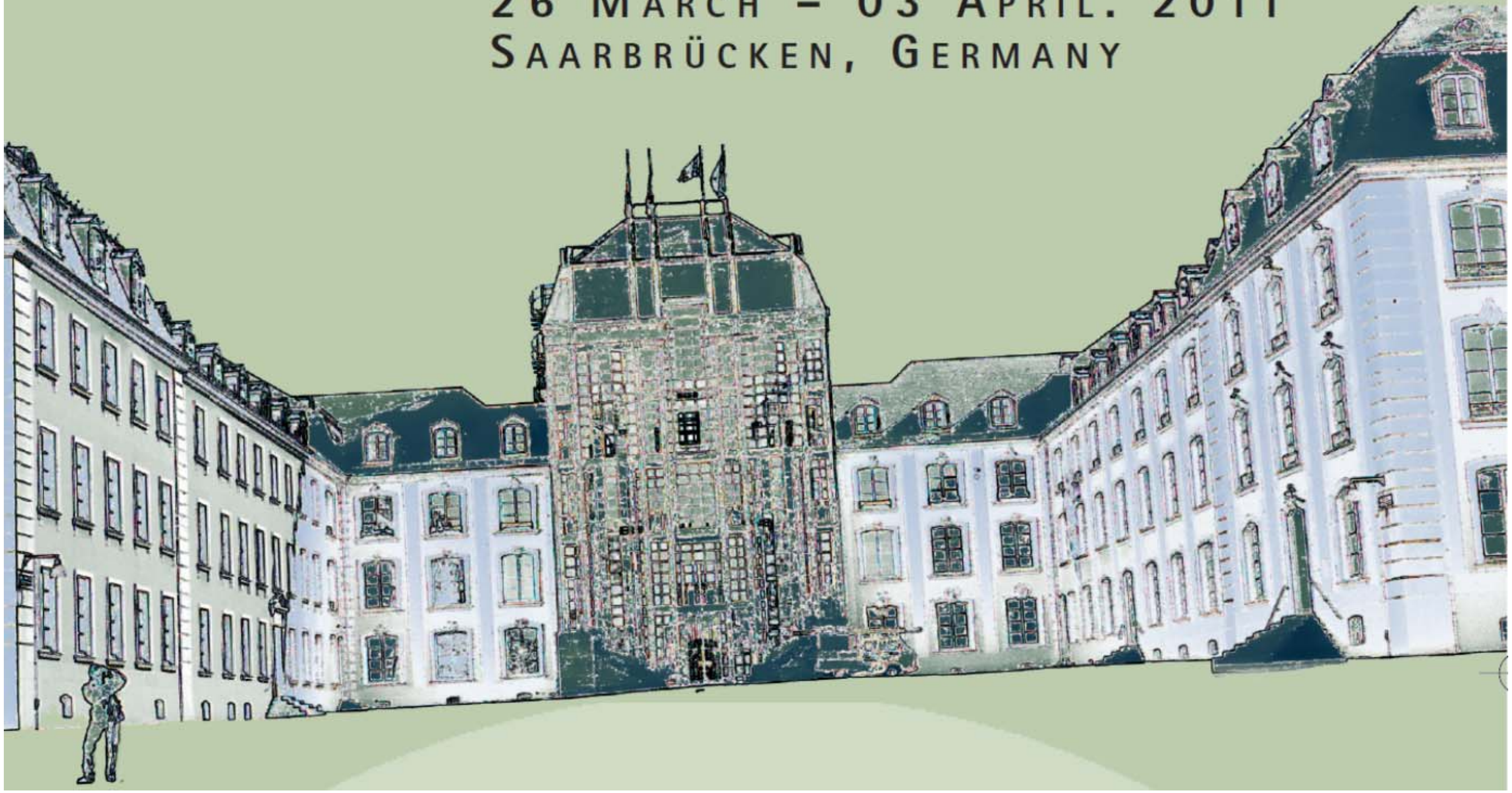
ETAPS 2011

European Joint Conferences on
Theory and Practice of Software

26 March – 03 April. 2011
Saarbrücken, Germany



What we have seen so far

Model Construction Basics: Parallel Composition

Discrete Time Markov Chains

PCTL Model Checking for DTMCs

Probabilistic Automata . . . MDPs

PCTL Model Checking for MDPs

Parametric DTMC Model Checking

Probabilistic CEGAR for MDPs

Probabilistic Timed Automata

Recall:

Complexity

- Overall complexity:

  polynomial in the size of $\mathcal{D}$,

  linear in the size of Φ,

  linear in the maximal step bound n.

Bisimulation

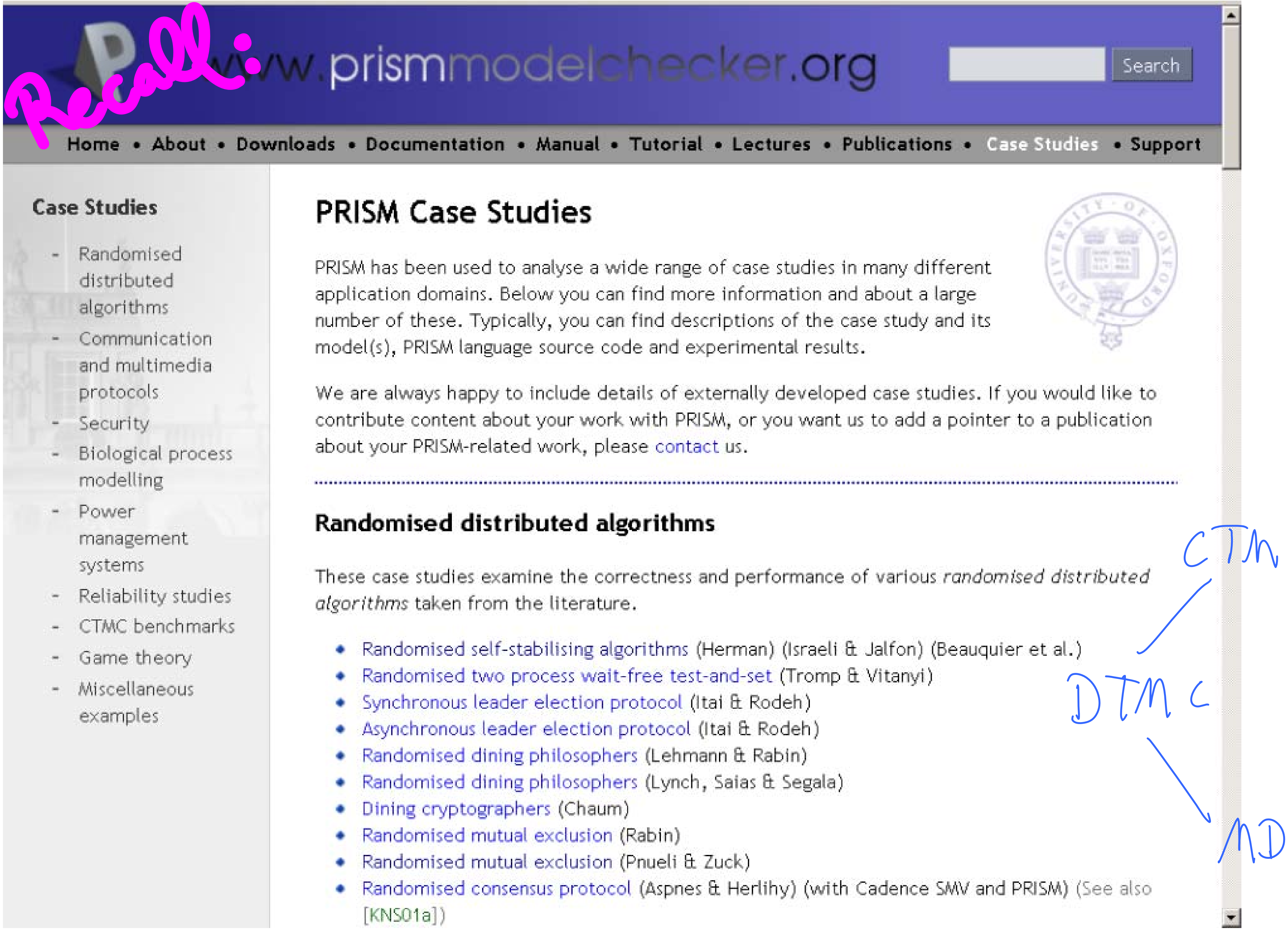
…can be lifted to probabilistic automata, such that:

State labels: PCTL-equivalence and bisimulation agree.

Transition labels: Bisimulation is a congruence for $\|_{\mathsf{Syn}}$.

Both: Bisimulation is a congruence for $\|_{\mathsf{Syn}}$ and implies PCTL-equivalence.

Recall:


www.prismmodelchecker.org
Search
Home • About • Downloads • Documentation • Manual • Tutorial • Lectures • Publications • Case Studies • Support

Case Studies

- Randomised distributed algorithms
- Communication and multimedia protocols
- Security
- Biological process modelling
- Power management systems
- Reliability studies
- CTMC benchmarks
- Game theory
- Miscellaneous examples

PRISM Case Studies

PRISM has been used to analyse a wide range of case studies in many different application domains. Below you can find more information and about a large number of these. Typically, you can find descriptions of the case study and its model(s), PRISM language source code and experimental results.

We are always happy to include details of externally developed case studies. If you would like to contribute content about your work with PRISM, or you want us to add a pointer to a publication about your PRISM-related work, please contact us.

Randomised distributed algorithms

These case studies examine the correctness and performance of various *randomised distributed algorithms* taken from the literature.

- Randomised self-stabilising algorithms (Herman) (Israeli & Jalfon) (Beauquier et al.)
- Randomised two process wait-free test-and-set (Tromp & Vitanyi)
- Synchronous leader election protocol (Itai & Rodeh)
- Asynchronous leader election protocol (Itai & Rodeh)
- Randomised dining philosophers (Lehmann & Rabin)
- Randomised dining philosophers (Lynch, Saias & Segala)
- Dining cryptographers (Chaum)
- Randomised mutual exclusion (Rabin)
- Randomised mutual exclusion (Pnueli & Zuck)
- Randomised consensus protocol (Aspnes & Herlihy) (with Cadence SMV and PRISM) (See also [KNS01a])

CTMC

DTMC

MDP

Continuous-time Markov chains

Recall: Stochastic Process in continuous time

A stochastic process $\{X(t) \mid t \in \mathbb{R}\}$ is a family of random variables. Notation: $X_t = X(t)$

Continuous-time Markov chain (CTMC)

Markov property: for all $0 = t_0 < t_1 < ... < t_n < t_{n+1}$ and $s_i \in S$:

$$\begin{aligned} P(X_{t_{n+1}} = s_{n+1} \mid X_{t_n} = s_n, X_{t_{n-1}} = s_{n-1}, \ldots, X_{t_0} = s_0) \\ = P(X_{t_{n+1}} = s_{n+1} \mid X_{t_n} = s_n) \end{aligned}$$

Time homogeneity:

$$= P(X_{t_{n+1}-t_n} = s_n \mid X_0 = s_0)$$

Notation

The transient probabilities at time t are, for $j \in S$, $t \geq 0$:

$$\pi_j(t) := P(X(t) = j)$$

By the law of total probabilities, the transient probabilities at any time t constitute a distribution over S.

The vector $\pi(t)$ can be obtained via the timed jump-probabilities:

$p_{ij}(t) \; := \; P(X_t = j \mid X_0 = i) \; = \; P(X_{t+v} = j \mid X_v = i)$ for any v.
$p_{ii}(0) = 1$ and $p_{ij}(0) = 0$ if $i \neq j$,

And, with $\mathbf{P}(t) \; = \; (p_{ij}(t))_{i,j \in S}$, we have:

$$\pi(t) \; = \; \pi(0)\,\mathbf{P}(t)$$

However, $\mathbf{P}(t)$ cannot be obtained easily!

A generator for $\mathbf{P}(t)$

Given a continuous-time Markov chain $\{X(t) \mid t \in \mathbb{R}\}$, with corresponding probability space $(\Omega, \mathcal{F}, P)$, state space $S = \{0, 1, \ldots\}$, and $i, j, s \in S$.

The infinitesimal generator matrix

For $i, j \in S$, we define the infinitesimal generator matrix $\mathbf{Q}$:

$$q_{ij} := \lim_{h \to 0} \frac{p_{ij}(h) - p_{ij}(0)}{h}$$

This leads to the Chapman-Kolmogorov equations (in matrix form):

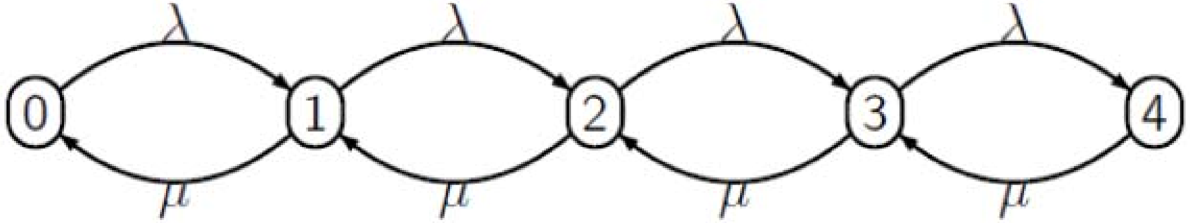
$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{P}(t)\,\mathbf{Q} = \mathbf{Q}\,\mathbf{P}(t)$$

Continuous-time Markov chains

Graph-based definition

A continuous-time Markov chain is a tuple: $(S, \mathbf{Q}, \pi(0))$ where

- S is the set of states,
- $\mathbf{Q} : S \times S \rightarrow \mathbb{R}$ is the transition rate matrix such that: $q_{ij} \geq 0$ if $i \neq j$ and $q_{ii} \leq 0$ with $-q_{ii} = \sum_{j \neq i} q_{ij}$
- $\pi(0)$ is the initial distribution



$$\mathbf{Q} = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & 0 \\ \mu & -\lambda-\mu & \lambda & 0 & 0 \\ 0 & \mu & -\lambda-\mu & \lambda & 0 \\ 0 & 0 & \mu & -\lambda-\mu & \lambda \\ 0 & 0 & 0 & \mu & -\mu \end{pmatrix}$$

Transient probabilities

For homogeneous CTMC:

$$\frac{d\pi_j(t)}{dt} = \sum_{i \in S} \pi_i(t) q_{ij} \qquad \text{or in matrix form} \quad \frac{d\pi(t)}{dt} = \pi(t)\,\mathbf{Q}$$

Symbolic solution

The transient probability vector is:

$$\pi(t) = \pi(0)\, e^{\mathbf{Q}t} \qquad \text{where} \quad e^{\mathbf{Q}t} = \sum_{i=0}^{\infty} \frac{(\mathbf{Q}\, t)^i}{i!}$$

Tiny Problem: $(\mathbf{Q}t)^i$ is unstable to compute,

thus the infinite sum is not easily truncated.

Uniformisation

is the numerical method of choice. (see lecture notes.)

SIAM REVIEW
Vol. 45, No. 1, pp. 3–000

Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later*

Cleve Moler[†]
Charles Van Loan[‡]

Abstract. In principle, the exponential of a matrix could be computed in many ways. Methods involving approximation theory, differential equations, the matrix eigenvalues, and the matrix characteristic polynomial have been proposed. In practice, consideration of computational stability and efficiency indicates that some of the methods are preferable to others, but that none are completely satisfactory.

Most of this paper was originally published in 1978. An update, with a separate bibliography, describes a few recent developments.

Steady state probabilities

For a finite CTMC, the limit:

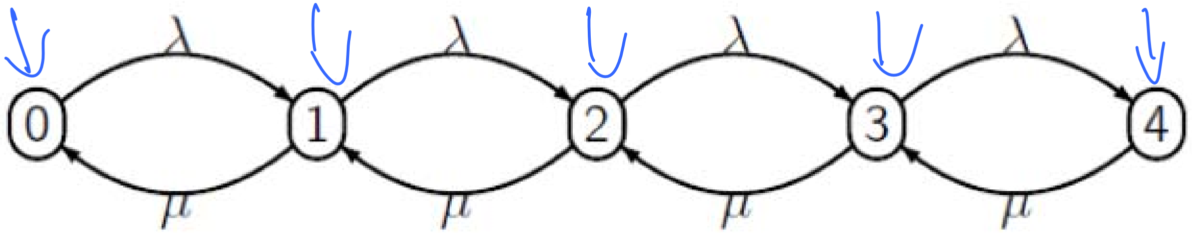
$$\tilde{\pi} \; = \; \lim_{t \to \infty} \pi(t)$$

always exists. It satisfies

$$\lim_{t \to \infty} \frac{d\pi_j(t)}{dt} \; = \; 0$$

which implies

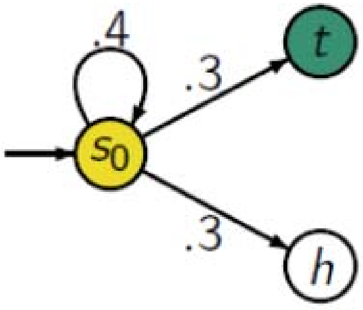
$$\tilde{\pi}\,\mathbf{Q} \; = \; 0$$

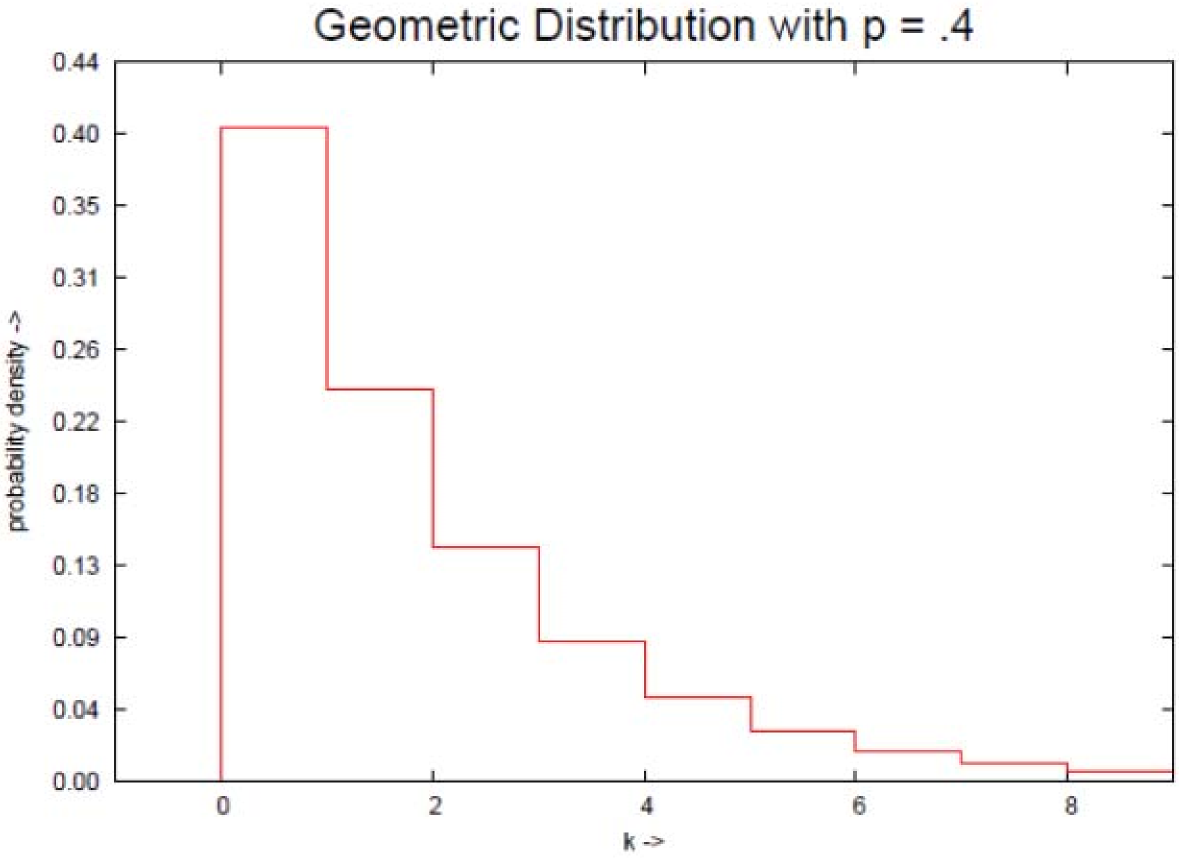


Recall:

DTMC: What to remember

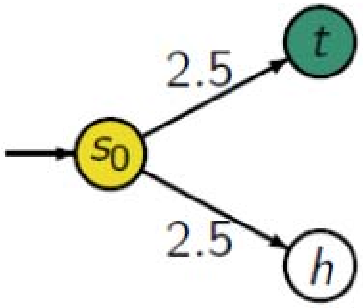
- Finite – homogeneous – discrete-time Markov chains.
- Transient behaviour: $\pi(n) \quad = \quad \pi(0)\ \mathbf{P}^n$.
- Stationary behaviour: $\tilde{\pi} \quad = \quad \tilde{\pi}\ \mathbf{P}$.
- Sojourn time is geometrically distributed: $P(SJ = k) \quad = \quad p\,(1-p)^k$.

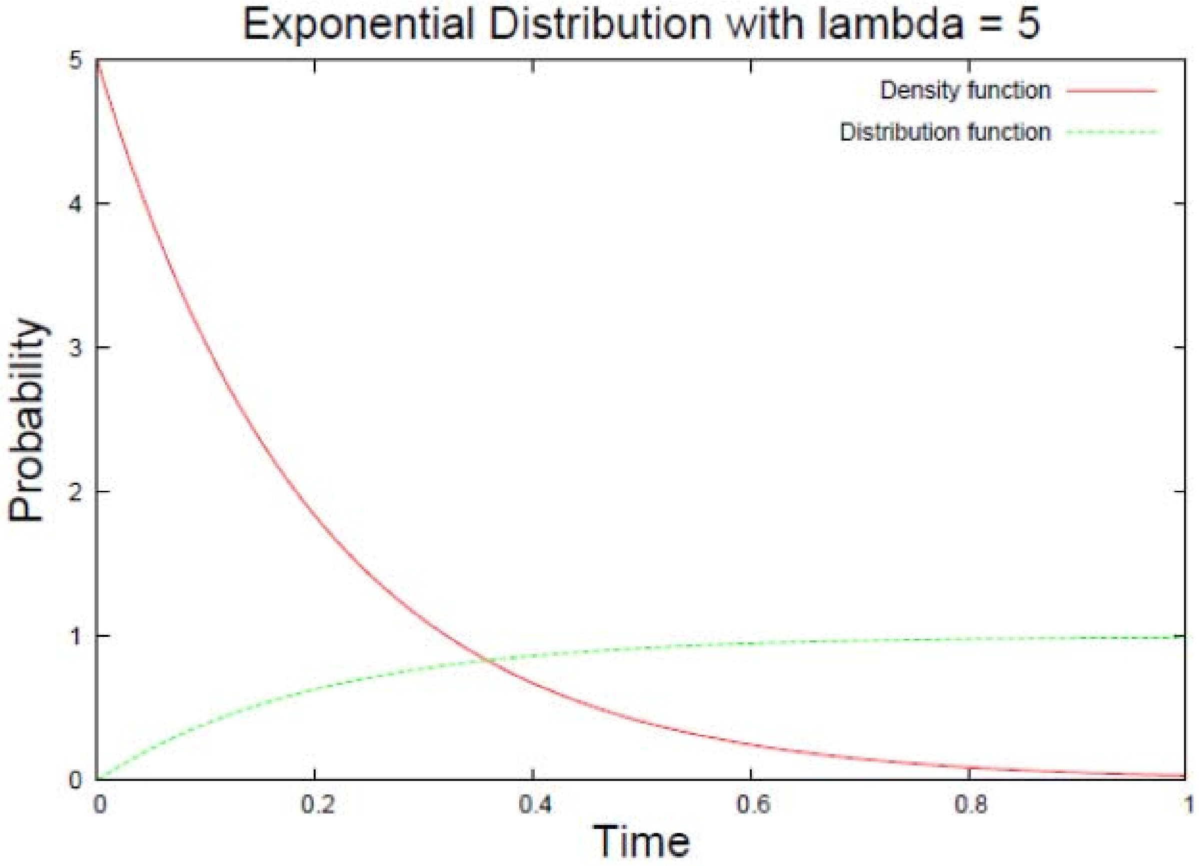




CTMC: What to remember

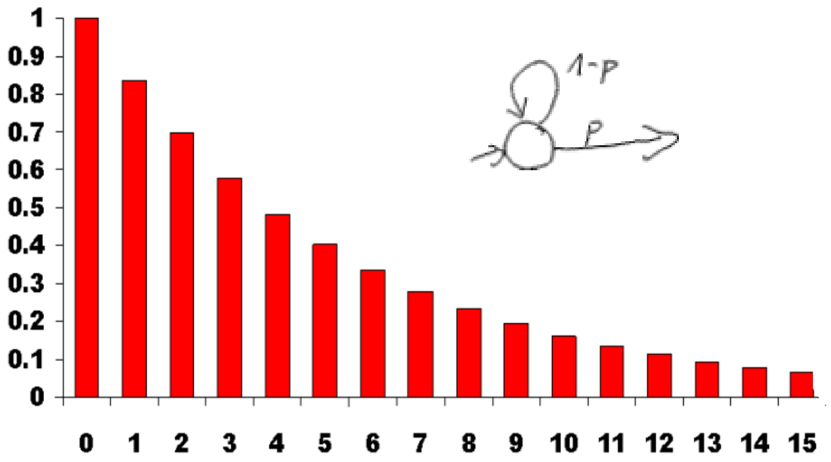
- Finite – homogeneous – continuous-time Markov chains.
- Transient behaviour: $\pi(t) \;=\; \pi(0)\, e^{\mathbf{Q}t}$.
- Stationary behaviour: $\tilde{\pi}\,\mathbf{Q} \;=\; 0$.
- Sojourn time is exponentially distributed: $P(SJ \leq t) \;=\; 1 - e^{-\lambda t}$.

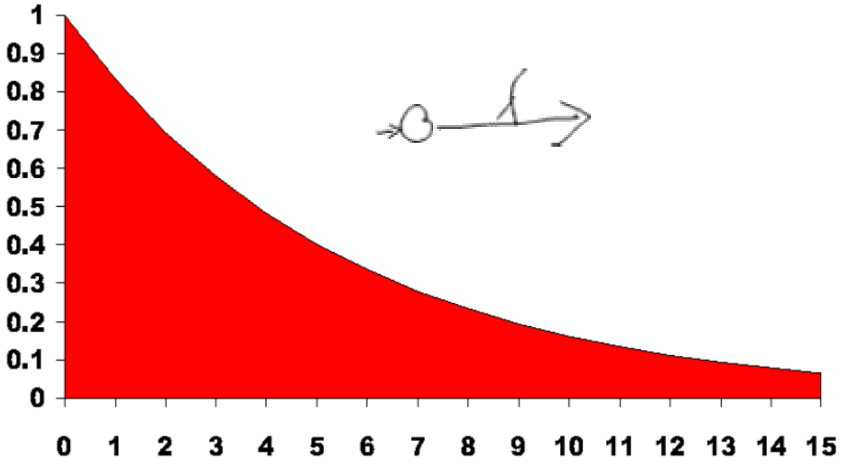




CTMC: What to remember

- Finite – homogeneous – continuous-time Markov chains.
- Transient behaviour: $\pi(t) \;=\; \pi(0)\, e^{\mathbf{Q}t}$.
- Stationary behaviour: $\tilde{\pi}\,\mathbf{Q} \;=\; 0$.
- Sojourn time is exponentially distributed: $P(SJ \leq t) \;=\; 1 - e^{-\lambda t}$.





What comes next

We are going to discuss

Model Construction, and

Model Checking

for CTMCs

CTMC with labels

We equip states of CTMC with labels to identify state properties:

- AP denotes the set of atomic propositions
- A CTMC is a tuple $\mathcal{C} = (S, \mathbf{R}, \pi(0), AP, L)$ where $L : S \rightarrow AP$.

  $L(s)$ specifies properties hold in state s.

Continuous Stochastic Logic

Syntax

State formulas:

$$\Phi := \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_J(\phi) \mid \mathbb{L}_J(\Phi)$$

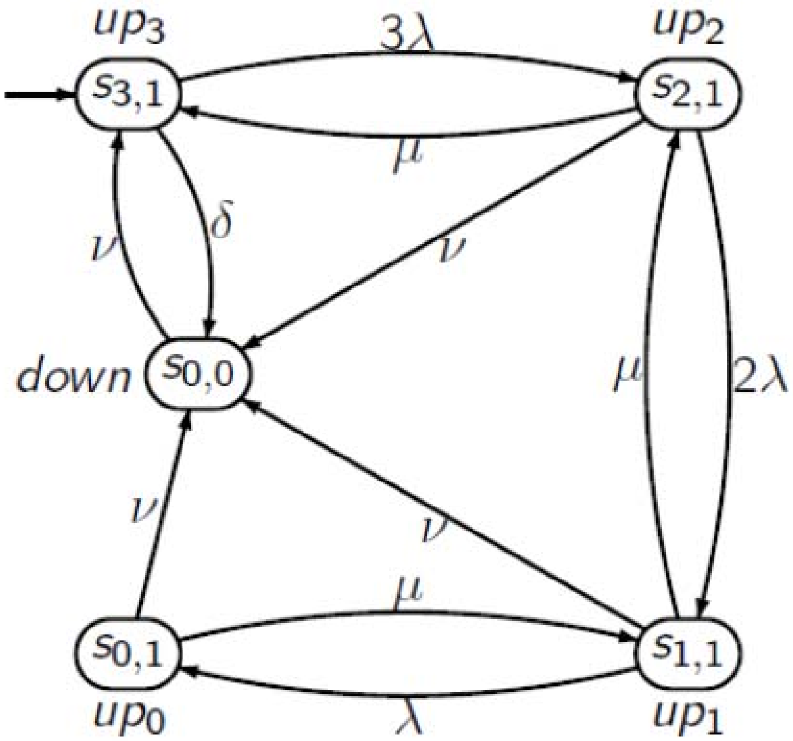
where $a \in AP$, $J \subseteq [0,1]$ is an interval with rational bounds.
Path formulas:

$$\phi := \mathcal{X}^I \Phi \mid \Phi_1 \mathcal{U}^I \Phi_2$$

where $n \in \mathbb{N}$, and $I \subseteq \mathbb{R}^{\geq 0}$ denotes an interval.

Specifying properties using CSL

We consider a triple modular redundant system:



Some CSL properties

- In the steady state, the probability that the system is in state $s_{2,1}$ is at most p,

?

- The transient probability at time t in state s' meets the bound $> p$,

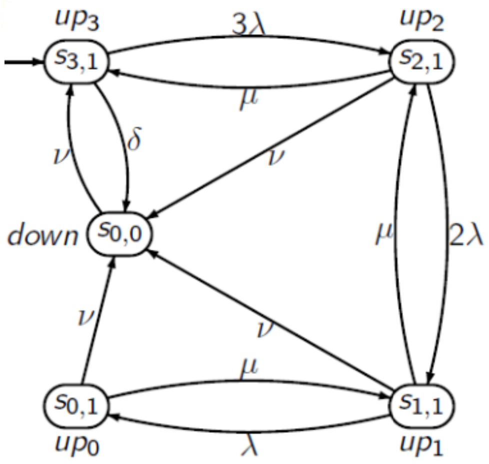
?

- The probability of reaching *down* state within 10 time units after having continuously operated with at least two processors is at most 0.01

?

- In steady state, with probability at least 0.9, the probability that the system will not go down within 10 time units is at least 0.8

$$\mathbb{L}_{\geqslant 0.9}\left(\mathbb{P}_{\geqslant 0.8}\left(\Diamond^{\leqslant 10} \, down\right)\right)$$



Availability properties using CSL

- $\mathbb{L}_J(up)$: steady-state availability
- $\mathbb{P}_J(\Diamond^{[t,t]}up)$: instantaneous availability at time t
- $\mathbb{P}_J(\Phi\mathcal{U}^{[t,t]}up)$: conditional instantaneous availability at time t
- $\mathbb{P}_J(\Box^{[t,t']}up)$: interval availability,
- $\mathbb{L}_J(\mathbb{P}_J(\Box^{[t,t']}))$: steady-state interval availability
- $\mathbb{P}_J(\Phi\mathcal{U}^{[t,t']}\mathbb{L}_J(up))$: conditional time-bounded steady-state availability

Semantics

Satisfaction relation for CSL state formulas

Given a CTMC $\mathcal{C} = (S, \mathbf{R}, \pi(0), AP, L)$, state $s \in S$, the satisfaction relation $\models$ is defined by:

- $s \models a$ iff $a \in L(s)$,
- $s \models \neg\Phi$ iff $s \not\models \Phi$,
- $s \models \Phi \wedge \Psi$ iff $s \models \Phi$ and $s \models \Psi$,
- $s \models \mathbb{P}_J(\phi)$ iff $\mathbf{Pr}_s(\phi) \in J$,
- $s \models \mathbb{L}_J(\Phi)$ iff $\tilde{\pi}_s(Sat(\Phi)) \in J$.

Satisfaction relation for CSL path formulas

Given a CTMC $\mathcal{C} = (S, \mathbf{R}, \pi(0), AP, L)$, path σ, the satisfaction relation $\models$ is defined by:

time spent in state s_0

- $\sigma \models \mathcal{X}^I \Phi$ iff $\underline{\sigma[1]} \models \Phi$ and $\delta(\sigma, 0) \in I$,

	Φ								

state occupied at time t

- $\sigma \models \Phi \mathcal{U}^I \Psi$ iff there exists $t \in I$ with $\sigma@t \models \Psi$ and for all $t' < t$, $\sigma@t' \models \Phi$

Φ	Φ	·	·	·	Φ	Ψ			

$\uparrow$ t

Measurability of CSL events

Let $\mathcal{C} = (S, \mathbf{R}, \pi(0), AP, L)$ be a CTMC and $s \in S$, then, for each CSL path formula ϕ, the set $Paths(\phi) := \{\sigma \in Paths(s) \mid \sigma \models \phi\}$ is measurable.

Model checking steady state operator

Steady state operator $\mathbb{L}_J(\Phi)$: very similar to the DTMC case

- Assume Φ is computed recursively.
- Determine the set $\mathbf{B}$ of bottom strongly connected components, BSCCs.
- Compute the probability of reaching each BSCC B.
- For each B compute $\tilde{\pi}$ restricted to B.
- Finally, compute $\tilde{\pi}_s(\Phi)$ as follows:

$$\tilde{\pi}_s(\Phi) = \sum_{B \in \mathbf{B}} \left(\mathbf{Pr}_s(\Diamond B) \sum_{s' \in B, s' \models \Phi} \tilde{\pi}^B(s') \right)$$

using that $\tilde{\pi}^B$ is the unique solution of $\pi^B \mathbf{Q}^B = 0$.

The computation of $\mathbf{Pr}_s(\phi)$

The case $\phi = \mathcal{X}^I\Phi$

The set $Sat(\Phi)$ is computed recursively. Let $I = [a, b]$, it holds:

$$\begin{aligned}\mathbf{Pr}_s(\mathcal{X}^I\Phi) &= \mathbf{Pr}_s(a < SJ_s < b) \sum_{s' \in Sat(\Phi)} \mathbf{P}(s, s') \\ &= (e^{-E(s)a} - e^{-E(s)b}) \sum_{s' \in Sat(\Phi)} \mathbf{P}(s, s')\end{aligned}$$

For $I = [0, \infty)$, it simplifies to:

$$\mathbf{Pr}_s(\mathcal{X}\Phi) = \sum_{s' \in Sat(\Phi)} \mathbf{P}(s, s')$$

The computation of $\mathbf{Pr}_s(\phi)$

The case $\phi = \Phi \, \mathcal{U}^I \, \Psi$

The vector $\mathbf{Pr}_s(\Diamond(B)) : S \times \mathbf{I} \rightarrow [0,1]$ is the least fixed point of the higher-order operator $\nabla : (S \times \mathbf{I} \rightarrow [0,1]) \rightarrow (S \times \mathbf{I} \rightarrow [0,1])$ which is defined as follows: $\nabla(F)(s, I)$ equals

$$\begin{cases} \displaystyle\int_0^b E(s)e^{-E(s)t} \cdot \sum_{s' \in S} \mathbf{P}(s,s') \cdot F(s', I \ominus t)\, dt & \text{if } s \notin B \\ \displaystyle e^{-E(s)a} + \int_0^a E(s)e^{-E(s)t} \cdot \sum_{s' \in S} \mathbf{P}(s,s') \cdot F(s', I \ominus t)\, dt & \text{if } s \in B. \end{cases}$$

where $I = [a, b]$.

Luckily, we do not have to solve these integral equations.
Instead, there is a neat way of reducing everything to the computation of $\pi(a)$
and of $\pi(b - a)$
– the latter in a CTMC that is initialised with $\pi(a)$

Complexity

- Solving the equation systems is in polynomial time.
- Matrix vector multiplication is also in polynomial time.
- The computation will be repeated for each state sub-formula.
- Uniformisation requires matrix vector multiplications in the order of $q\,t$ where q is the maximal exit rate appearing.
- Overall complexity:

  polynomial in the size of $\mathcal{M}$,

  linear in the size of Φ,

  linear in qt.

Recall:

What comes next

We are going to discuss

Model Construction, and

Model Checking

for CTMCs

Recall:

Interleaving operator for transition systems

$\mathcal{T}_1 = (S_1, \mathrm{Act}_1, \longrightarrow_1, s_{01}, AP_1, L_1)$ $\qquad$ $\mathcal{T}_2 = (S_2, \mathrm{Act}_2, \longrightarrow_2, s_{02}, AP_2, L_2)$

The composite transition system $\mathcal{T}_1 \,|||\, \mathcal{T}_2$ is:

$$\mathcal{T}_1 \,|||\, \mathcal{T}_2 \;=\; (S_1 \times S_2,\, \mathrm{Act}_1 \cup \mathrm{Act}_2,\, \longrightarrow,\, \langle s_{01}, s_{02}\rangle, AP, L)$$

where the transition relation $\longrightarrow$ is given by:

$$\frac{s_1 \xrightarrow{\alpha}_1 s_1'}{\langle s_1, s_2\rangle \xrightarrow{\alpha} \langle s_1', s_2\rangle} \qquad \frac{s_2 \xrightarrow{\alpha}_2 s_2'}{\langle s_1, s_2\rangle \xrightarrow{\alpha} \langle s_1, s_2'\rangle}$$

atomic propositions: $AP = AP_1 \uplus AP_2$

labeling function: $L(\langle s_1, s_2\rangle) = L_1(s_1) \cup L_2(s_2)$

Interleaving operator for CTMC

$\mathcal{M}_1 = (S_1, \mathbf{Q}_1, \ldots)$ $\mathcal{M}_2 = (S_2, \mathbf{Q}_2, \ldots)$

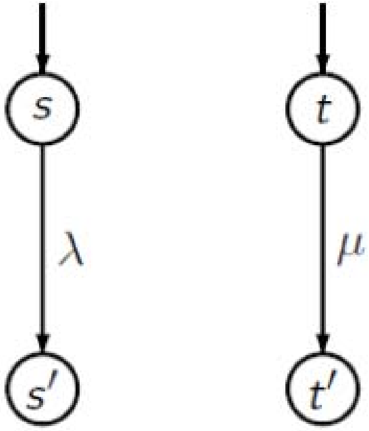
The composite CTMC $\mathcal{M}_1 \,|||\, \mathcal{M}_2$ is:

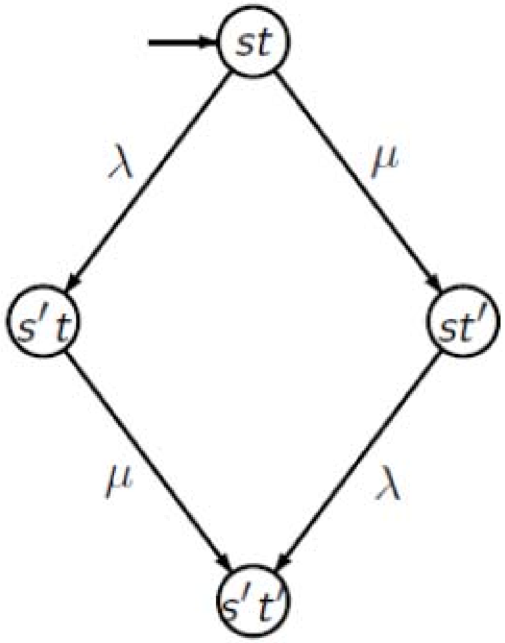
$$\mathcal{M}_1 \,|||\, \mathcal{M}_2 \;=\; (S_1 \times S_2,\, \mathbf{R},\, \ldots),$$

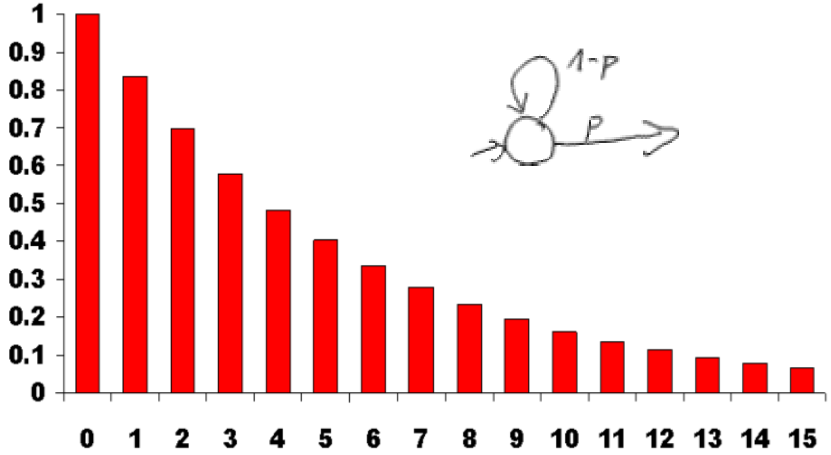
where the generator matrix $\mathbf{Q}$ is given by:

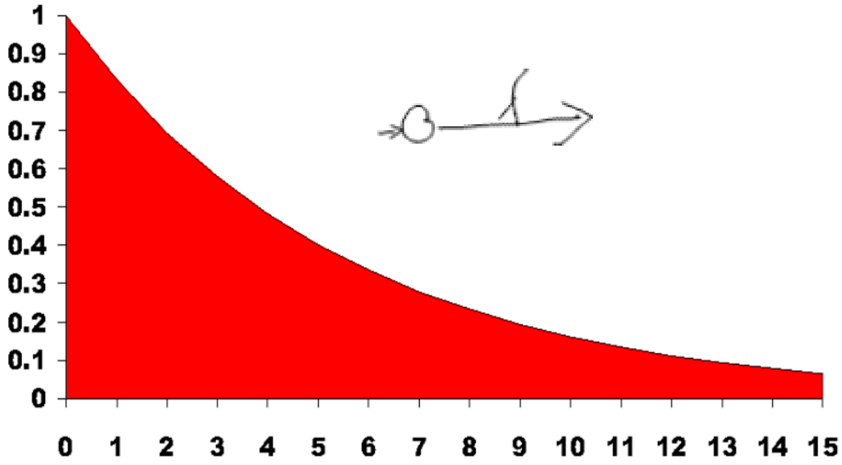
$$\frac{s_1 \xrightarrow{\lambda}_1 s_1'}{\langle s_1, s_2 \rangle \xrightarrow{\lambda} \langle s_1', s_2 \rangle} \qquad \frac{s_2 \xrightarrow{\lambda}_2 s_2'}{\langle s_1, s_2 \rangle \xrightarrow{\lambda} \langle s_1, s_2' \rangle}$$

Interleaving for CTMCs









Recall:

Synchronous product

$\mathcal{D}_1 = (S_1, \mathsf{Act}_1, \mathbf{P}_1, \ldots)$ $\mathcal{D}_2 = (S_2, \mathsf{Act}_2, \mathbf{P}_2, \ldots)$

The synchronous product $\mathcal{D}_1 \otimes \mathcal{D}_2$ is:

$$\mathcal{D}_1 \otimes \mathcal{D}_2 \;=\; (S_1 \times S_2, \mathsf{Act}, \longrightarrow, \ldots)$$

where the *probability matrix* $\mathbf{P}$: is given by:

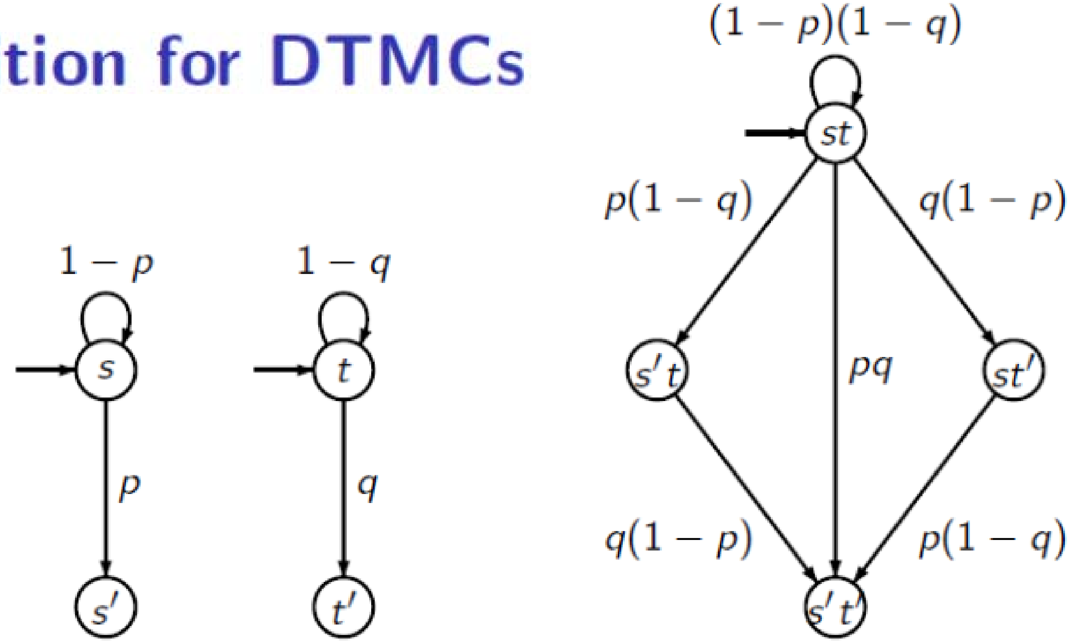
$$\frac{\mathbf{P}_1(s_1, \alpha, s_1') = p > 0 \;\wedge\; \mathbf{P}_2(s_2, \beta, s_2') = q > 0}{\mathbf{P}(\langle s_1, s_2\rangle, \alpha * \beta, \langle s_1', s_2'\rangle) \;=\; pq}$$

for parallel DTMCs with fully synchronous execution

Connecting DTMC and CTMC Composition

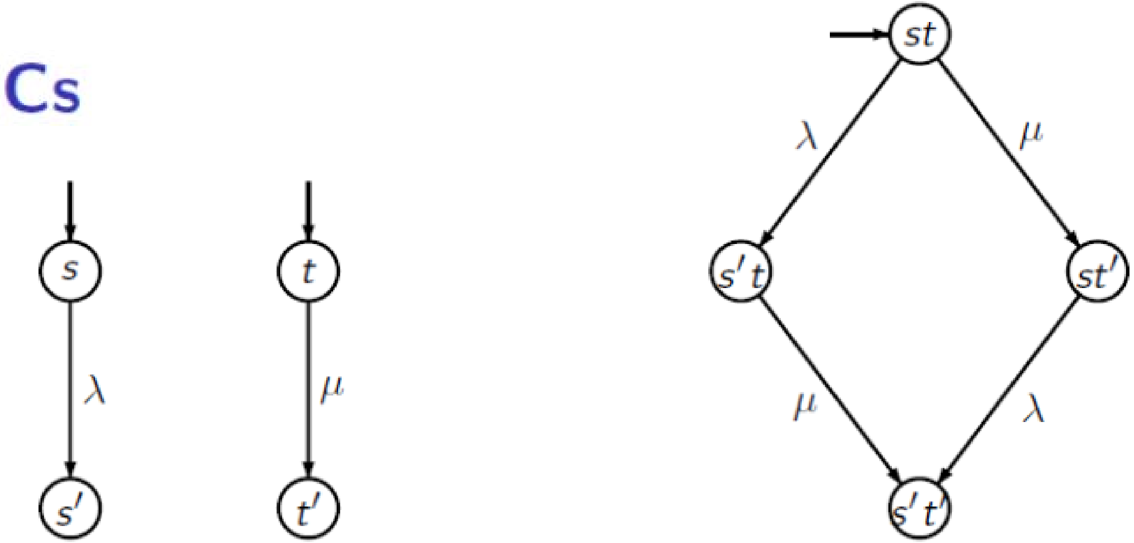
Synchronous Composition for DTMCs

$\mathcal{D}_{\mathbf{\Delta}} \otimes \mathcal{D}'_{\mathbf{\Delta}}$



Interleaving for CTMCs

$\mathcal{M} \,|||\, \mathcal{M}'$



Connecting DTMC and CTMC Composition

- For a given time step Δ a discretised exponential distribution is a geometric distribution.
- In the limit $\Delta \to 0$ a geometric distribution is an exponential distribution.
- This can be lifted to MCs:
  For each CTMC $\mathcal{M}$ and time step Δ, there is discretised DTMC $\mathcal{D}_{\boldsymbol{\Delta}}$.

Lemma

Let

- $\mathcal{M} = (S, \mathbf{Q}, s_0)$ *and* $\mathcal{M}' = (S', \mathbf{Q}', s_0')$ *be two CTMCs,*
- $\mathcal{D}_{\boldsymbol{\Delta}}, \mathcal{D}'_{\boldsymbol{\Delta}}$ *be the corresponding discretised DTMCs for time step* Δ*,*
- $\mathbf{Q}^{|||}$ *be the generator matrix of* $\mathcal{M} \,|||\, \mathcal{M}'$*,*
- $\mathbf{P}^{\otimes}_{\boldsymbol{\Delta}}$ *be the probability matrix of* $\mathcal{D}_{\boldsymbol{\Delta}} \otimes \mathcal{D}'_{\boldsymbol{\Delta}}$*.*

Then,

$$\mathbf{Q}^{|||} = \lim_{\Delta \to 0} (\mathbf{P}^{\otimes}_{\boldsymbol{\Delta}} - \mathbf{I})/\Delta$$

Recall:

How to decorate a DTMC with actions? (3)

So far: A DTMC is a tuple: $(S, \mathbf{P}, \pi(0), \ldots)$ where

- S is the set of states,
- $\mathbf{P} : S \times S \rightarrow [0,1]$ with $\sum_{s' \in S} \mathbf{P}(s,s') = 1$ is the transition matrix,
- $\pi(0)$ is the initial distribution.

Instead:

- S is the set of states,
- $\mathbf{P} : S \times S \rightarrow [0,1]$ with $\sum_{s'} \mathbf{P}(s,s') = 1$, the transition matrix,
- $\longrightarrow \subseteq S \times \mathrm{Act} \times S$, a labelled transition relation,
- $\pi(0)$ is the initial distribution.

This model goes back to Hans Hansson, with some twisting it is equipotent.

How to decorate a CTMC with actions?

So far: A continuous-time Markov chain is a tuple: $(S, \mathbf{Q}, \pi(0), ...)$ where

- S is the set of states,
- $\mathbf{Q} : S \times S \rightarrow \mathbb{R}$ is the generator matrix such that $q_{ij} \geq 0$
- $\pi(0)$ is the initial distribution

Instead: An interactive Markov chain is a tuple: $(S, \mathrm{Act}, \rightarrow, \mathbf{Q}, \pi(0), ...)$ where

- S is the set of states,
- $\mathbf{Q} : S \times S \rightarrow \mathbb{R}$ is the generator matrix such that $q_{ij} \geq 0$
- $\longrightarrow : S \times \mathrm{Act} \times S$, a labelled transition relation
- $\pi(0)$ is the initial distribution

Interleaving operator for interactive Markov chains

$\mathcal{I}_1 = (S_1, \mathbf{Q}_1, \ldots)$ $\mathcal{I}_2 = (S_2, \mathbf{Q}_2, \ldots)$

The composite transition system $\mathcal{I}_1 \,|||\, \mathcal{I}_2$ is:

$$\mathcal{I}_1 \,|||\, \mathcal{I}_2 \;=\; (\mathsf{S}_1 \times \mathsf{S}_2,\ \mathsf{Act}_1 \cup \mathsf{Act}_2,\ \ldots)$$

where $\longrightarrow$ and $\mathbf{Q}$ are given by:

$$\frac{s_1 \xrightarrow{\alpha}_1 s_1'}{\langle s_1, s_2\rangle \xrightarrow{\alpha} \langle s_1', s_2\rangle} \qquad \frac{s_2 \xrightarrow{\alpha}_2 s_2'}{\langle s_1, s_2\rangle \xrightarrow{\alpha} \langle s_1, s_2'\rangle}$$

$$\frac{s_1 \xrightarrow{\lambda}_1 s_1'}{\langle s_1, s_2\rangle \xrightarrow{\lambda} \langle s_1', s_2\rangle} \qquad \frac{s_2 \xrightarrow{\lambda}_2 s_2'}{\langle s_1, s_2\rangle \xrightarrow{\lambda} \langle s_1, s_2'\rangle}$$

Recall:

Synchronous message passing for transition systems

$\mathcal{T}_1 = (S_1, \mathrm{Act}_1, \longrightarrow_1, \ldots)$ $\mathcal{T}_2 = (S_2, \mathrm{Act}_2, \longrightarrow_2, \ldots)$

The concurrent execution with **synchronization** over all actions in Syn is:

$$\mathcal{T}_1 \,\|_{\mathrm{Syn}}\, \mathcal{T}_2 \;=\; (S_1 \times S_2, \mathrm{Act}_1 \cup \mathrm{Act}_2, \to, \ldots)$$

where $\mathrm{Syn} \subseteq \mathrm{Act}_1 \cap \mathrm{Act}_2$ set of synchronization actions

interleaving for $\alpha \in \mathrm{Act}_i \setminus \mathrm{Syn}$:

$$\frac{s_1 \xrightarrow{\alpha}_1 s_1'}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1', s_2 \rangle} \qquad \frac{s_2 \xrightarrow{\alpha}_2 s_2'}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1, s_2' \rangle}$$

handshaking (rendezvous) for $\alpha \in \mathrm{Syn}$:

$$\frac{s_1 \xrightarrow{\alpha}_1 s_1' \quad \wedge \quad s_2 \xrightarrow{\alpha}_2 s_2'}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1', s_2' \rangle}$$

Synchronous message passing for IMCs

$\mathcal{I}_1 = (S_1, \mathsf{Act}_1, \longrightarrow_1, \ldots)$ $\qquad$ $\mathcal{I}_2 = (S_2, \mathsf{Act}_2, \longrightarrow_2, \ldots)$

The concurrent execution with **synchronization** over all actions in Syn is:

$$\mathcal{I}_1 \parallel_{\mathsf{Syn}} \mathcal{I}_2 \;=\; (S_1 \times S_2, \mathsf{Act}_1 \cup \mathsf{Act}_2, \rightarrow, \ldots)$$

where $\mathsf{Syn} \subseteq \mathsf{Act}_1 \cap \mathsf{Act}_2$ set of synchronization actions

interleaving for $\alpha \in \mathsf{Act}_i \setminus \mathsf{Syn}$:

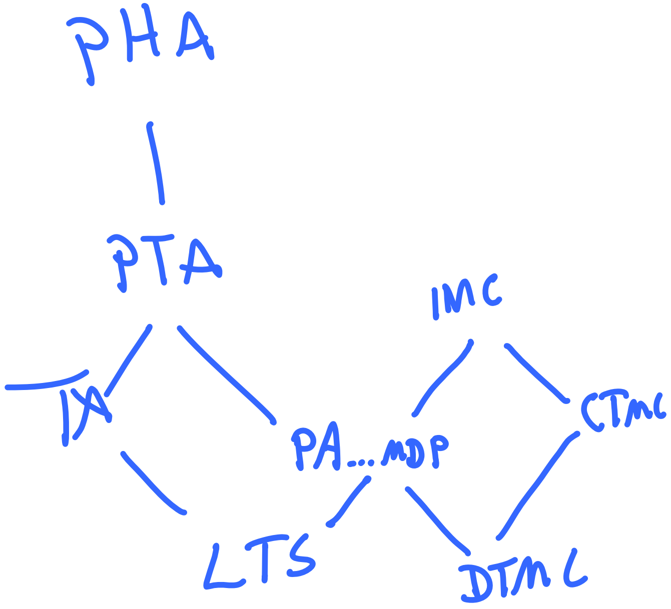
$$\frac{s_1 \xrightarrow{\alpha}_1 s_1'}{\langle s_1, s_2\rangle \xrightarrow{\alpha} \langle s_1', s_2\rangle} \quad \frac{s_2 \xrightarrow{\alpha}_2 s_2'}{\langle s_1, s_2\rangle \xrightarrow{\alpha} \langle s_1, s_2'\rangle} \quad \frac{s_1 \xrightarrow{\lambda}_1 s_1'}{\langle s_1, s_2\rangle \xrightarrow{\lambda} \langle s_1', s_2\rangle} \quad \frac{s_2 \xrightarrow{\lambda}_2 s_2'}{\langle s_1, s_2\rangle \xrightarrow{\lambda} \langle s_1, s_2'\rangle}$$

handshaking (rendezvous) for $\alpha \in \mathsf{Syn}$:

$$\frac{s_1 \xrightarrow{\alpha}_1 s_1' \quad \wedge \quad s_2 \xrightarrow{\alpha}_2 s_2'}{\langle s_1, s_2\rangle \xrightarrow{\alpha} \langle s_1', s_2'\rangle}$$

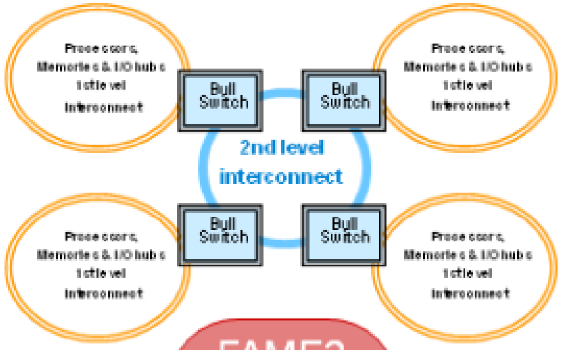
Agenda

- Add Rewards (or Costs)
- Go beyond exponential distributions
- Develop abstraction techniques
- Look into continuous dynamics
- Integrate PA and IMC
- Tools, Tools, Tools
- Applications, Applications, Applications, Applications, Applications, Applications, Applications, Applications, Applications, Applications



MINALOGIC / MULTIVAL

http://www.inrialpes.fr/vasy/multival/


Processors, Memories & I/O hubs 1st level interconnect
Bull Switch
Bull Switch
2nd level interconnect
Bull Switch
Bull Switch
Processors, Memories & I/O hubs 1st level interconnect
Processors, Memories & I/O hubs 1st level interconnect
Processors, Memories & I/O hubs 1st level interconnect

Functional verification & performance evaluation of asynchronous hardware using formal methods

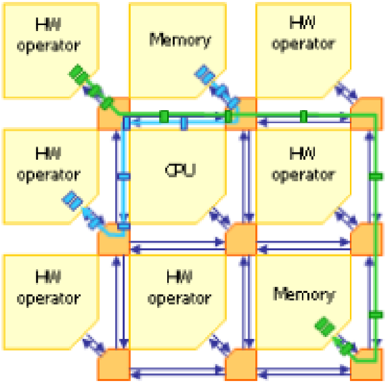
FAME2
Bull

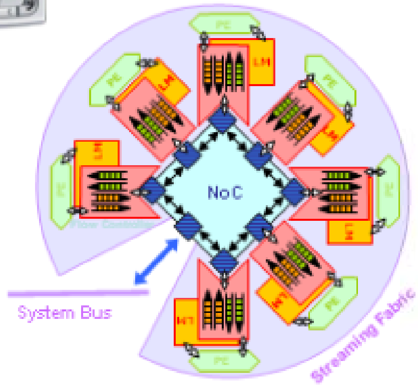
Distributed and asynchronous systems

They have to be validated

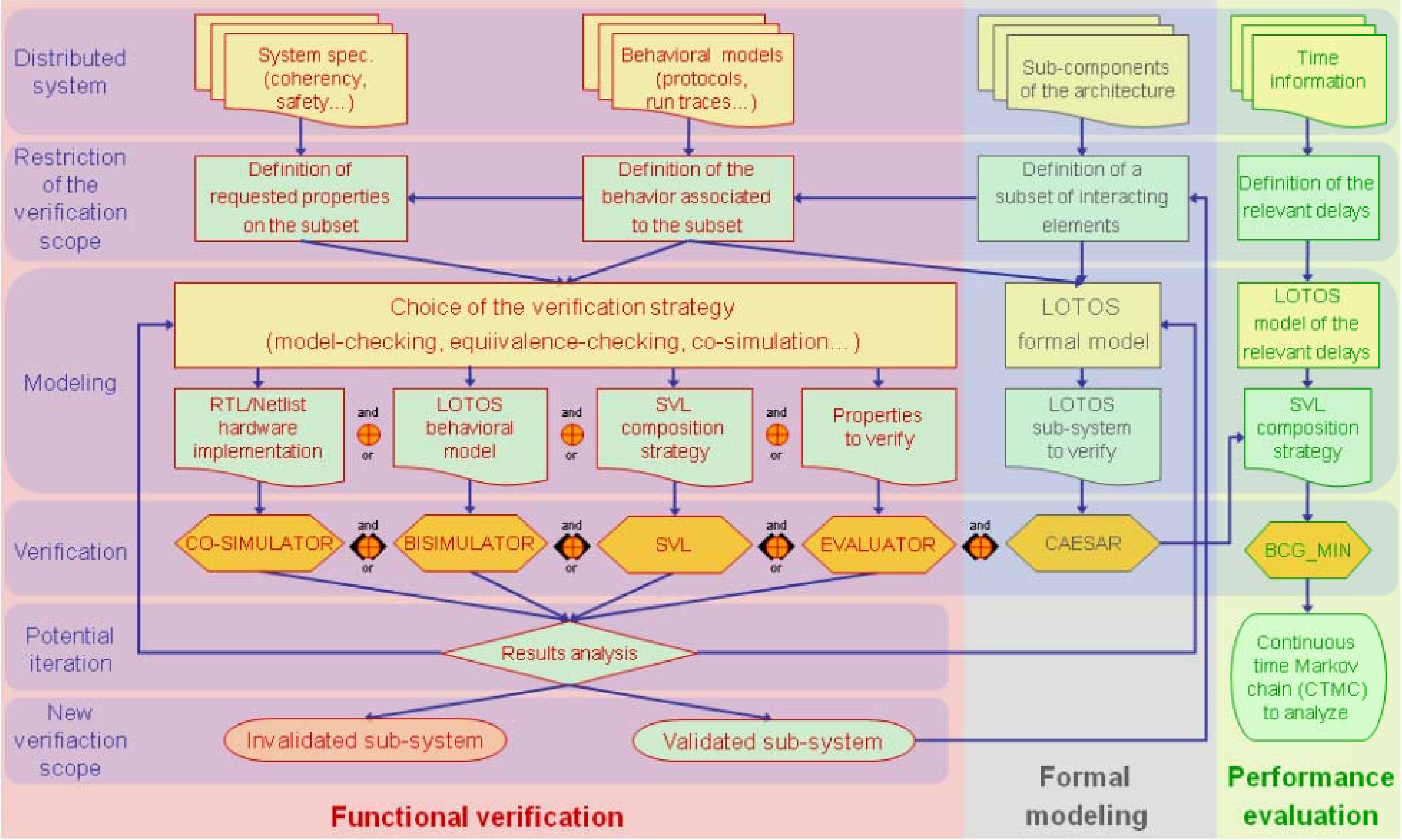
FAUST
CEA Léti

xSTream
ST



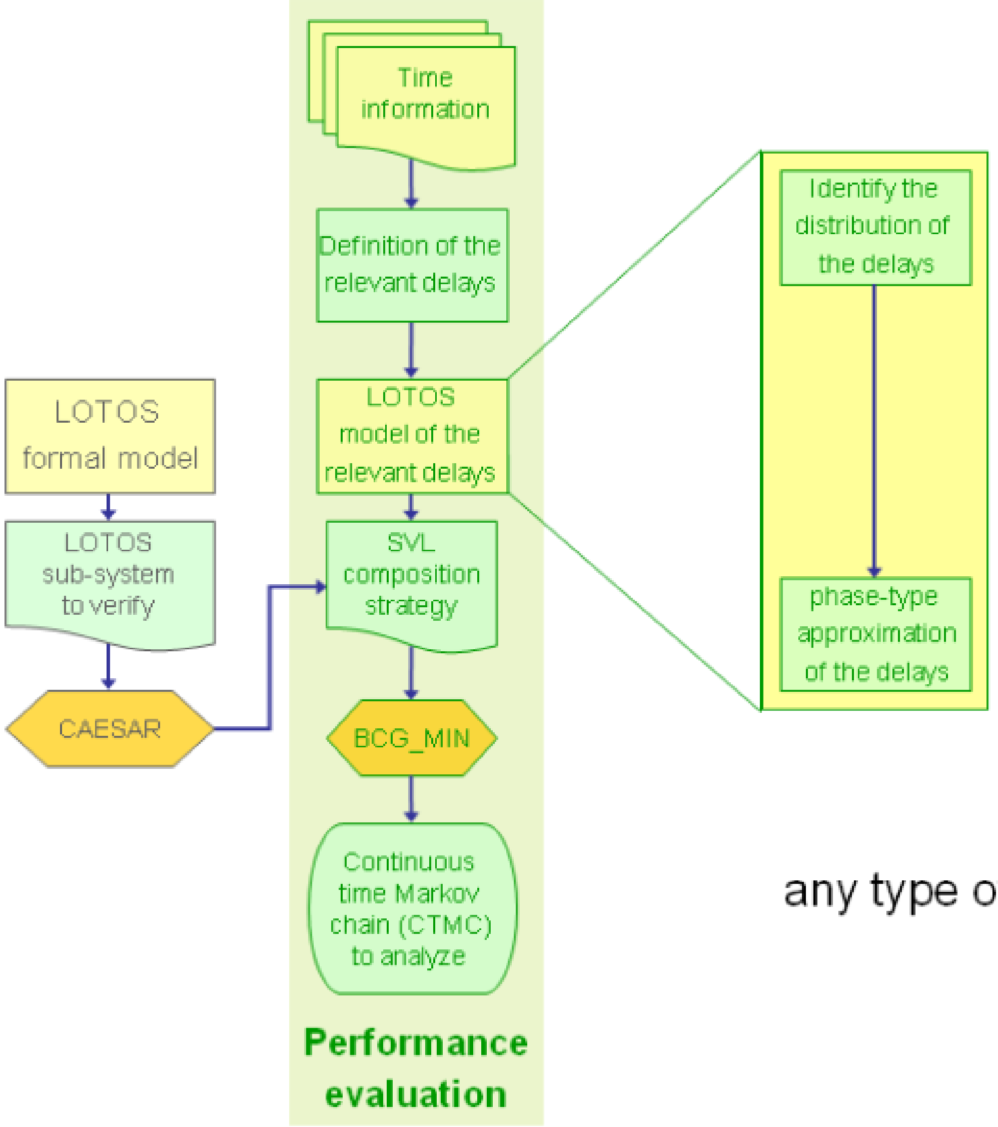


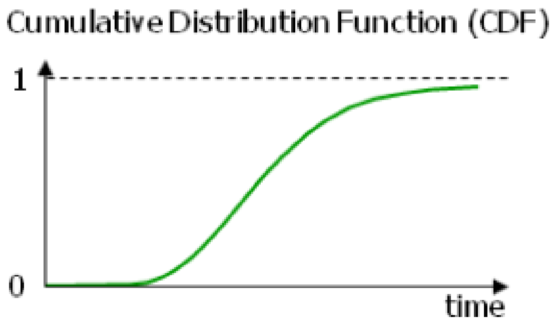
CADP methodology

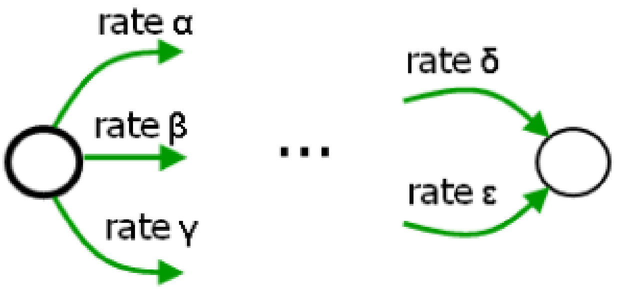


Distributed system
System spec. (coherency, safety...)
Behavioral models (protocols, run traces...)
Sub-components of the architecture
Time information
Restriction of the verification scope
Definition of requested properties on the subset
Definition of the behavior associated to the subset
Definition of a subset of interacting elements
Definition of the relevant delays
Modeling
Choice of the verification strategy (model-checking, equiivalence-checking, co-simulation...)
LOTOS formal model
LOTOS model of the relevant delays
RTL/Netlist hardware implementation
and
or
LOTOS behavioral model
SVL composition strategy
Properties to verify
LOTOS sub-system to verify
SVL composition strategy
Verification
CO-SIMULATOR
BISIMULATOR
SVL
EVALUATOR
CAESAR
BCG_MIN
Potential iteration
Results analysis
Continuous time Markov chain (CTMC) to analyze
New verifiaction scope
Invalidated sub-system
Validated sub-system
Formal modeling
Performance evaluation
Functional verification

From LOTOS to Markov chains

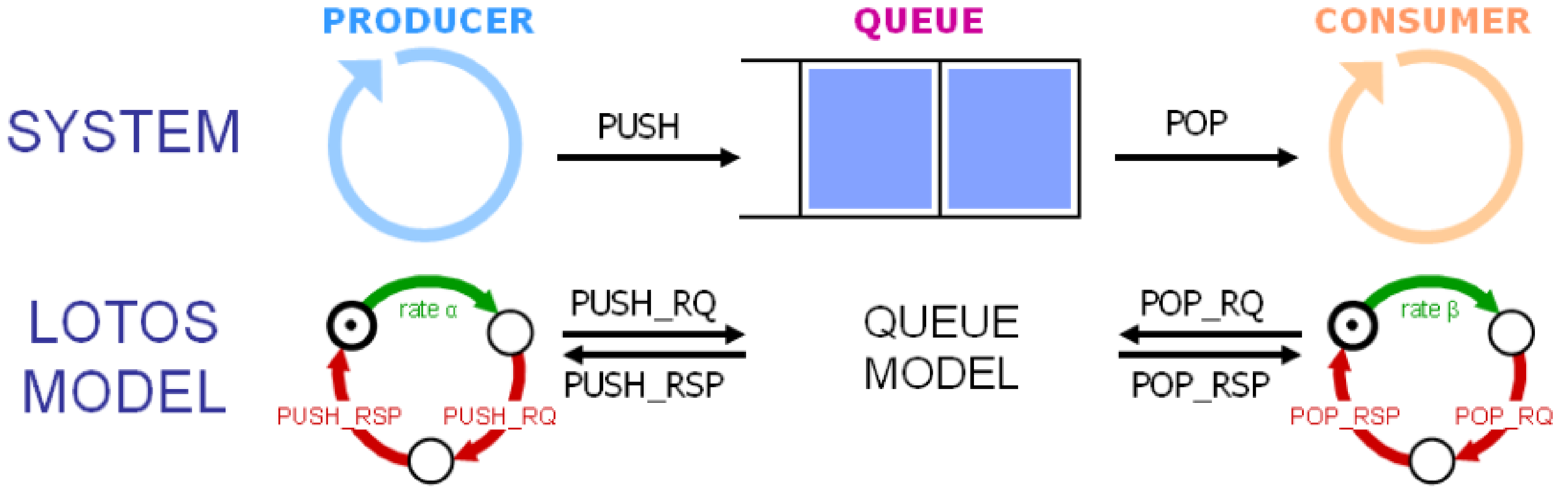

Time information
Definition of the relevant delays
LOTOS model of the relevant delays
SVL composition strategy
BCG_MIN
Continuous time Markov chain (CTMC) to analyze
Performance evaluation
LOTOS formal model
LOTOS sub-system to verify
CAESAR
Identify the distribution of the delays
phase-type approximation of the delays





any type of general distribution can be fitted by a phase-type distribution

Example: a simplified xSTream queue



PRODUCER

QUEUE

CONSUMER

SYSTEM

PUSH

POP

LOTOS MODEL

rate α

PUSH_RSP

PUSH_RQ

PUSH_RQ

PUSH_RSP

QUEUE MODEL

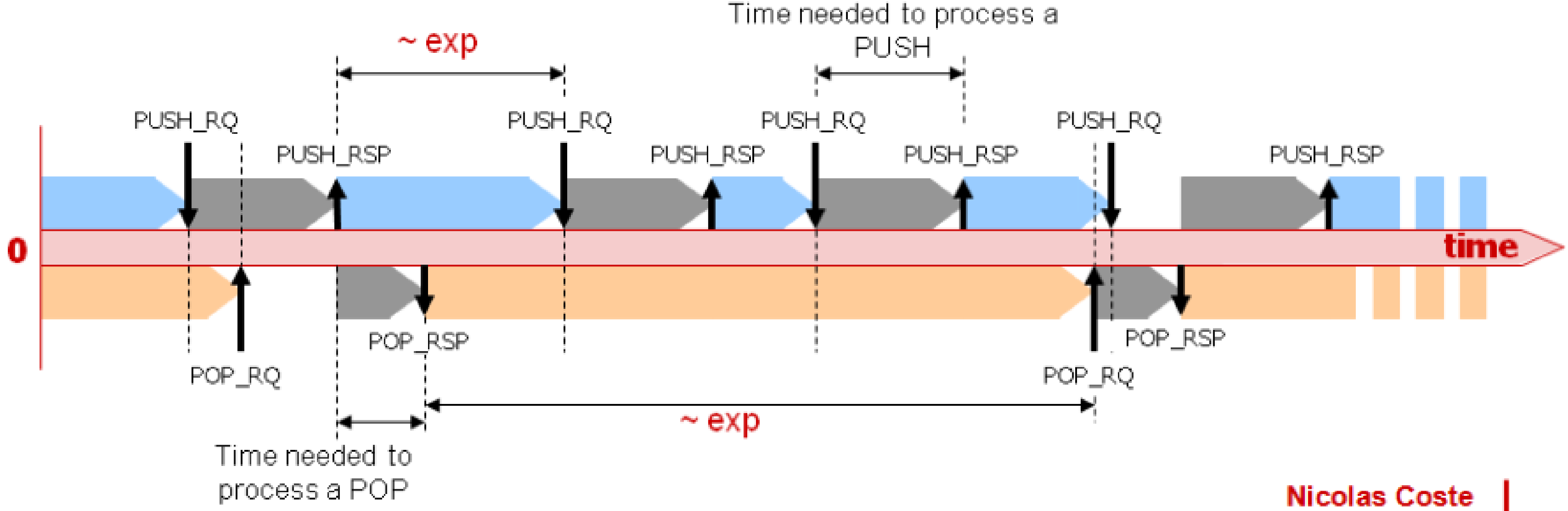
POP_RQ

POP_RSP

rate β

POP_RSP

POP_RQ



~ exp

Time needed to process a PUSH

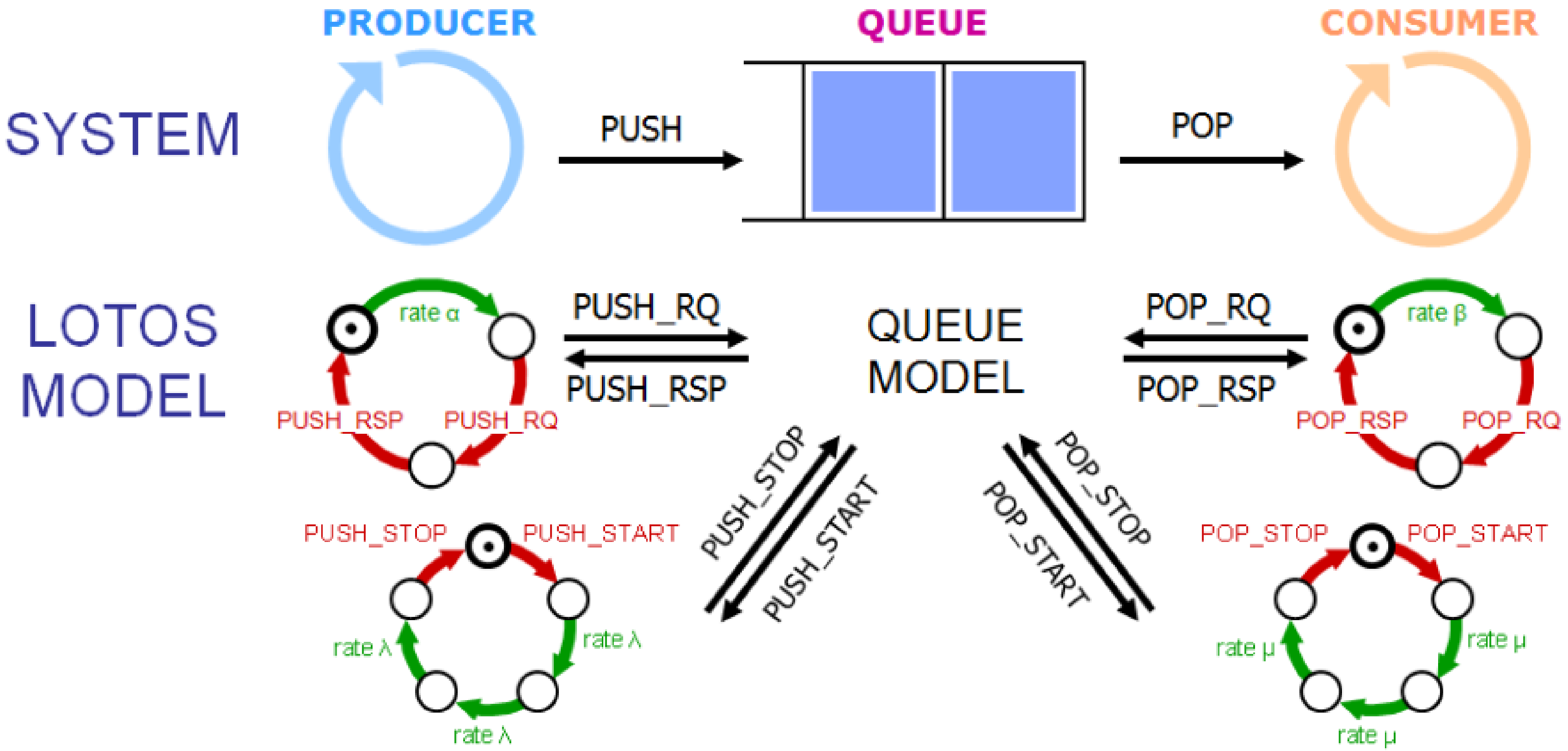
0

time

~ exp

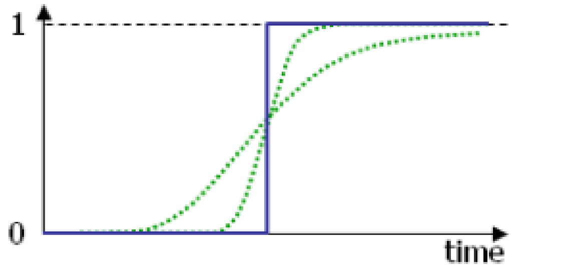
Time needed to process a POP

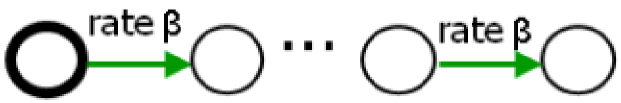
Example: a simplified xSTream queue



Time needed to process a PUSH and time needed to process a POP

Cumulative Distribution Function (CDF)

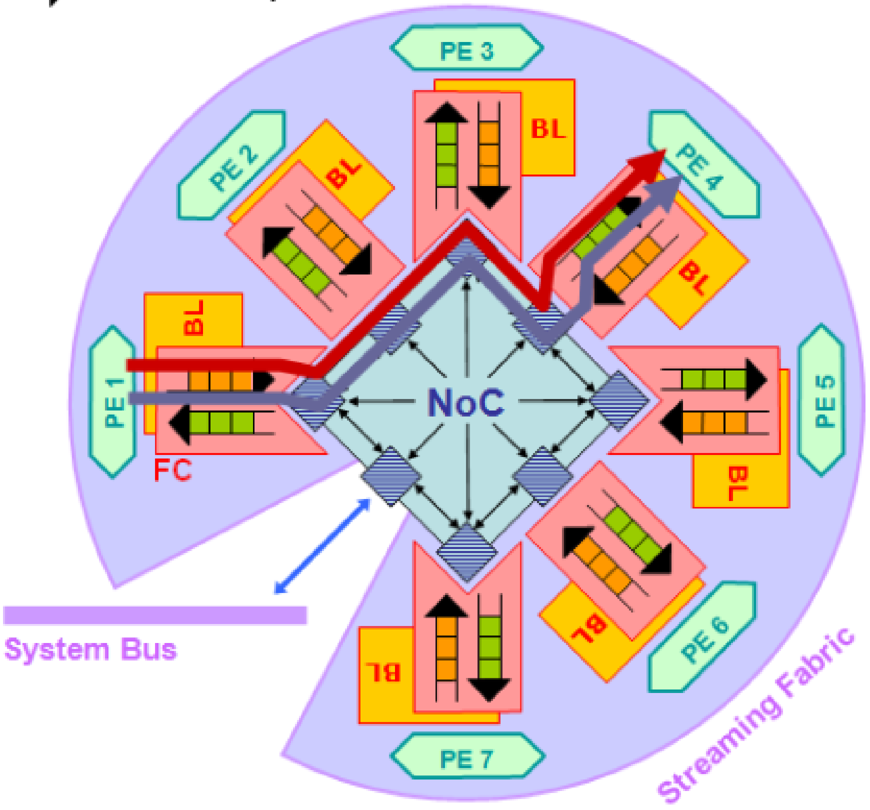




Erlang distribution

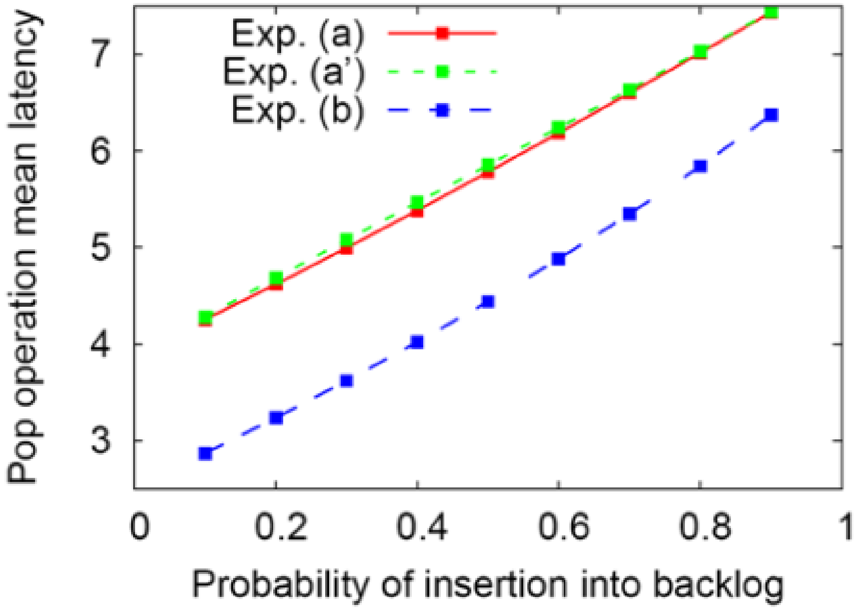
The xSTream Case-Study

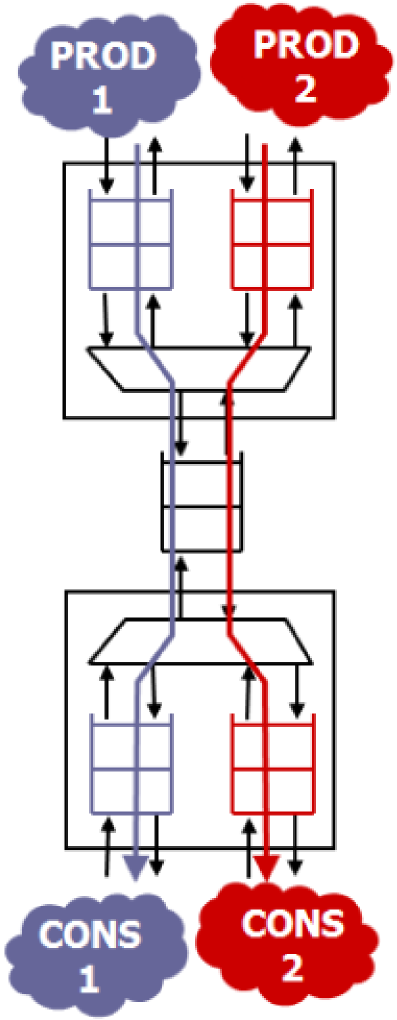
PE	Processing Element
NoC	Network on Chip
BL	BackLog Memory
FC	Flow controller
⇉	xSTream queue



- Two flows from PE 1 to another PE, say PE 4.
- Time to remove an element from a Pop queue ?
  - Available elements → Pop done
  - No element → Pop blocked
- A Pop latency close to its minimal value ensures that the communication architecture fulfills its task

The xSTream Case-Study : results





- Experiments (a) and (a') : credit protocol worst case

  The production/consumption rate in experiment (a') are greater than in experiment (a)

- Experiment (b) : a better credit protocol use

Agenda

- Add Rewards (or Costs)
- Go beyond exponential distributions
- Develop abstraction techniques
- Look into continuous dynamics
- Integrate PA and IMC
- Tools, Tools, Tools
- Applications, Applications, Applications, Applications, Applications, Applications, Applications, Applications, Applications, Applications