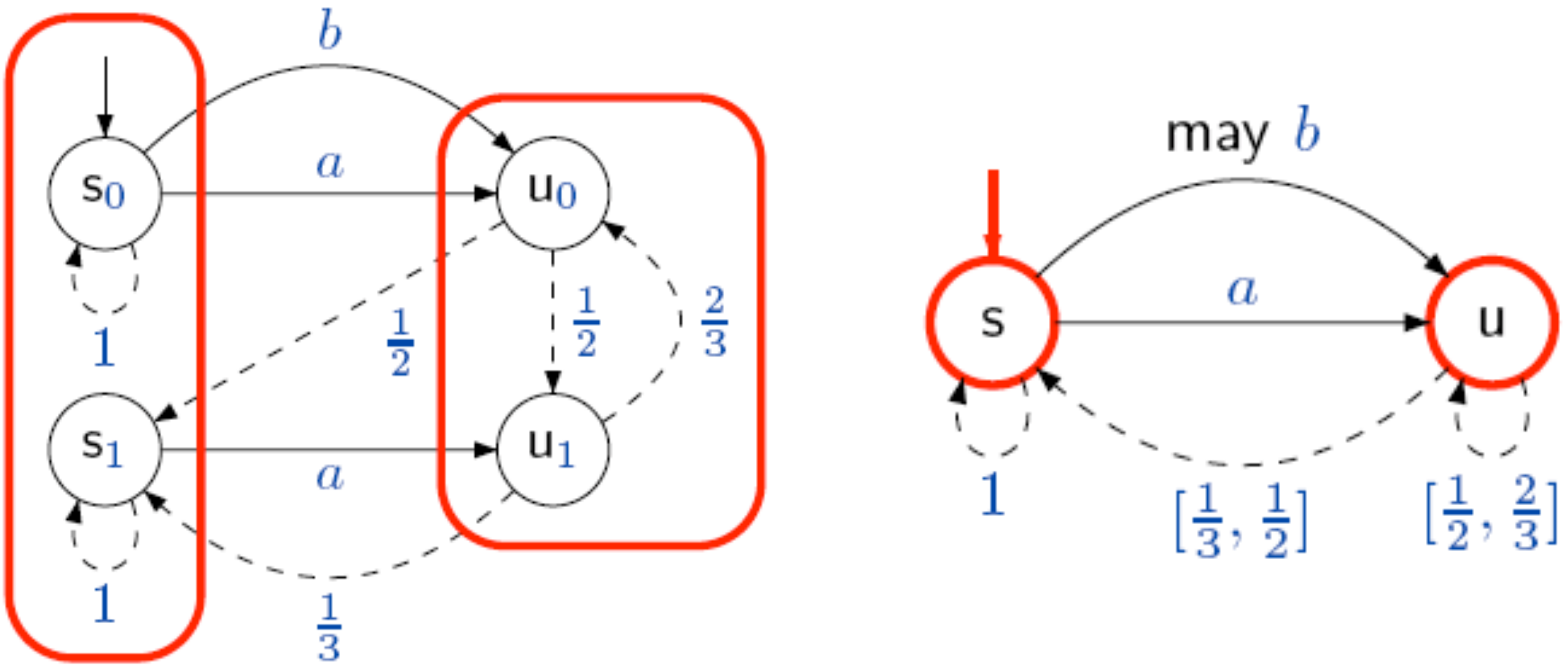
Compositional abstraction



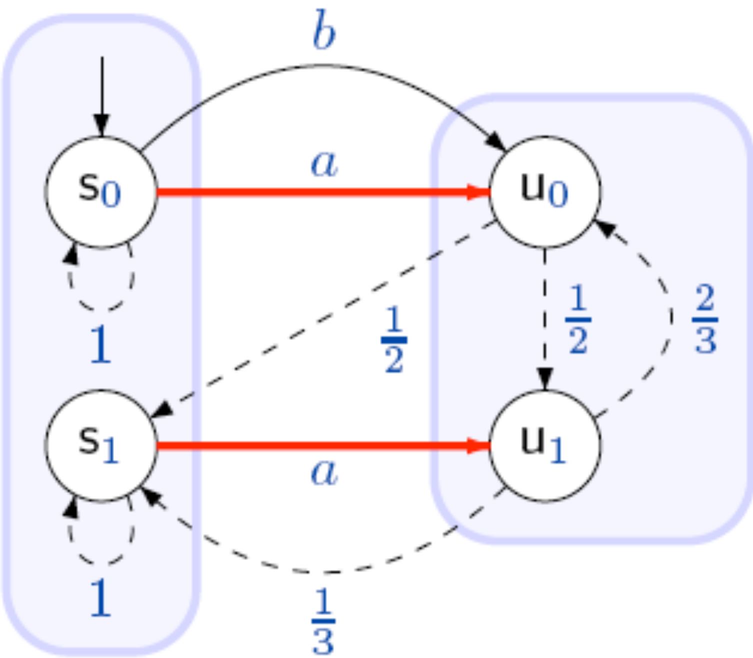
Compositional abstraction

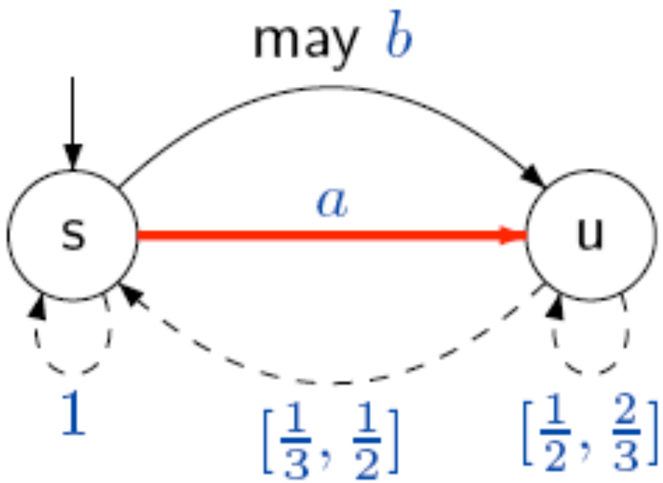


Compositional abstraction

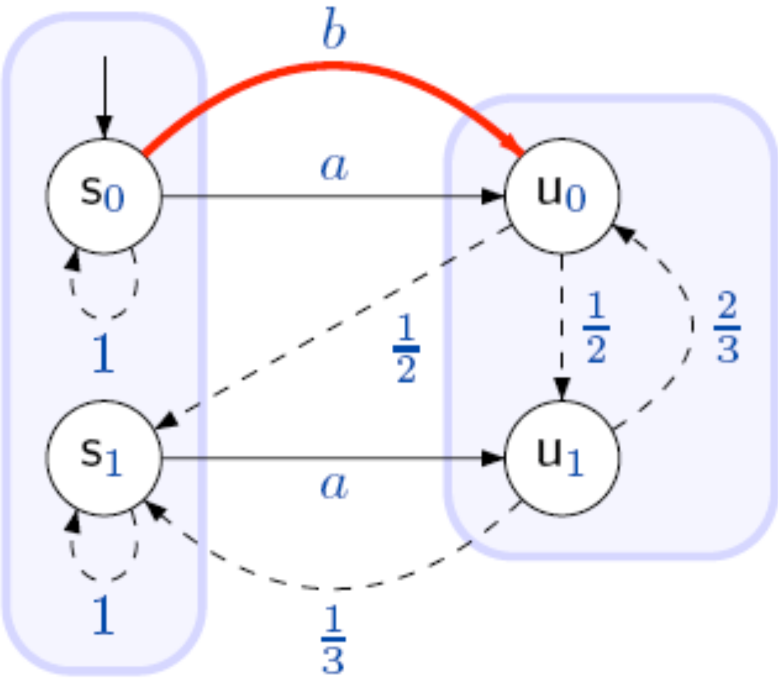


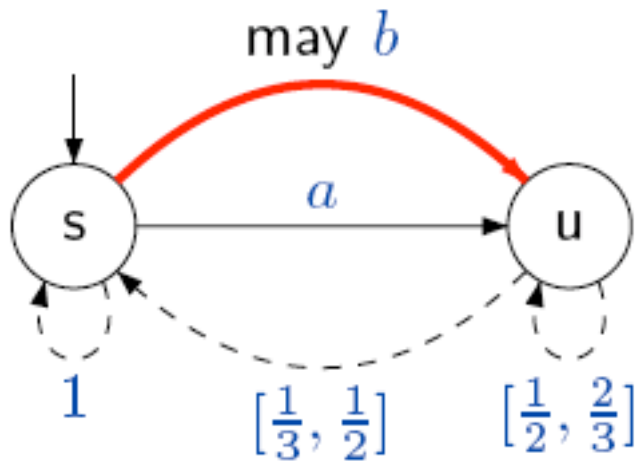
Compositional abstraction



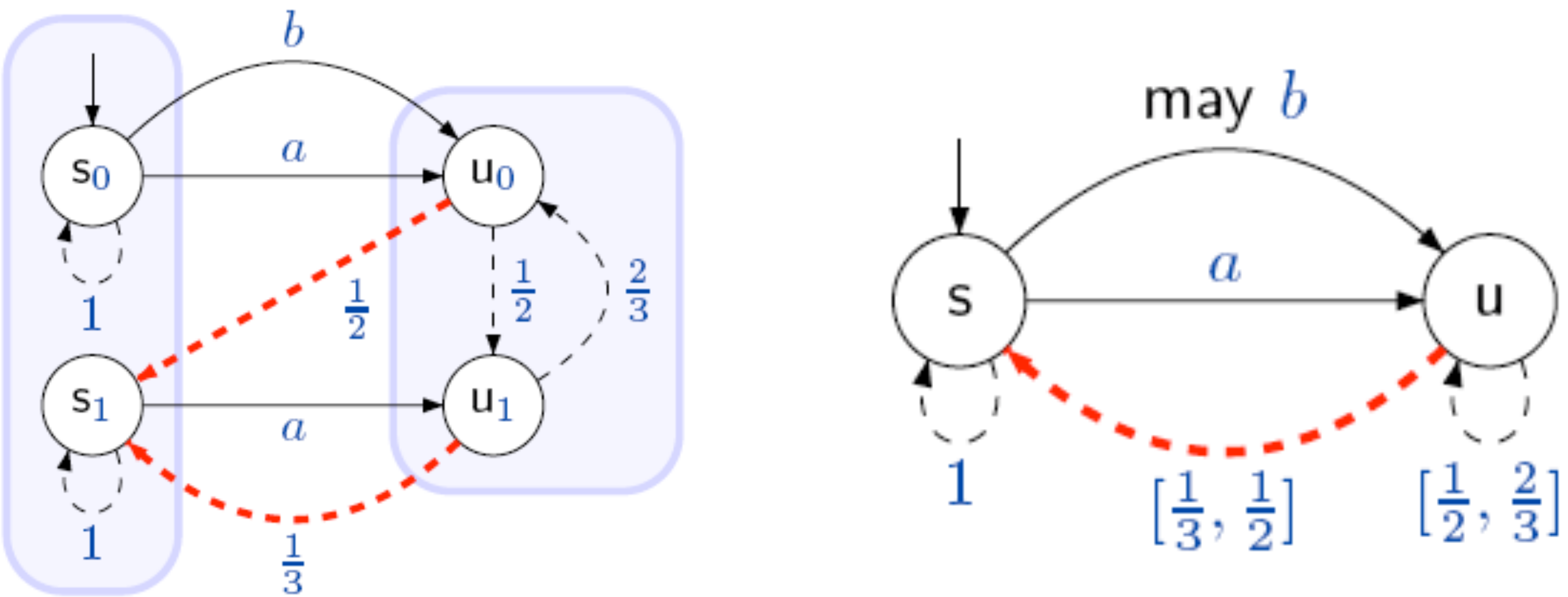


Compositional abstraction

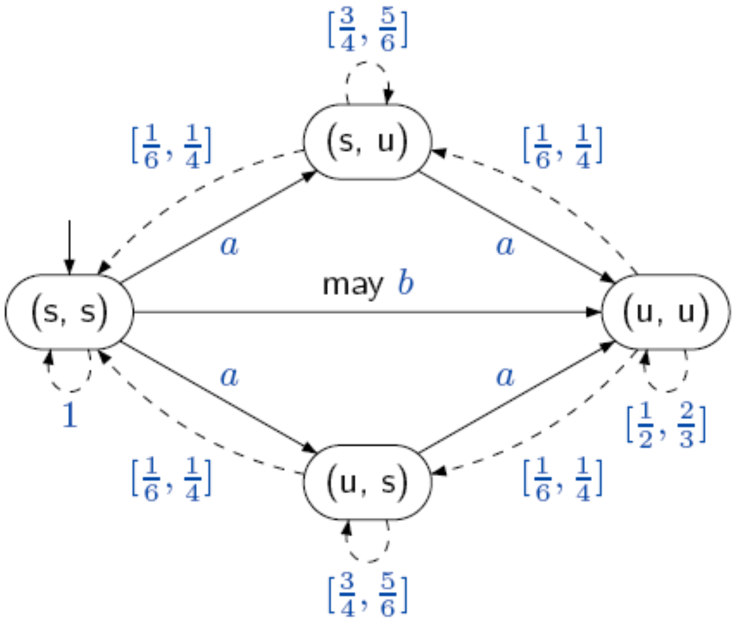




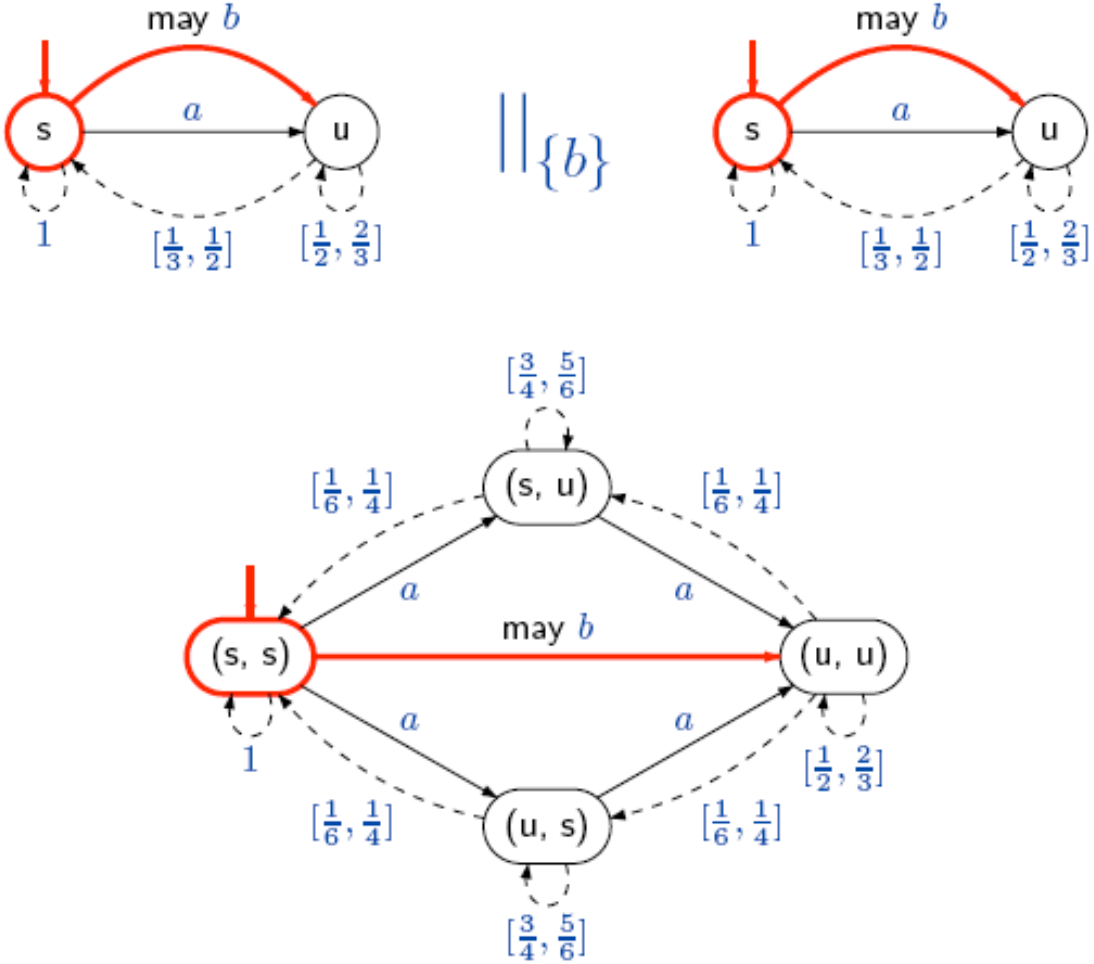
Compositional abstraction



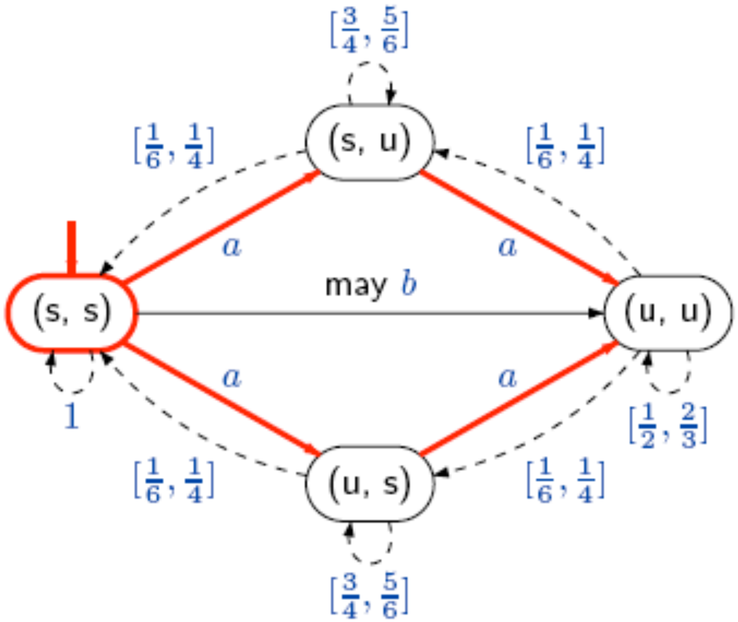
Parallel composition



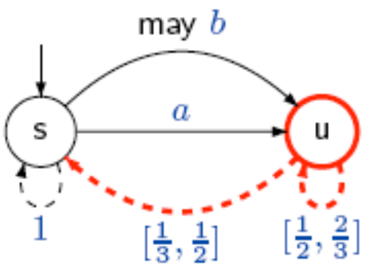
Parallel composition



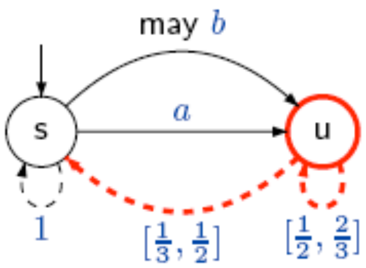
Parallel composition

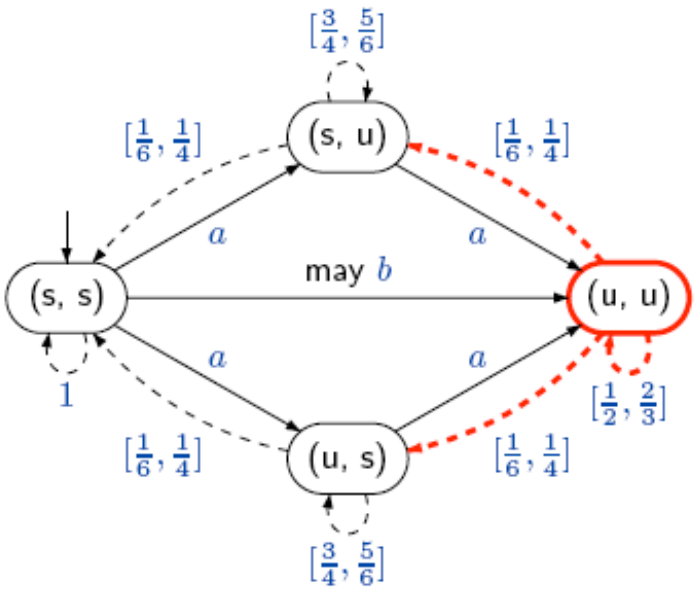


Parallel composition



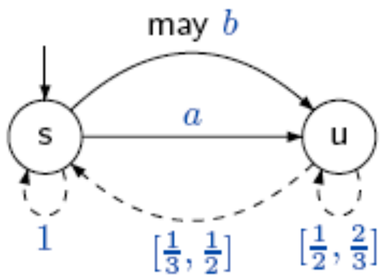
$\|_{\{b\}}$

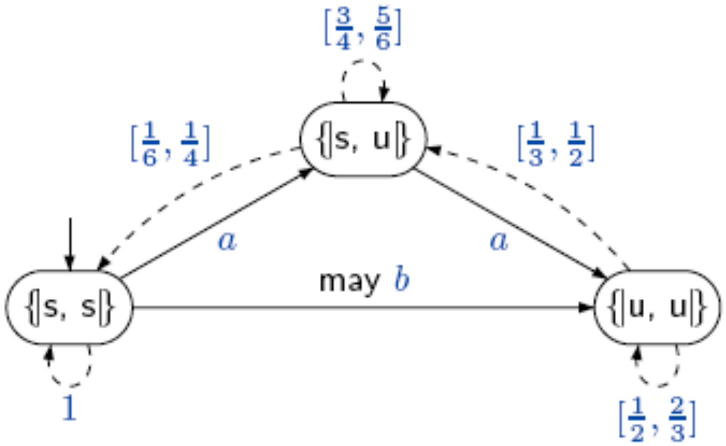




Symmetric composition

$$|||^{2}_{\{b\}}$$





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

Theoretical results

- Symmetric composition and parallel composition are bisimilar

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

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

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

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

- Abstracting many parallel "similar" components:

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

A production example

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

$$(\mathcal{M}_1 \,||_\emptyset\, \mathcal{M}_2 \,||_\emptyset\, \mathcal{M}_3) \,||_A\, \mathcal{Q} \qquad 22528 \text{ states}$$

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

$$(\mathcal{N} \,||_\emptyset\, \mathcal{N} \,||_\emptyset\, \mathcal{N}) \,||_A\, \mathcal{Q} \qquad 9504 \text{ states}$$

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

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

Some related work

- Modal transition systems (Larsen and Thomsen LICS 1988)
- MDP abstraction for reachability (Larsen *et al.* PAPM 2002)
- Game-based abstraction (Kwiatkowska *et al.* QEST 2007)
- Magnifying lens abstraction (De Alfaro & Roy CAV 2007)
- Optimal abstraction of DTMCs (Huth *et al.* QEST 2008)
- Probabilistic CEGAR (Hermanns *et al.* CAV 2008)
- Sliding-window abstraction (Henzinger *et al.* CAV 2009)

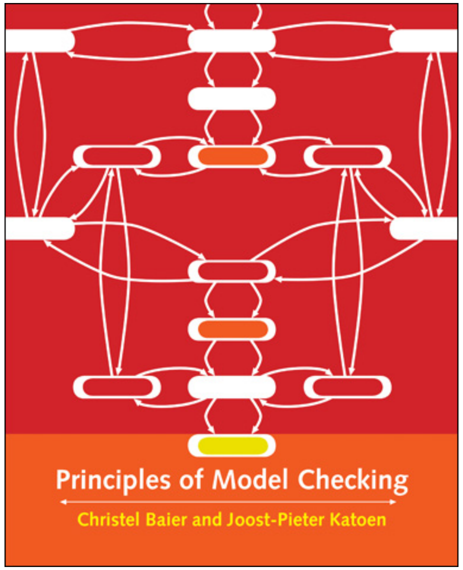
but mostly discrete-time setting only and not compositional

Abstraction

- . . . . . . has a well-founded theory
- . . . . . . allows for analyzing huge models
- . . . . . . applicable to discrete- and continuous models
- . . . . . . can be applied compositionally
- . . . . . . extendible with costs
- . . . . . . offers many interesting challenges!

more information: `moves.rwth-aachen.de/˜katoen`

Principles of Model Checking



CHRISTEL BAIER

TU Dresden, Germany

JOOST-PIETER KATOEN

RWTH Aachen University, Germany, and

University of Twente, the Netherlands

"This book offers one of the most comprehensive introductions to logic model checking techniques available today. The authors have found a way to explain both basic concepts and foundational theory thoroughly and in crystal clear prose. Highly recommended for anyone who wants to learn about this important new field, or brush up on their knowledge of the current state of the art."

(**Gerard J. Holzmann, NASA JPL, Pasadena**)

Literature

- Bisimulation minimization during model checking
  - [Katoen, Kemna, Zapreev & Jansen, TACAS 2007]
- Theory of interval abstraction
  - [Katoen, Klink, Leucker & Wolf, CAV 2007]
- Improvement by Erlang abstraction
  - [Katoen, Klink, Leucker & Wolf, CONCUR 2008]
- Abstraction of M/PH/k queues
  - [Klink, Remke, Haverkort & Katoen, QEST 2009]
- Compositional abstraction
  - [Katoen, Klink & Neuhäusser, FORMATS 2009]